A Scalable Approach for Statistical Learning in Semantic Graphs

Editor(s): Claudia d'Amato, Universita degli Studi di Bari, Italy
Open review(s): Paulo Cesar G. Costa, George Mason University, US; Joaquin Vanschoren, Leiden University, Netherlands; Lorand Dali, Jozef Stefan Institute, Slovenia

Yi Huang [a,*], Volker Tresp [a,b], Maximilian Nickel [b], Achim Rettinger [c] and Hans-Peter Kriegel [b]
[a] *Corporate Technology, Siemens AG, Otto-Hahn-Ring 6, 81739 Munich, Germany*
E-mail: {yihuang,volker.tresp}@siemens.com
[b] *Department of Computer Science, Ludwig Maximilian University, Oettingenstraße 67, 80538 Munich, Germany*
E-mail: {nickel,kriegel}@dbs.ifi.lmu.de
[c] *Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany*
E-mail: rettinger@kit.de

Abstract.
Increasingly, data is published in the form of semantic graphs. The most notable example is the Linked Open Data (LOD) initiative where an increasing number of data sources are published in the Semantic Web's Resource Description Framework and where the various data sources are linked to reference one another. In this paper we apply machine learning to semantic graph data and argue that scalability and robustness can be achieved via an urn-based statistical sampling scheme. We apply the urn model to the SUNS framework which is based on multivariate prediction. We argue that multivariate prediction approaches are most suitable for dealing with the resulting high-dimensional sparse data matrix. Within the statistical framework, the approach scales up to large domains and is able to deal with highly sparse relationship data. We summarize experimental results using a friend-of-a-friend data set and a data set derived from DBpedia. In more detail, we describe novel experiments on disease gene prioritization using LOD data sources. The experiments confirm the ease-of-use, the scalability and the good performance of the approach.

Keywords: Statistical Machine Learning, Linked Open Data, Semantic Web, Statistical Relational Learning, Linked Life Data

1. Introduction

Increasingly, data is published in the form of semantic graphs. The most notable example is the Linked Open data (LOD)[1] initiative where an increasing number of data sources are published in the Semantic Web's Resource Description Framework (RDF) [2] and where the various data sources are linked to reference one another. LOD forms a huge semantic graph with links from subject nodes to object nodes. Information access is enabled via query languages such as SPARQL —potentially enriched with reasoning— retrieval and search. LOD is potentially also a great source for machine learning applications, which is the topic of this paper. In particular we focus on Statistical Relational learning (SRL) which is able to exploit statistical patterns in interlinked domains. This is in contrast to regular machine learning, which exploits patterns in attribute-based representations, and to Inductive Logic Programming (ILP), which concerns machine learning in interconnected deterministic, or close-to-deterministic, domains. We argue that LOD has particular properties that should be taken into account in SRL. Most importantly, LOD is typ-

[*] Corresponding author.

ically not collected systematically as the data originates from different sources and is also incomplete in many cases. This is true, in particular, for relationships across different data sources: the referencing between data sources is typically sparse and incomplete. Thus in our work we have a preference for approaches that can deal with the sparsity of the data, the incompleteness of the data, and which scale well with the growing size of LOD. An approach fulfilling these requirements is the SUNS[1] approach, which has been introduced in a series of papers [3,4,5] and is readily applicable to LOD data. Here we review the SUNS approach and describe applications to different LOD domains. We also show how ontological background knowledge can be integrated into SUNS and how learned information can be integrated into an extended SPARQL language for querying. A novel contribution of this paper is a detailed description of the application of the SUNS model to Linked Life Data (LLD) [6], which is a curated subset of LOD with a focus on life science data.

In the next section, we review related work. In Section 3 we discuss machine learning in the context of LOD and define a setting with a clear statistical interpretation. In Section 4 we describe the learning approach. The following sections are concerned with experiments using LOD. In Section 5 we discuss the application of the SUNS model to social network data from LOD and illustrate how learned results can be queried using an extended SPARQL query. In Section 6 we apply our approach to LOD's DBpedia data. Here, we demonstrate how ontological background knowledge can be integrated into learning. Section 7 discusses the application to LLD where the task is to predict new gene disease relationships from known relationships. Section 8 presents our conclusions.

2. Related Work and Some Background

2.1. Related Learning Approaches

In [7] the authors describe SPARQL-ML, a framework for adding data mining support to SPARQL. SPARQL-ML was inspired by Microsoft's Data Mining Extension (DMX). A particular ontology for specifying the machine learning experiment is developed. The approach uses Relational Bayes Classifier (RBC) and Relational Probabilistic Trees (RPT).

The work on inductive databases [8] also considers relational representations but is focused on the better curated data situation in relational databases.

Bayesian approaches (examples that are suitable for the relational domains are [9,10,11,12]) are quite flexible and interpretable and provide a probability distribution over a relational domain (see Section 3.2). Experiments on LOD data are still quite limited [13].

Inductive logic Programming (ILP) considers deterministic or close-to-deterministic dependencies. Typical examples are many ILP approaches [14,15] and propositionalized ILP approaches [16,17]. [18] describes a perspective of ILP for the Semantic Web.

Multivariate prediction generalizes supervised learning to predict several variables jointly, conditioned on some inputs. The improved predictive performance in multivariate prediction, if compared to simple supervised learning, has been attributed to the sharing of statistical strength between the multiple tasks, i.e., data is used more efficiently (see [19] and citations therein for a review). Due to the large degree of sparsity of the relationship data in typical semantic graph domains, we expect that multivariate prediction is quite interesting and we apply it in this paper.

Recently, there has been quite some work on the relationship between kernels and graphs [20,21,22,23]. Relation prediction on graphs is quite related to semi-supervised learning as surveyed in [24] where the goal is to predict node labels based on known node labels in a graph. Kernels for semi-supervised learning have, for example, been derived from the spectrum of the Graph-Laplacian. In [25,26] approaches for Gaussian process based link prediction have been presented. Link prediction in relational graphs has also been covered from the relational learning and the ILP communities [27,28,29]. Kernels for semantically rich domains have been developed by [30]. Link prediction is covered and surveyed in [31,32]. Inclusion of ontological prior knowledge to relational learning has been discussed in [33].

In last years, tensor factorization based approaches have been also applied to relational learning. [34] introduces the Bayesian Clustered Tensor Factorization (BCTF) and applies it to various smaller relational data sets. TripleRank [35] applies the CP [36] tensor decomposition to RDF graphs for faceted browsing. A scalable tensor factorization approach to relational learning, RESCAL, proposed in [37], performs collective learning and can be applied to classification and entity resolution problems. Very recently, it has been used to factor the YAGO2 ontology [38].

[1] SUNS stands for *Statistical Unit Node Set*.

From early on there has been considerable work on supporting ontologies using machine learning [39,40, 41], while data mining perspectives for the Semantic Web have been described by [42,43]. [44] provides a survey of the statistical learning models and related techniques for Semantic Web data representations.

2.2. Related Work on Disease Gene Prioritization

Kann [45] provides an up-to-date review on the latest advances in the field of translational bioinformatics, focusing on the advances of computational techniques to search for and classify disease genes.

Certain gene properties differentiate disease genes and have been used as the basis for computational tools to prioritize disease gene candidates derived from these experiments. All of the current approaches are based on the integration of different sources such as: gene function (disease genes are expected to share common functional properties), pathways (disease genes are most likely to share common pathways), gene expression (disease genes are expected to be co-expressed), gene regulation (genes within the same gene-regulation network are expected to affect similar diseases), sequence properties, and protein interaction (disease genes are often highly connected with other genes from the same disease).

Current limitations result from the amount and quality of the available experimental data generated by these techniques is a major limitation of the gene-prioritization techniques. For instance, protein-protein interaction-based methods suffer from the incompleteness and low quality of the data currently available for interaction networks in mammals. Another source of uncertainty is the disease mapping information used to train and evaluate the computational methods, for it is of variable resolution and expected to contain large numbers of false positives.

Machine learning methods, e.g., decision trees and similarity based methods are widely used. For example, PhenoPred derives a similarity score that represents the chance of the gene disease association to be true [46]. In the following we will compare our system with the ToppGene Suite, which is one of the state-of-the art approaches with an easy to use interface [47]. It uses a fuzzy-based similarity measure between the genes in the training and test set based on their semantic annotation.

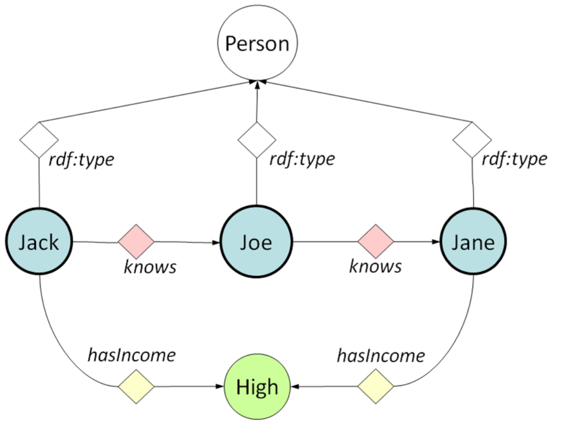


Fig. 1. Example of an RDF graph displaying a social friendship network in which the income of a person is an attribute. Concepts (resources) are represented by circular nodes and triples are represented by labeled directed links from subject node to object node. The diamond-shaped nodes stand for random variables which are in state *one* if the corresponding triples exist. Nodes representing statistical units (here: *Persons*) have a darker rim.

3. Statistical Learning in Semantic Graphs

3.1. RDF and Linked Open Data

Semantic domains can be represented as directed labeled graphs where nodes stand for concepts such as objects, categories or attributes and links describe simple subject-predicate-object statements: a directed arc points from the subject node (e.g., representing an object or a category), to an object node (e.g., representing an object, a category or an attribute) (Figure 1). The link is labeled by the predicate. Examples of such semantic graphs are the RDF graphs of the Semantic Web [2] and LOD [1], the underlying data structure in the YAGO2 ontology [48] and the semantic graph format in [49]. We assume that the graph only contains links that are known to exist.[2] In this paper we consider the learning task of predicting links which are not present in the semantic graph but which likely exist, considering statistical patterns that are implicit in the data. For example, we might predict the likelihood that Jack wants to be friends with Jane or that Jack's income is high or that Jack is a human being, and not a parrot.

In the next sections we briefly discuss statistical approaches to networked data and motivate our approach.

[2] A link, resp. the associated subject-predicate-object statement, is sometimes referred to as a triple.

3.2. Network Models in Statistical Relational Learning

Let's consider the following mapping between a semantic RDF graph and a probabilistic model. In semantic graphs, nodes represent *objects* in the real world and links represent simple subject-predicate-object *statements*. In a probabilistic setting the RDF links become random variables (Figure 1). In a graphical probabilistic model, the *random variables*, i.e. the links in the RDF graph, become nodes and links between the random variables represent *direct probabilistic dependencies*. Bayesian graphical models with directed links and Markov networks with undirected links are two common representatives of graphical probabilistic models.

Since in most networked domains it is reasonable to assume that information at one location influences via some paths information at other locations in the network, a graphical probabilistic model would reflect this global dependency as well. Statistical network models, which have been developed from this view point are, e.g., PRMs [9], MLNs [10], the IHRM [11], IRM [12] and, most recently, tensor models [37]. Advantages of these approaches are their elegance, power and generality. A disadvantage is that learning and prediction can be computationally expensive since inference needs to take into account all evidence in the network by crossing over all nodes in the network. The problem can sometimes be alleviated, e.g., by lifted inference [50] and by exploiting the structure of the network. Another problem here is that generalization to new entities requires that these entities and their structure are integrated into the existing graphical probabilistic model, requiring inference over the joint domain.

One might argue that a graphical probabilistic model modeling a complete RDF graph is most close to the ground truth and might reflect the true belief of a user about dependencies, independencies, and probabilistic distributions. This fits well to a Bayesian view as a rational measure of belief, and in fact, most network models in SRL follow a Bayesian approach [9,10,11, 12].

3.3. Approaches with a Frequentist Flavor Using an Urn Model

In a frequentist approach, probabilities are defined in the limit of an infinite number of trials. At first, it might appear to be difficult to apply frequentist modeling to networked domains since one often needs to work with only one network (e.g., one university domain). On the other hand this seems to be a common situation in statistical modeling: e.g., if one samples persons and studies their properties, one often ignores the fact that these persons participate in different networks, e.g., that two persons might be members of the same family. A statistical setting here can be well defined considering an urn process. In statistics, an urn problem is an idealized mental exercise in which some objects of real interest are represented as colored balls in an urn or other container and are the statistical units in the experiment. One pretends to draw (remove) one or more balls from the urn; the goal is to determine the probability of drawing one color or another, or some other properties. If objects are returned to the urn (sampling with replacement), a statistical process is well-defined and generalization from the sample to the population via inductive statistics is well-defined as well. In our context, the objects of interest (the "balls") are defined as entities with specified properties in the semantic graph, e.g., all students in Munich. These entities define the balls in the urn, i.e., the population. A sample is the formed by randomly selecting entities from the population.

Generalization to entities outside of the population (i.e., beyond the semantic graph in the data base) can also be guaranteed if both the entities in the population and the entities in the test set can be considered unbiased samples of a common population. Thus, we consider the situations where, e.g., the population consists of all 20-year old students in Munich and the sample consists of a subset. Statistical inference allows us to generalize from the sample to the population. The question whether one can generalize to another population, e.g., to all students in Munich, depends on the fact if the new population is similar to the original population. e.g., if 20-year old students are statistically identical to the whole student population.

We want to note that a Bayesian analysis using the urn model is possible as well. It would now be required that the user formulates the prior modeling assumptions with respect to the urn sampling model.

As in any statistical analysis, care must be taken. Some pitfalls specific to a networked domain are discussed in [51].

4. Scalable Kernel Machine Learning for Semantic Graphs

The SUNS approach is based on the scalable urn model described in Section 3.3 and has been introduced in [3] where a feature-based approach was described. Since the number of features is very large in the applications we are considering in this publication, we formulate the problem in terms of inner products of features, i.e., we use a kernel formulation. The kernel solution is made scalable by using the Nyström approximation.

4.1. The Nyström Approximation

We now assume that for any two statistical units i and j in the population a kernel $k_{i,j}$ is defined. A subset of the population of size N, i.e., the sample, defines the training set. Let K be the kernel matrix (i.e., Gram matrix) for the training instances. In many applications N can be very large, therefore we now follow [52] and use the Nyström approximation to scale up kernel computations to large data sets.

The Nyström approximation is based on an approximation to eigen functions and starts with the eigen decomposition

$$K = UDU^\top \tag{1}$$

of the kernel matrix. The Nyström approximation to the kernel for two arbitrary instances i and j can be written as

$$k_{i,j} \approx k_{.,i}^\top \, U_r \; \text{diag}_r\,(1/d_l) \;\; U_r^\top \; k_{.,j}$$

where $\text{diag}_r\,(1/d_l)$ is a diagonal matrix containing the inverse of the r leading eigenvalues in D and where U_r contains the corresponding r columns of U.[3] Here, $k_{.,i}$ is a vector of kernels between instance i and the training instances.

The vector of approximate kernels between a statistical unit i and all units in the training data can be written as

$$k_{.,i} \approx U_r U_r^\top k_{.,i} \tag{2}$$

and the matrix of approximate kernels between all pairwise units in the training data is

$$K \approx U_r \; \text{diag}_r\,(d_l)\, U_r^\top. \tag{3}$$

These modified kernels can now be used in kernel approaches such as SVM learning or Gaussian process learning. In particular, the reduced rank approximation Equation 3 can greatly reduce the computational requirements [52].[4]

4.2. Example: Regularized Least Squares Solutions for Multivariate Prediction

We now assume that for an instance i we have L targets or random variables $y_i = (y_{i,1}, \ldots, y_{i,L})^\top$ available. We want to train a model of the form $\hat{y}_i = k^\top(.,i)W$ where W is an $N \times L$ weight matrix.

A regularized least squares cost function can be formulated as

$$\text{trace}(Y - KW)(Y - KW)^\top + \lambda \; \text{trace} W^\top K W$$

where $Y = (y_1, \ldots, y_N)^\top$ and where $\lambda \geq 0$ is a regularization parameter. If we use the Nyström approximation for the kernels we obtain as least squares solution for the weight matrix

$$W_{LS} = U \; \text{diag}_r \left(\frac{1}{d_l + \lambda} \right) \; U^\top Y.$$

The prediction for the training data (i.e., in smoothing or transduction) is

$$\hat{Y} = U \; \text{diag}_r \left(\frac{d_l}{d_l + \lambda} \right) \; U^\top Y$$

and in general

$$\hat{y}_i = k^\top(.,i) W_{LS}. \tag{4}$$

We now consider some special kernels. Assume that for each instance i, in addition to the random variables of interest y_i, we also have covariates x_i available. Covariates might, for example, represent aggregated information. If the kernel can be written as an inner product of the covariates $k^x_{i,j} = x_i^\top x_j$, our Nyström ap-

[3] Based on this approximation the rank of any kernel matrix is less than or equal to $r \leq N$.

[4] We use the Nyström approximation slightly differently from [52]. There, Equation 1 is used on a submatrix of K and Equation 2 is then used to approximate K.

proximation is equivalent to regularized PCA regression in that covariate space. Another interesting solution is when $k^y_{i,j} = y_i^\top y_j$ in which case our Nyström approximation is equivalent to regularized matrix reconstruction via PCA, often used in collaborative filtering. Note that in the latter case the low rank Nyström approximation is not only a necessity to obtain a scalable solution but is also necessary to obtain valid predictions at all: with $\lambda \to 0$ and $r = N$ we would obtain the trivial $\hat{Y} = Y$. Finally, with $k^z_{i,j} = z_i^\top z_j$ where $z_i = (\alpha x_i^\top, y_i^\top)^\top$, we obtain the reduced rank penalized regression (RRPP) algorithm in the SUNS framework [4]. Here, α is a positive weighting factor balancing the influence of the two information sources.

4.3. Kernel for Semantic Graphs

So far the discussion has been quite general and the Nyström approximation can be used for any kernel defined between instances in the population. As discussed in Section 2, there are a number of interesting kernels defined for nodes in a graph but most of them are not directly applicable to the rich domain of a semantic graph with many different node types and many different relation types. An exception is [30], which defines kernels exploiting rich ontological background knowledge.

We here present the kernel based on the SUNS framework [53]. The random variables represent the likelihood of links where the statistical unit is the subject or object. Additional features describe aggregated information. Although features are explicitly calculated, a kernel approach is still preferred since in the applications that we are considering the number of features can be quite large whereas N, the size of the sample, can be controlled more easily.

4.3.1. The Random Variables or Targets in the Data Matrix

Figure 1 shows a simple semantic graph with nodes *Person, Jack, Joe, Jane, High* and relation types *rdf:type, knows, knows, hasIncome*. We now introduce for each potential triple a *triple node* drawn as a diamond-shaped node in Figure 1. A triple node is in state *one* (*true*) if the triple is known to exist and is in state *zero* (*false*) if the triple is known not to exist. Graphically, one only draws the triple nodes in state *one*, i.e., the existing triples.

We now associate some triples with statistical units. The idea is to assign a triple to a statistical unit if the statistical unit appears in the triple. Let's consider the statistical unit *Jane*. Based on the triples she is participating in, we obtain *(?personA, rdf:type, Person)*, *(Joe, knows, ?personA)*, and *(?personA, hasIncome, High)* where *?personA* is a variable that represents a statistical unit. The expressions form the random variables (outputs) and define columns in the data matrix.[5] By considering the remaining statistical units *Jack* and *Joe* we generate the expressions (columns), *(?personA, knows, Jane)* and *(Jack, knows, ?personA)*. We will not add *(Jane, knows, ?personA)* since Jane considers no one in the semantic graph to be her friend. We iterate this procedure for all statistical units in the sample and add new expressions (i.e., columns in the data matrix), if necessary. Note that expressions that are not represented in the sample will not be considered.

In [3] the triples associated with a statistical unit were denoted as *statistical unit node set* (SUNS). The data matrix formed with the N statistical units as rows and the random variables as columns is denoted as Y. Note that Y contains random variables derived for multiple different predicates.

4.3.2. Non-random Covariates in the Data Matrix

The columns in the data matrix that we have derived so far represent truth values of actual or potential triples. Those triples are treated as random variables in the analysis. If the machine learning algorithm predicts that a triple is very likely, we can enter this triple in the semantic graph. We now add columns to the data matrix that provide additional information for the learning algorithm but which we treat as covariates or fixed inputs.

First, we derive simplified relations from the semantic graph. More precisely, we consider the expressions derived in the last subsection and replace constants by variables. For example, from *(?personA, knows, Jane)* we derive *(?personA, knows, ?personB)* and count how often this expression is true for a statistical unit *?personA*, i.e., we count the number of friends of person *?personA*.

Also consider that often a textual description is available for a node from which non-random features can be derived (See the experiments in Section 6).

Second, we consider a simple type of aggregated covariate from outside a SUNS. Consider first a binary triple *(?personA, knows, Jane)* . If Jane is part of another binary triple, in the example, *(?personA, hasIncome, High)* then we form the expression (*?personA,*

[5] Don't confuse a random variable representing the truth value of a statement with a variable in a triple, representing an object.

knows, ?personB) $\wedge$ *(?personB, hasIncome, High)* and count how many rich friends a person has. A large number of additional covariates are possible but so far we restricted ourselves to these two types. The matrix formed with the N statistical units as rows and the covariates as columns is denoted as X and the complete data matrix becomes the matrix $(\alpha X, Y)$.

Covariates are of great importance, in particular if statistical units are rather disconnected. For example, to predict the social status of two professors at different universities in different countries, it might be relevant how many students they administer, but not exactly which students, or it might be important that they are the dean of some department, but not of which department. In social network terms: it might be relevant that they play the same roles.

5. Experiments and Querying Using FOAF Data

5.1. Data Set and Experimental Setup

5.1.1. Data Set

The experiments are based on friend-of-a-friend (FOAF) data. The purpose of the FOAF project [54] is to create a web of machine-readable pages describing people, their relationships, and people's activities and interests, using W3C's RDF technology. The FOAF ontology is based on RDFS/OWL and is formally specified in the FOAF Vocabulary Specification 0.91[6].

We gathered our FOAF data set from user profiles of the community website LiveJournal.com[7]. All extracted entities and relations are shown in Figure 2. In total we collected 32,062 persons and all related attributes. An initial pruning step removed little connected persons and rare attributes. The resulting data matrix, after pruning, has 14,425 rows (persons) and 15,206 columns. Among those columns 14,425 ones (friendship attributes) refer to the property *knows*. The remaining 781 columns (general attributes) refer to general information about age, location, number of blog posts, attended school, online chat account and interest.

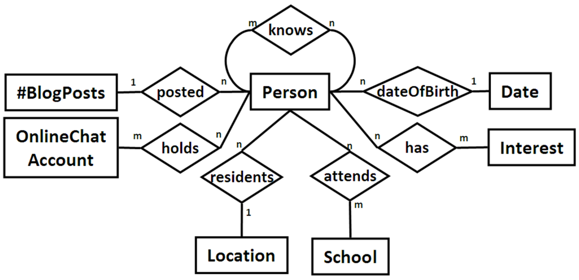


Fig. 2. Entity-relationship diagram of the LJ-FOAF domain

5.1.2. Evaluation Procedure and Evaluation Measure

The task is to predict potential friends of a person, i.e., *knows* statements. For each person in the data set, we randomly selected one *knows* friendship statement and set the corresponding matrix entry to *zero*, to be treated as unknown (test statement). In the test phase we then predicted all unknown friendship entries, including the entry for the test statement. The test statement should obtain a high likelihood value, if compared to the other unknown friendship entries. Here we use the normalized discounted cumulative gain (nDCG) [55] (described in the Appendix) to evaluate a predicted ranking.

5.1.3. Baseline Methods

As a baseline we create a random ranking for all unknown triples, i.e., every unknown triple gets a random probability assigned. Another baseline is Pearson correlation coefficient (PCC) where the prediction for a user x_i is calculated as $\hat{x}_i = \frac{1}{N}\sum_{j \neq i} x_j \frac{<x_j, x_i>}{\|x_j\| \|x_i\|}$ and N is a normalization term.

5.1.4. Data Retrieval and Sampling Strategies

Setting 1 describes the situation where the depicted part of the Semantic Graph is randomly accessible, meaning that all instances can be queried directly from triple stores. Statistical units in the sample for training are randomly sampled and statements for other randomly selected statistical units are predicted for testing (inductive setting). In this setting, persons are rarely connected by the *knows* relations. The *knows* relation in the training and test set is very sparse (0.18%).

Setting 2 also concerns the situation where statistical units in the sample are randomly selected, but this time the truth values of statements concerning the statistical units *in the training sample* are predicted (transductive setting). Some instances of the *knows* relation of the selected statistical units are withheld from training and used for prediction. Prediction should be easier here since the statistics for training and prediction match perfectly.

[6] http://xmlns.com/foaf/spec/

[7] http://www.livejournal.com/bots/

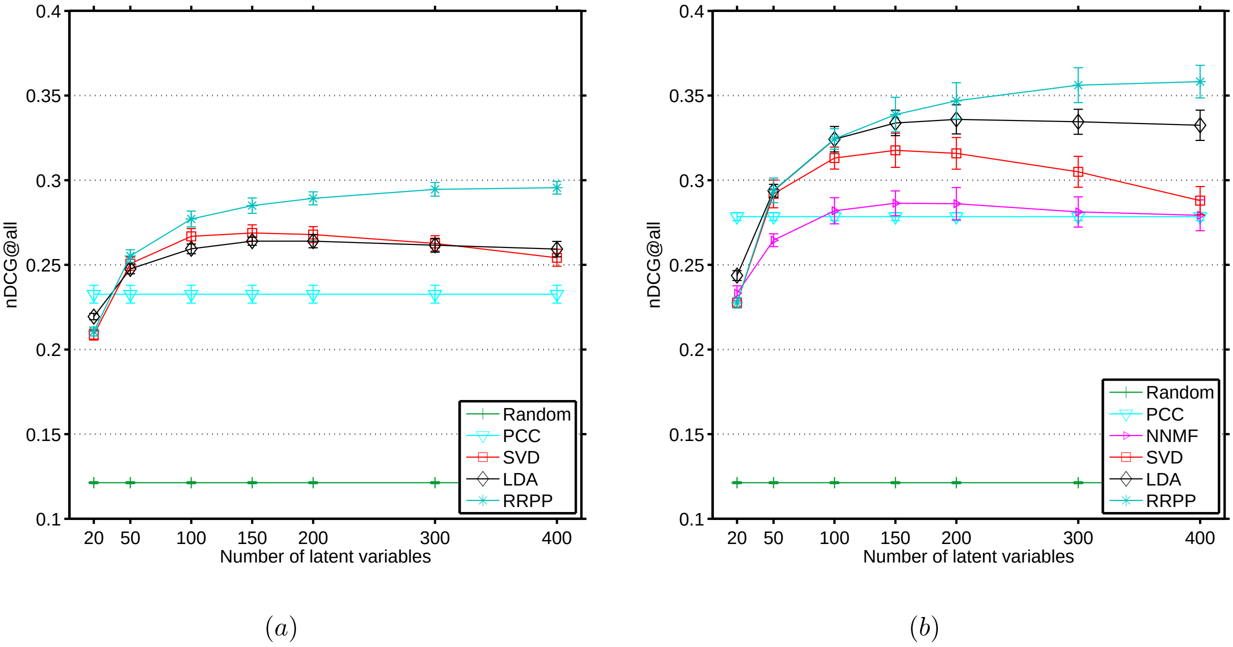


Fig. 3. Comparison between different algorithms. *nDCG all* is plotted against the number of latent variables: (a) and (b) for settings 1 and 2 respectively.

5.2. *Results*

Besides RRPP of the SUNS approach we investigate three other multivariate prediction approaches based on matrix completion, i.e., singular value decomposition (SVD), non-negative matrix factorization (NNMF) [56] and latent Dirichlet allocation (LDA) [57]. All approaches estimate unknown matrix entries via a low-rank matrix approximation. NNMF is a decomposition under the constraints that all terms in the factoring matrices are non-negative. LDA is based on a Bayesian treatment of a generative topic model. After matrix completion of the *zero* entries in the data matrix, the entries are interpreted as certainty values that the corresponding triples are true. After training, the models can also be applied to statistical units in the population outside the sample.

Figure 3 shows the experimental results for our FOAF data set. The error bars show the 95% confidence intervals based on the standard error of the mean over the samples. The figure plots the *nDCG all* score of the algorithms against the number of latent variables. RRPP clearly outperforms the other methods and also the baseline and does not exhibit overfitting with increasing rank. LDA is a Bayesian method which performs second best but overfits with increasing rank. Note that NNMF does not appear in Figure 3 (a), since it is not applicable in a inductive setting.

A more detailed description of the experimental results can be found in [4].

5.3. *Comparison to IHRM*

It is interesting to compare our method to statistical relational models. The IHRM model is an infinite latent-class multi-relational Bayesian learning method and was evaluated in [33] using a FOAF dataset. This data set was also gathered from the website LiveJournal.com. A difference to the data set used in this paper is that there was no information available about *Interest* and rather an attribute expressing whether a user has an image. The details about the data set can be obtained in [33]. We utilize the same data set and the same experimental setting and calculate the area under the ROC curve (AUC) as evaluation measure. When predicting *knows* relations, PPRR achieves $0.9322(\pm 0.003)$ which is clearly better than the AUC score performed by IHRM $0.813(\pm 0.005)$.

5.4. Extended SPARQL and More Sampling Strategies

We want to demonstrate how learned probabilistic statements can be queried. The following SPARQL query illustrates a query for LiveJournal users who live in Munich and might want to be Trelena's friend:

```
PREFIX ya: http://blogs.yandex.ru/schema/foaf
PREFIX foaf: http://xmlns.com/foaf/0.1/
PREFIX dc: http://purl.org/dc/elements/1.1/
SELECT DISTINCT ?person
WHERE {
  ?person ya:located ?city .
  ?person foaf:knows <http://trelana.
      livejournal.com/trelana>
    WITH PROB ?prob .
  FILTER REGEX(?city, "Munich") .
}
ORDER BY DESC(?prob)
```

Listing 1: The query includes the predicted *knows* triples for Trelena and rates them by predicted probability.

Figure 4 shows an output of a knows prediction application.

6. Experiments with DBpedia Data

DBpedia [58] is part of LOD and contains structured information extracted from Wikipedia. At the time of writing this paper, it describes more than 3.4 million concepts, including 312,000 persons, 413,000 places and 94,000 music albums, DBpedia does not only serve as a "nucleus for the web of data", but also holds great potential to be used in conjunction with machine learning approaches. Yet, even though DBpedia already provides a great value, it is still limited in the information it provides and in terms of quality. For example, although there are many cities covered in DBpedia, most information, like its most famous citizens and its most spectacular sights, is not very useful for machine learning purposes. Here we report results using a population consisting of all members of the German Bundestag to evaluate our approach. This population has been created by collecting all triples that are returned by the SPARQL query

```
SELECT ?s ?p ?o WHERE {
  ?s ?p ?o .
  ?s skos:subject dbp-cat:
      Members_of_the_German_Bundestag .
}
```

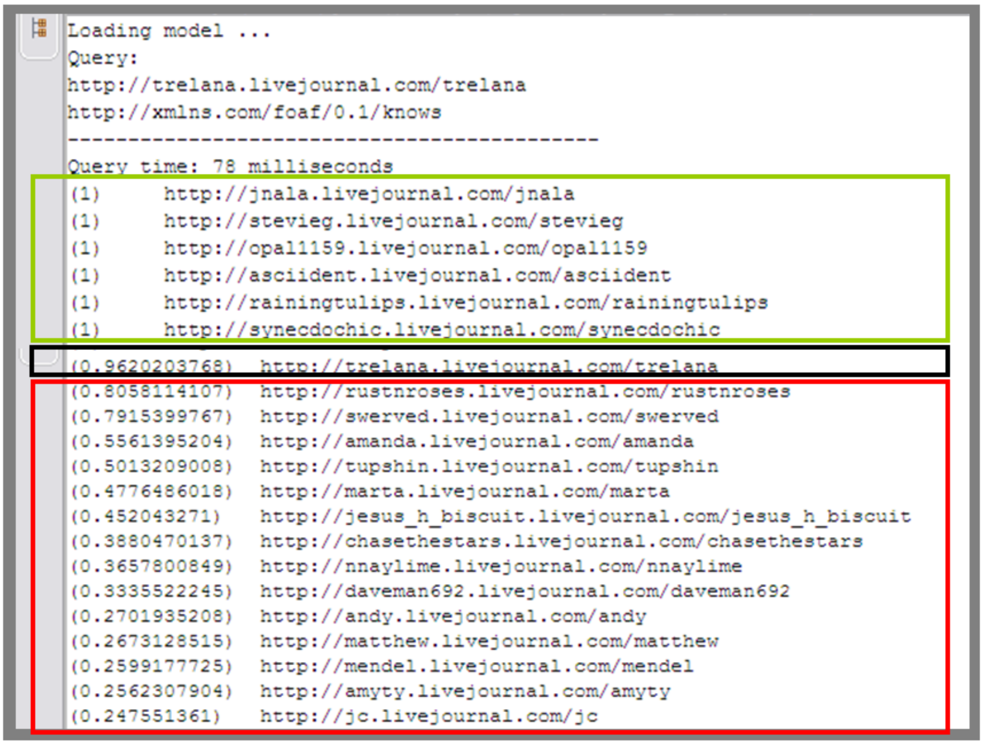
Listing 2: The query generates the population of the German Bundestag.

6.1. Data Quality

A great benefit of LOD data is that by one simple SPARQL query the sample is defined. While DBpedia has great potential for machine learning, there are also challenges when these machine learning approaches are applied to DBpedia data. The first issue is related to the problem of incomplete data. It is very common for subjects in a DBpedia population to share only a subset of predicates. For instance, only 101 of 293 members of the German Bundestag represented in DBpedia have an entry for the predicate `dbp-ont:party` or `dbp-prop:party`. Therefore, in order to handle DBpedia data, a machine learning algorithm has to be able to deal with missing or incomplete data. The second issue is related to noisy predicates. For predicates it is often the case that there are semantical duplicates, e.g. `dbp-prop:party` and `dbp-ont:party`. While duplicate predicates are not a big problem by default, they can become a challenge when they are used inconsistently, which can greatly increase the preprocessing effort. Third, even more serious than noisy predicates are noisy objects. E.g. the Christian Democratic Union of Germany was represented by the literals `"CDU"` and `"Christian Democratic Union"` or the resources `dbpedia:Christian_Democratic_Union` and `dbpedia :Christian_Democratic_Union_(Germany)`. Thus the true members of this party would have been divided into four distinct subsets and this needs to be resolved prior to learning. Finally, we have to consider the size of the dataset, as the sample can get quite large when all available information in DBpedia is used for a population.

6.2. Predicting Party Membership

In the following experiments the learning challenge was to correctly predict the political party for each subject, where the party is identified by the object of the predicate `dbp-prop:party`. Duplicate predicates would bias the experiments as they are heavily cor-



```
Loading model ...
Query:
http://trelana.livejournal.com/trelana
http://xmlns.com/foaf/0.1/knows
--------------------------------------------
Query time: 78 milliseconds
(1)       http://jnala.livejournal.com/jnala
(1)       http://stevieg.livejournal.com/stevieg
(1)       http://opal1159.livejournal.com/opal1159
(1)       http://asciident.livejournal.com/asciident
(1)       http://rainingtulips.livejournal.com/rainingtulips
(1)       http://synecdochic.livejournal.com/synecdochic
(0.9620203768)  http://trelana.livejournal.com/trelana
(0.8058114107)  http://rustnroses.livejournal.com/rustnroses
(0.7915399767)  http://swerved.livejournal.com/swerved
(0.5561395204)  http://amanda.livejournal.com/amanda
(0.5013209008)  http://tupshin.livejournal.com/tupshin
(0.4776486018)  http://marta.livejournal.com/marta
(0.452043271)   http://jesus_h_biscuit.livejournal.com/jesus_h_biscuit
(0.3880470137)  http://chasethestars.livejournal.com/chasethestars
(0.3657800849)  http://nnaylime.livejournal.com/nnaylime
(0.3335522245)  http://daveman692.livejournal.com/daveman692
(0.2701935208)  http://andy.livejournal.com/andy
(0.2673128515)  http://matthew.livejournal.com/matthew
(0.2599177725)  http://mendel.livejournal.com/mendel
(0.2562307904)  http://amyty.livejournal.com/amyty
(0.247551361)   http://jc.livejournal.com/jc
```

Fig. 4. Query: Who wants to be Trelena's friends. Her actual friends are predicted first with confidence values on one (green frame). Then interestingly, it is predicted that she should be her own friend (black frame), followed by a ranked list of predicted friends (red frame).

related with the target predicate. Therefore predicates like `dbp-ont:party` or `dbp-ont:Person/party` were removed. Moreover, predicate-object pairs that are very closely related to a party membership like `(?s, skos:subject, dbp-cat:Politicians_of_the_Social_Democratic_Party_of_Germany)` or `(?s, rdf:type, yago:GermanGreenPartyPoliticians)` were also removed. Rare features were sometimes pruned. In order to demonstrate the aforementioned challenges associated with DBpedia data, we conducted the following experiments

- ORIG: The original data from DBpedia (version 3.5.1). After pruning, this dataset consisted $N = 293$ rows (persons) and 804 columns.
- DISAMB: In this experiment the objects of the target predicate were manually disambiguated to solve the noisy objects problem. After the disambiguation exactly one concept (resource) for each party (CDU, CSU, SPD, FDP, Alliance '90/The Greens, The Left, Centre Party) remained in the data set. Thus, for each statistical unit we estimate $L = 8$ variables. Furthermore, in the original data set only 101 of 293 statistical units had an entry for `dbp-prop:party` `dbp-ont:party`. Since machine learning algorithms benefit from a larger number of examples we manually added the party for the remaining 192 units. After pruning, this data set had 802 columns.
- PEARSON: We used the Pearson correlation coefficient on the disamb data set as a baseline method.
- AGE: In this experiment the age of each politician was added as a continuous feature, by subtracting the birth year (when available) from the year 2010. To prevent that the age values dominated the remaining columns, age values were normalized. After pruning this data set had 804 columns.
- WEIGHT: We used a weighting coefficient of $\alpha = 0.4$ to put less importance on the covariates (see Section 4.2).
- STATE: The predicates `dbp-prop:birthPlace` or `dbp-ont:birthPlace` specify the city or village of birth. For the members with no entry here, we filled in the entry manually. Naturally, the birthplace is not a useful attribute for our task, whereas the state of the birthplace can be quite valuable, since in Germany exist clear local party preferences. Filling in the state information from the

birthplace information can easily be done by exploiting geographical part-of-relationships with OWL reasoning.

- TEXT: Finally associated textual information was exploited by tokenizing the objects of the predicates `rdf:comment` and `dbp-prop:abstract` and by adding one column for each occurring token. When a token was present for a particular statistical unit, the entry was set to one, else to zero. After pruning the data set had 2591 columns.
- ALL: In this experiment all previously described approaches were combined. Since the number of attributes changed, we also changed the weighting factor to $\alpha = 0.2$. After pruning this data set had 2623 columns.

Except for ORIG, the basis for all experiments was the DISAMB data set. To evaluate how well the party membership is predicted, we performed leave-one-out cross-validation by iterating over all subjects. In each iteration we set all `dbp-prop:party` entries for the subject of the current iteration to *zero* and used predicted estimates for ranking. As evaluation measures we used nDCG and bpref [59], the latter often being used in TREC tracks designed for evaluation environments with incomplete relevance data.

Figure 5 and 6 show the results for nDCG and bpref. As expected, the results obtained from the raw data were worst with a score of 0.722. The effect of data cleaning from disambiguation improved the score by 7 points. A small improvement in score can be achieved by adding the age. This shows that age is a weak predictor of party membership, at least in this Bundestag data set. Furthermore, an improvement in score can be achieved by putting more weight on the quantity of interest, i.e., the party membership. The textual description sometimes contains strong hints on party membership and the score improves to 0.928. The state information is also quite relevant as an input, which is well explained by the peculiarities of German politics. Finally, quite a high score of 0.963 is achieved by a combination of all methods.

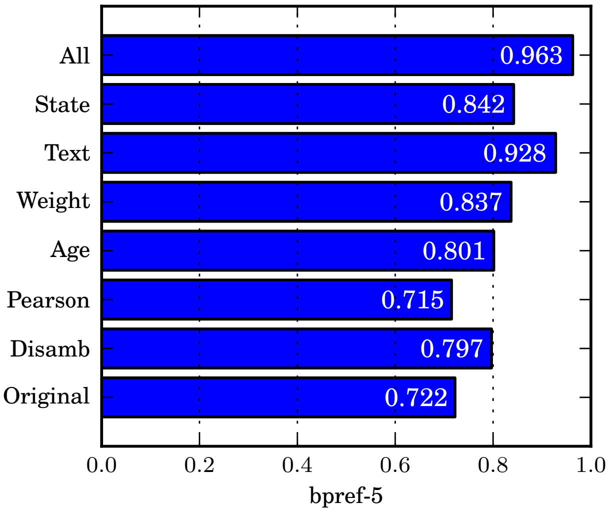


Fig. 5. The bpref scores on the Bundestag population.

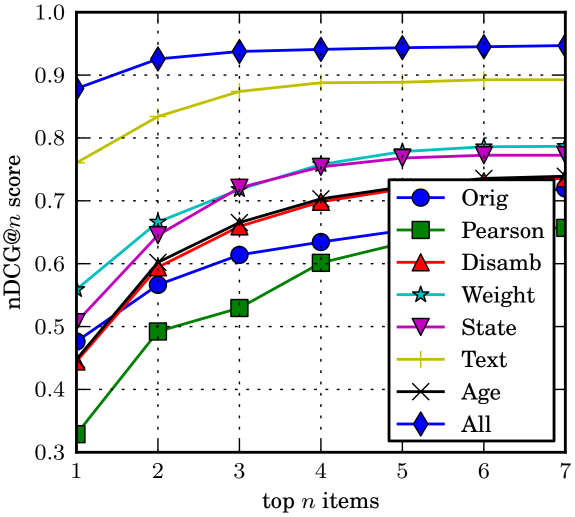


Fig. 6. The nDCG scores on the Bundestag population.

7. Experiments using Linked Life Data

Life science data forms a significant part of the LOD cloud. To a large extent, the life science data has been extracted from well maintained data bases such that this portion of LOD is of high quality. We apply our approach to an important problem in life sciences, i.e., the prediction of gene-disease relationships and show that we obtain competitive results to state-of-the-art solutions.

7.1. Gene-Disease Prediction

Disease genes are those genes involved in the causation of, or associated with a particular disease. At this stage, more than 2500 disease genes have been discovered. Unfortunately, the relationship between genes and diseases is far from simple since most diseases are polygenic and exhibit different clinical phenotypes. High-throughput genome-wide studies like linkage analysis and gene expression profiling typically result in hundreds of potential candidate genes and it is still a challenge to identify the disease genes among them. One reason is that genes can often perform several functions and a mutational analysis of a particular gene reveal dozens of mutation cites that lead to different phenotype associations to diseases like cancer [45]. Analysis is further complicated because environmental and physiological factors come into play as well as exogenous agents like viruses and bacteria.

Despite this complexity, it is quite important to be able to rank genes in terms of their predicted relevance for a given disease as a valuable tool for researchers and with applications in medical diagnosis, prognosis, and a personalized treatment of diseases. During the last years, a number of solutions and tools have been developed for this purpose, such as ToppGene [47]. Those tools use features and gene similarity measures to generalize from known disease genes to others. Kann [45] provides an up-to-date review on the latest advances in the field of translational bioinformatics, focusing on the advances of computational techniques to search for and classify disease genes.

Certain gene properties differentiate disease genes and have been used as the basis for computational tools to prioritize disease gene candidates derived from these experiments. All of the current approaches are based on the integration of different sources such as: gene function (disease genes are expected to share common functional properties), pathways (disease genes are most likely to share common pathways), gene expression (disease genes are expected to be co-expressed), gene regulation (genes within the same gene-regulation network are expected to affect similar diseases), sequence properties, and protein interaction (disease genes are often highly connected with other genes from the same disease).

The quality of the available experimental data is a major limitation of the gene-prioritization techniques. For instance, protein-protein interaction-based methods suffer from the incompleteness and low quality of the data currently available for interaction networks in mammals.

In the following we will compare our system with the ToppGene Suite, which is one of the state-of-the-art approaches with an easy to use interface [47] and uses a fuzzy similarity measure between genes.

7.2. Gene-Disease Data

A great benefit of LOD is that the data required for the experiments, i.e., gene-disease relationships, gene attributes, and disease attributes, can be extracted without much effort. For gene-disease relationships manually curated and well maintained data bases were used. In particular, we used OMIM [60], UniProt [61], PharmGKB [62] and CTD [63]. All of these databases except UniProt use Entrez Gene [64] identifiers for genes. However, the mapping of UniProt to Entrez Gene is complete enough to use only Entrez Gene identifiers as the gene vocabulary. Unfortunately this doesn't apply to diseases. Some of the databases use OMIM terms while others use MeSH. The mapping of OMIM to MeSH or vice versa is non-trivial and subject of ongoing research (e.g. see [65]). For this reason only data with MeSH identifiers have been used. After the gene-disease relations had been extracted, this information was stored in an RDF triple store, by creating triples of the kind `(<Entrez Gene ID>, siemens: related_to, <MeSH ID>)`.

In the next step we retrieved data for the gene attributes. We used the attributes mentioned in a recent review article [45]. For instance, we accessed BioGRID [66] and HPRD [67] which provide protein interaction information. UniProt [61] and Reactome [68] provide data about pathways and Gene Ontology [69] provides gene function annotations.

Again it was quite helpful that the relevant data sources were available in the LOD cloud and were as such easily accessible. In particular, Bio2RDF [69] and Linked Life Data (LLD)[8] are two projects from the Semantic Web and the Linked Data community that integrate various bioinformatic databases and publish this information in the form of RDF triples. At the time of this writing the Bio2RDF website[9] lists 57 different databases. Figure 7 shows the various databases and how they are connected.

In Bio2RDF, resources are accessible via a REST-like interface and identified by normalized URIs of the form `http://bio2rdf.org/<namespace:id>`. Therefore, for each Entrez Gene ID we queried the Bio2RDF with the appropriate identifier, retrieved all available information as RDF triples and added them to the existing gene-disease graph. In doing so, we were able to derive attributes for protein interaction by using data from e.g. BioGRID or HPRD, gene function by us-

[8] http://linkedlifedata.com

[9] http://bio2rdf.wiki.sourceforge.net/

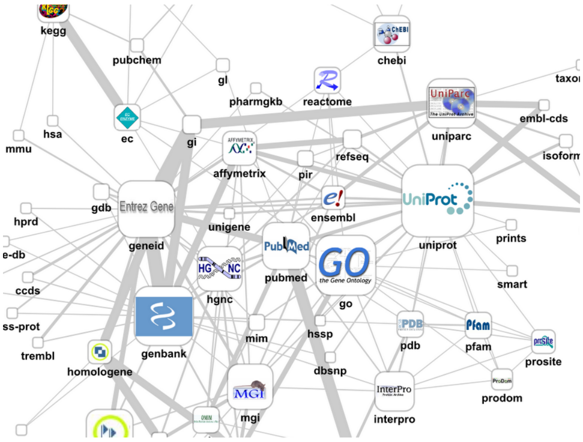


Fig. 7. Bio2RDF databases and connections. (Source http://bio2rdf.wiki.sourceforge.net)

ing Gene Ontology annotations, Pathway through data from Pathway Commons, UniProt or Reactome. Moreover, we included information about the protein domain from CDD [70] as well as PubMed co-citations. The gene length was added in a postprocessing step as a normalized continuous attribute. Therefore, most of the attributes that have been identified in [45] as important for gene-disease priorization could be retrieved by one single request to Bio2RDF and one postprocessing step for any gene in the gene-disease relation graph.

Finally, we gathered data for disease attributes. As mentioned above we used only MeSH diseases in our experiments for gene-disease relationship prediction. Their attributes are neither available as easily nor in any RDF format. We crawled the MeSH tree structure of the diseases via MeSH Browser (2010 MeSH)[10]. Then, we split tree numbers according to the levels of the MeSH hierarchy from the top one to the leafs. For instance, the disease “abetalipoproteinaemia” (mesh:D000012) is located in three nodes in the hierarchy and one of them is C16.320.565. Therefore, we form the attributes C16, C16.320 and C16.320.565. In this way we obtained for 1138 MeSH diseases totally 4389 attributes which are arranged in a 10-level hierarchy.

The gene-disease data set is available at `http://www.dbs.ifi.lmu.de/~huang/index.html#datasets`.

7.3. The Data Matrices

We compared two experiments. In the first experiment, the genes were treated as statistical units and in the second experiment the diseases were treated as statistical units. In the first experiment the data matrix consisted of the $N \times M$ data matrix Y where the rows of Y represent the N genes and the columns of Y represent the M diseases. The element $y_{i,j}$ is equal to *one* if it is known that gene i affects disease j and is *zero* otherwise. We explored $N = 3820$ genes and $M = 3102$ diseases, 1138 of which are MeSH terms. Y is very sparse and contains only 0.07 % *ones* where the MeSH part of Y has 0.13% *ones*. In addition, the data matrix contains the matrix X_G with the attributes describing the genes. In the raw data, we obtained almost a million attributes. This number was reduced to less than 100000 after cutting the attributes that are associated with only one gene. X_G is very sparse as well, filled with 0.13% *ones*.

In the second experiment, the diseases were treated as statistical units and the data matrix consisted of the Y^T. In addition, the data matrix contains the matrix X_D with the attributes describing the diseases. We used 4389 disease attributes. Only 0.28% of the entries of X_D are *ones*.

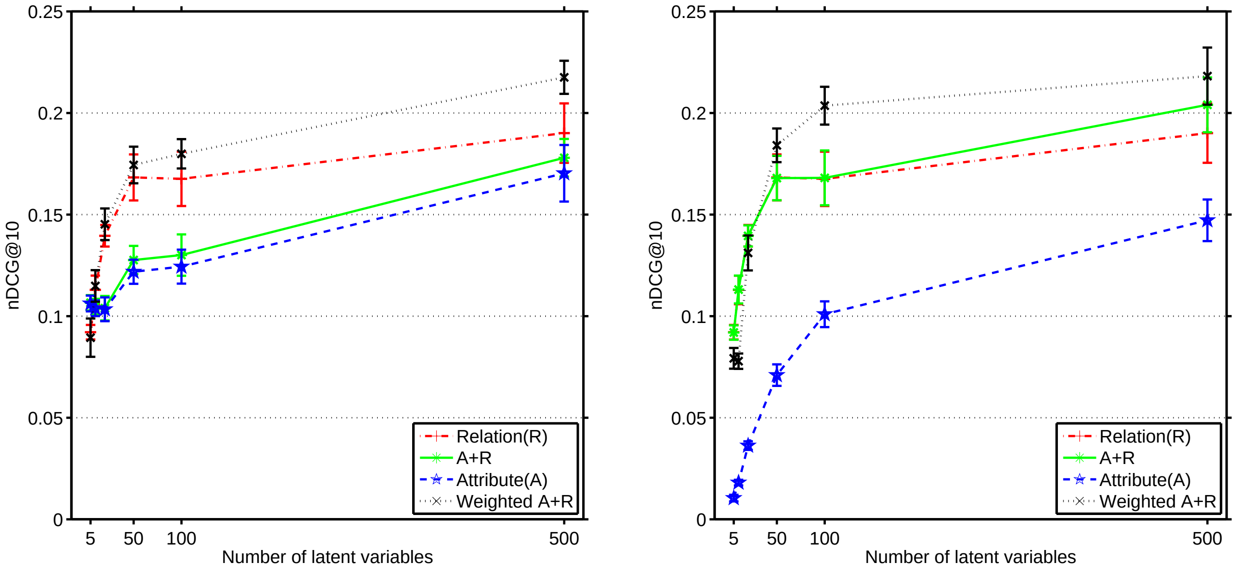
7.4. Experiments on Gene-Disease Modeling

7.4.1. Evaluation

In the experiments we removed a known gene-disease relationship for each gene (i.e., we set the entry to *zero*) and predicted the likelihood of all *zero* entries. We then evaluated how the known relationship was ranked in comparison to the other zeros using the nDCG@n score with $n = 10$ (the nDCG score is described in the Appendix). We repeated the procedure 5 times to be able to produce error bars and mean values.

Figure 8 shows the nDCG score against rank r for the basis SUNS models on the gene data set (a) and the disease data set (b). In both experiments an attribute-based prediction (only using X_G, resp. X_D for the kernel) gave worst results (att) (regularized PCA regression of Section 4.2). Models using Y for the kernel performed much better (rel) (regularized matrix reconstruction using PCA of Section 4.2). When both Y and X_G, resp. X_D were used with $\alpha = 1$, the results were suboptimal, due to the dominance of the attributes. Best performance could be achieved by carefully tuning α. Due to the large number of attributes, in the first experiment a small $\alpha = 0.1$ gave best results whereas

[10] http://www.nlm.nih.gov/mesh/MBrowser.html



(a) gene data set

(b) disease data set

Fig. 8. (a) The nDCG@10 score against rank r for the SUNS model where the statistical units were the genes. The bottom (blue, att) line shows the performance of the SUNS model using only X_G for the kernel and the second line from the top (red, rel) shows the performance of the SUNS model using Y for the kernel. We can see that the relationships were better predictors. An RRPP SUNS model with $\alpha = 1$ (second line from bottom, green, att+rel) was worse than the relationship model alone. An RRPP SUNS model with $\alpha = 0.1$ gave best results. (b) The same but for the SUNS model where the statistical units were the diseases. At high rank r the attributes were also quite informative. An RRPP SUNS model with $\alpha = 0.5$ gave best results, when rank r exceeded 50.

in the second experiment, we used $\alpha = 0.5$. With an optimally tuned α both choices of statistical units resulted in similar performances (RRPP). An interesting question is if it is beneficial to combine both models. Figure 9 shows the nDCG score against rank r for the multi-population SUNS models in which the prediction of both models are simply averaged. With RRPP SUNS models as components (b), we got overall best performance by simply averaging predictions.[11]

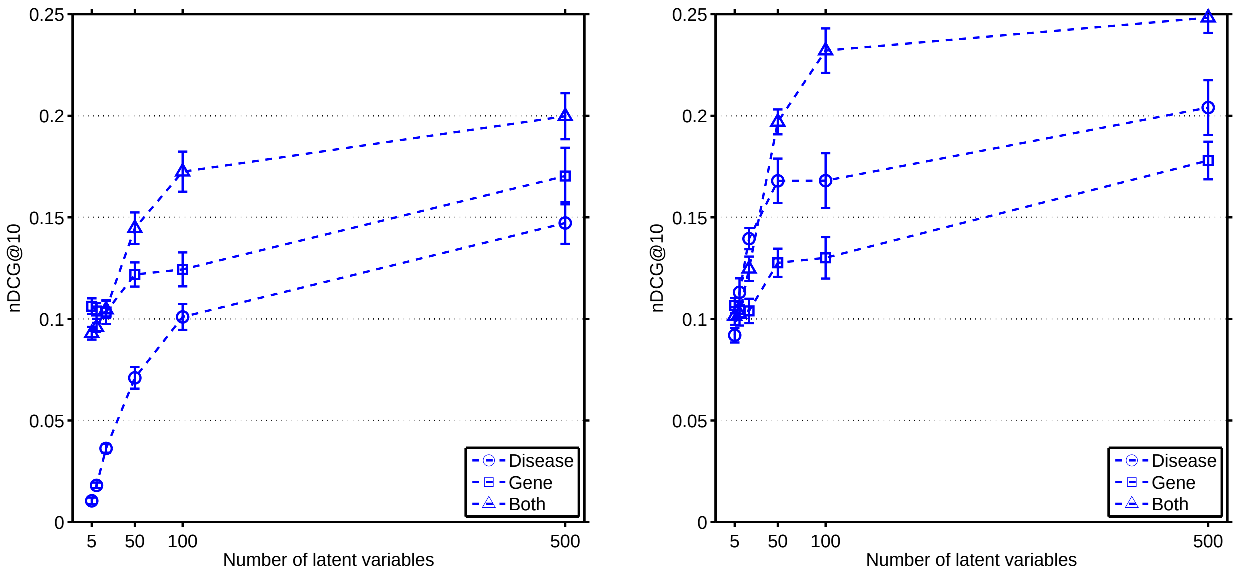
7.4.2. Comparison with ToppGene

In order to compare our approach with other existing gene prioritization tools, we have selected the state-of-the-art service ToppGene. Therefore the same evaluation procedure as described in [71] and [72] has been employed. First, various diseases have been selected from the OMIM and MeSH databases together with their related genes. The number of related genes ranges from 18 for Hypertension to 47 for Asthma. Additionally, 99 random, but unrelated genes have been selected for each disease. For each disease the following validation procedure has been applied: Perform leave-one-out validation for each related gene by removing this gene (the "target" gene) from the training set (ToppGene) or setting the target relation to *zero*. Then, a ranking of the 99 random genes combined with the target gene is computed. Hereby, sensitivity was defined as the frequency of the target genes that are ranked above a given threshold (e.g. in the top 10% of the results) and specificity as the frequency of genes that are ranked below the threshold. Since we used data different from the original ToppGene paper, we submitted all training and test data manually through the ToppGene web interface[12] and recorded the ranking. All ToppGene results have been computed with the standard parameters suggested by the interface.

Table 1 shows the evaluation results of SUNS and ToppGene on 4 different diseases. Each number represents an average over up to 50 genes. We can see that in 7 out of 12 cases SUNS was superior to ToppGene. In Psoriasis ToppGene is superior for top 20%. We have observed a trend that for higher percentage

[11] Different weighting factors did not improve with respect to simple averaging.

[12] http://toppgene.cchmc.org/prioritization.jsp



(a) attribute only

(b) attribute and relationship

Fig. 9. (a) The nDCG@10 score against rank r for the multi-population SUNS model with equal weights and the SUNS models using X_G, resp. X_G for calculating the kernel. We see that the multi-population model gave best results. (b) The same but for the RRPP SUNS models as components. The multi-population model gave overall best results.

Table 1

Validation results of SUNS and ToppGene. The best results for each disease are printed bold.

	Top 5%	Top 10%	Top 20%
Autistic Disorder (mesh:D001321)			
ToppGene	0.20	0.26	0.36
SUNS	**0.36**	**0.40**	**0.46**
Psoriasis (mesh:D011565)			
ToppGene	0.44	0.52	**0.92**
SUNS	**0.64**	**0.68**	0.72
Hypertension, essential (omim:145500)			
ToppGene	0.61	0.83	1.0
SUNS	**0.89**	**1.0**	1.0
AIDS, delayed/rapid progression to (omim:609423)			
ToppGene	**1.0**	**1.0**	1.0
SUNS	0.91	0.91	1.0

values, ToppGene tends to be strong. We believe that these results are quite promising, considering that in ToppGene the weights on the relevance of the different attributes were carefully adjusted whereas in SUNS, all attributes essentially have the same weights. We see a good chance that with a careful weighting of attributes, our approach can even perform better.

8. Further Applications and Conclusions

This paper discussed approaches towards statistical machine learning in semantic graphs. We proposed the SUNS approach as a flexible, robust and scalable approach. In the typical data situation with many columns in the data matrix, we proposed a kernel approach. We demonstrated that the SUNS approach can easily be applied to a large number of learning tasks.

First, we presented results from a social network domain and we showed how learned information can be retrieved using an extended SPARQL query.

Secondly, we applied the approach to data extracted from DBpedia. Here the data is quite noisy and considerable preprocessing is needed to yield good results. Also, by including textual data the prediction results were considerably improved. This improvement can already be observed even if a simple keyword based representation is used without any sophisticated information extraction. Some of the data preprocessing steps can easily be executed with ontological (OWL-) reasoning, such as the generalization from city to state. In fact, materialization of facts derivable from logical reasoning is recommended as a preprocessing step. Other preprocessing steps, such as the calculation of age from the birthday and the current date, were done algorithmically.

In a third and novel application, we studied a life science domain with several hundred thousand covariates, demonstrating the scalability of the approach. We achieved excellent performance in gene prioritization.

The presented approach was also used in cooperations with other research teams. In [73] the SUNS approach is applied for item recommendations in the context of RDF stream reasoning. In contributions to the MEDICO use case in THESEUS[13] the SUNS model is used as a learning decision system for predicting procedures and diagnoses for a given patient. The recommendation is based on a data set of previously collected patient histories.

Scalability of the overall approach is guaranteed. First, we can control the number of instances considered in the Nyström approximation. Second we can control the rank of the approximation. Third, we can control the number of local features that are used to derive the kernel. In our experiments, M, the number of features, was always quite high. In this case the most costly computation is the calculation of the kernel requiring N^2M operations.

LOD data provides a great potential for applying machine learning approaches on that. Besides those examples in this paper, recently we investigated the application of a tensor factorization approach to predicting the `rdf:type` relation (e.g. `dbpedia-owl:Person`) in [38] and also the application of a combined learning approach to prediction the writer's nationality (e.g. `yago:FrenchWriters`) in [74], when textual data is included. In the last paper it is also discussed how a SUNS-like model can be extended to model a complete LOD data base with many types of entities and relationships.

Acknowledgements

We acknowledge funding by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project and by the EU FP 7 Large-Scale Integrating Project LarKC.

9. Appendix

Details on the nDCG Score

We use the normalized discounted cumulative gain (nDCG) to evaluate a predicted ranking. The nDCG score is calculated by summing over all the gains in the rank list R with a log discount factor as

$$\text{nDCG}(R) = \frac{1}{Z}\sum_k \frac{2^{r(k)}-1}{\log(1+k)},$$

where $r(k)$ denotes the target label for the k-th ranked item in R, and r is chosen such that a perfect ranking obtains value 1. To focus more on the top-ranked items, we also consider the *nDCG@n* which only counts the top n items in the rank list. These scores are averaged over all ranking lists for comparison.

References

[1] Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) (2009)

[2] Tauberer, J.: Resource Description Framework, http://rdfabout.com/

[3] Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In: Proceedings of the First ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web. (2009)

[13] http://theseus-programm.de/

[4] Huang, Y., Bundschus, M., Tresp, V., Rettinger, A., Kriegel, H.P.: Multivariate structured prediction for learning on the Semantic Web. In: Proceedings of the 20th International Conference on Inductive Logic Programming (ILP). (2010)

[5] Huang, Y., Nickel, M., Tresp, V., Kriegel, H.P.: A scalable kernel approach to learning in semantic graphs with applications to linked data. In: 1st Workshop on Mining the Future Internet. (2010)

[6] Momtchev, V., Peychev, D., Primov, T., Georgiev, G.: Expanding the pathway and interaction knowledge in Linked Life Data. In: Proceedings of International Semantic Web Challenge. (2009)

[7] Kiefer, C., Bernstein, A., Locher, A.: Adding data mining support to SPARQL via statistical relational learning methods. In: Extended Semantic Web Conference 2008, Springer-Verlag (2008)

[8] Raedt, L.D., Jaeger, M., Lee, S.D., Mannila, H.: A theory of inductive query answering. In: Proceedings of the IEEE International Conference on Data Mining. (2002)

[9] Getoor, L., Friedman, N., Koller, D., Pferrer, A., Taskar, B.: Probabilistic relational models. In Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. MIT Press (2007)

[10] Domingos, P., Richardson, M.: Markov logic: A unifying framework for statistical relational learning. In Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. MIT Press (2007)

[11] Xu, Z., Tresp, V., Yu, K., Kriegel, H.P.: Infinite hidden relational models. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI). (2006)

[12] Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). (2006)

[13] Rettinger, A., Nickles, M., Tresp, V.: Statistical relational learning of trust. Machine Learning (2010)

[14] Quinlan, J.R.: Learning logical definitions from relations. Machine Learning **5**(3) (1990)

[15] Muggleton, S., Feng, C.: Efficient induction of logic programs. In: Proceedings of the 1st Conference on Algorithmic Learning Theory, Ohmsma, Tokyo (1990)

[16] De Raedt, L.: Attribute-value learning versus inductive logic programming: The missing links (extended abstract). In: ILP '98: Proceedings of the 8th International Workshop on Inductive Logic Programming, Springer-Verlag (1998)

[17] Lavrač, N., Džeroski, S., Grobelnik, M.: Learning nonrecursive definitions of relations with LINUS. In: EWSL-91: Proceedings of the European Working Session on Machine Learning. (1991)

[18] Lisi, F.A., Esposito, F.: An ILP perspective on the Semantic Web. In: Semantic Web Applications and Perspectives. (2005)

[19] Tresp, V., Yu, K.: Learning with dependencies between several response variables. In: Tutorial at the International Conference on Machine Learning. (2009)

[20] Cumby, C.M., Roth, D.: On kernel methods for relational learning. In: Proceeding of the International Conference on Machine Learning. (2003)

[21] Vishwanathan, S.V.N., Schraudolph, N., Kondor, R.I., Borgwardt, K.: Graph kernels. Journal of Machine Learning Research - JMLR (2008)

[22] Gärtner, T., Lloyd, J., Flach, P.: Kernels and distances for structured data. Machine Learning **57**(3) (2004)

[23] Lösch, U., Bloehdorn, S., Rettinger, A.: Graph kernels for RDF data. In: Proceedings of the 9th Extended Semantic Web Conference (ESWC'12), Springer (2012)

[24] Zhu, X.: Semi-supervised learning literature survey. Technical report, Computer Sciences TR 1530 University of Wisconsin Madison (2006)

[25] Yu, K., Chu, W., Yu, S., Tresp, V., Xu, Z.: Stochastic relational models for discriminative link prediction. In: Proceedings of Advances in Neural Information Processing Systems. (2006)

[26] Xu, Z., Kersting, K., Tresp, V.: Multi-relational learning with gaussian processes. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). (2009)

[27] Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. In: Proceedings of Advances in Neural Information Processing Systems. (2003)

[28] Muggleton, S., Lodhi, H., Amini, A., Sternberg, M.J.E.: Support vector inductive logic programming. In Hoffmann, A., Motoda, H., Scheffer, T., eds.: Proceedings of the 8th International Conference on Discovery Science. Volume 3735 of LNCS., Springer (2005)

[29] Landwehr, N., Passerini, A., De Raedt, L., Frasconi, P.: kFOIL: Learning simple relational kernels. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). (2006)

[30] d'Amato, C., Fanizzi, N., Esposito, F.: Non-parametric statistical learning methods for inductive classifiers in semantic knowledge bases. In: Proceedings of the IEEE International Conference on Semantic Computing - ICSC 2008. (2008)

[31] Popescul, A., Ungar, L.H.: Statistical relational learning for link prediction. In: Workshop on Learning Statistical Models from Relational Data. (2003)

[32] Getoor, L., Diehl, C.P.: Link mining: a survey. SIGKDD Explorations (2005)

[33] Rettinger, A., Nickles, M., Tresp, V.: Statistical relational learning with formal ontologies. In: Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD). (2009)

[34] Sutskever, I., Salakhutdinov, R., Tenenbaum, J.B.: Modelling relational data using bayesian clustered tensor factorization. In: Advances in Neural Information Processing Systems. (2009)

[35] Franz, T., Schultz, A., Sizov, S., Staab, S.: Triplerank: Ranking Semantic Web data by tensor decomposition. In: 8th International Semantic Web Conference(ISWC2009). (2009)

[36] Bro, R.: PARAFAC. Tutorial and applications. Chemometrics and Intelligent Laboratory Systems **38** (1997) 149–171

[37] Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning. (2011)

[38] Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing Yago: Scalable machine learning for linked data. In: Proceedings of the 21st International World Wide Web Conference. (2012)

[39] Maedche, A., Staab, S.: Ontology Learning. In: Handbook on Ontologies 2004. Springer (2004)

[40] Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL: Concept learning in description logics. In: Proceedings of the International Conference on Inductive Logic Programming. (2008)

[41] Lehmann, J.: DL-learner: Learning concepts in description logics. Journal of Machine Learning Research - JMLR (2009)

[42] Berendt, B., Hotho, A., Stumme, G.: Towards Semantic Web mining. In: ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web. (2002)

[43] Mika, P.: Social Networks and the Semantic Web. Springer (2007)

[44] Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the Semantic Web - Statistical learning for next generation knowledge bases. Data Mining and Knowledge Discovery (2012)

[45] Kann, M.G.: Advances in translational bioinformatics: computational approaches for the hunting of disease genes. Briefings in Bioinformatics **11**(1) (2010) 96

[46] Radivojac, P., Peng, K., Clark, W.T., Peters, B.J., Mohan, A., Boyle, S.M., Mooney, S.D.: An integrated approach to inferring gene-disease associations in humans. Proteins **72** (2008)

[47] Chen, J., Bardes, E.E., Aronow, B.J., Jegga, A.G.: ToppGene suite for gene list enrichment analysis and candidate gene prioritization. Nucleic Acids Research **37** (2009)

[48] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web. (2007)

[49] Strube, M., Ponzetto, S.P.: WikiRelate! computing semantic relatedness using Wikipedia. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2006)

[50] de Salvo Braz, R., Amir, E., Roth, D.: Lifted first-order probabilistic inference. In Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. MIT Press (2007)

[51] Neville, J., Jensen, D.: Bias/variance analysis for relational domains. In: Proceedings of the International Workshop on Inductive Logic Programming - ILP. (2007)

[52] Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems 13. (2001)

[53] Tresp, V., Bundschus, M., Rettinger, A., Huang, Y.: Towards Machine Learning on the Semantic Web. In: Uncertainty Reasoning for the Semantic Web I. Lecture Notes in AI, Springer (2008)

[54] Brickley, D., Miller, L.: The Friend of a Friend (FOAF) project, http://www.foaf-project.org/

[55] Jarvelin, K., Kekalainen, J.: IR evaluation methods for retrieving highly relevant documents. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. (2000)

[56] Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature (1999)

[57] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003)

[58] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. The Semantic Web (2008)

[59] Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Proceedings of the 27th ACM SIGIR Conference on Research and Development in Information Retrieval. (2004)

[60] Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A., McKusick, V.A.: Online mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research **33**(Database Issue) (2005) D514

[61] Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al.: UniProt: the universal protein knowledgebase. Nucleic acids research **32**(Database Issue) (2004) D115

[62] Hewett, M., Oliver, D.E., Rubin, D.L., Easton, K.L., Stuart, J.M., Altman, R.B., Klein, T.E.: PharmGKB: the pharmacogenetics knowledge base. Nucleic Acids Research **30**(1) (2002) 163

[63] Mattingly, C., Rosenstein, M., Colby, G., Forrest Jr, J., Boyer, J.: The comparative toxicogenomics database (CTD): a resource for comparative toxicological studies. Journal of Experimental Zoology. Part A, Comparative Experimental Biology **305**(9) (2006) 689

[64] Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez gene: gene-centered information at NCBI. Nucleic acids research (2006)

[65] Isaac, K., Ming-Chin, L., Lewis, F.: Metadata mapping and reuse in caBIG. BMC Bioinformatics **10**

[66] Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucleic acids research **34**(Database Issue) (2006) D535

[67] Peri, S., Navarro, J.D., Kristiansen, T.Z., Amanchy, R., Surendranath, V., Muthusamy, B., Gandhi, T.K.B., Chandrika, K.N., Deshpande, N., Suresh, S., et al.: Human protein reference database as a discovery resource for proteomics. Nucleic acids research **32**(Database Issue) (2004) D497

[68] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., Bono, B.D., Jassal, B., Gopinath, G.R., Wu, G.R., Matthews, L., et al.: Reactome: a knowledgebase of biological pathways. Nucleic acids research **33**(Database Issue) (2005) D428

[69] Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P., Morissette, J.: Bio2RDF: towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics **41**(5) (2008)

[70] Marchler-Bauer, A., Anderson, J.B., Cherukuri, P.F., DeWeese-Scott, C., Geer, L.Y., Gwadz, M., He, S., Hurwitz, D.I., Jackson, J.D., Ke, Z., et al.: CDD: a conserved domain database for protein classification. Nucleic acids research **33**(Database Issue) (2005) D192

[71] Chen, J., Xu, H., Aronow, B., Jegga, A.: Improved human disease candidate gene prioritization using mouse phenotype. BMC Bioinformatics **8**(1) (2007) 392

[72] Aerts, S., Lambrechts, D., Maity, S., Loo, P.V., Coessens, B., Smet, F.D., Tranchevent, L., Moor, B.D., Marynen, P., Hassan, B., Carmeliet, P., Moreau, Y.: Gene prioritization through genomic data fusion. Nature Biotechnology **24**(5) (May 2006) 544, 537

[73] Barbieri, D., Braga, D., Ceri, S., Valle, E.D., Huang, Y., Tresp, V., Rettinger, A., Wermser, H.: Deductive and inductive stream reasoning for semantic social media analytics. IEEE Intelligent Systems **99** (2010)

[74] Jiang, X., Huang, Y., Nickel, M., Tresp, V.: Combining information extraction, deductive reasoning and machine learning for relation prediction. In: Proceedings of the 9st Extended Semantic Web Conference. (2012)