Knowledge Discovery meets Linked APIs

Julia Hoxha[1], Maria Maleshkova[1], and Peter Korevaar[2]

[1]Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
{julia.hoxha,maria.maleshkova}@kit.edu
[2]Institute KSRI, Karlsruhe Institute of Technology (KIT), Germany
peter.korevaar@partner.kit.edu

Abstract. Knowledge Discovery and Data Mining (KDD) is a very well-established research field with useful techniques that explore patterns and regularities in large relational, structured and unstructured datasets. Theoretical and practical development in this field have led to useful and scalable solutions for the tasks of pattern mining, clustering, graph mining, and predictions. In this paper, we demonstrate that these approaches represent great potential to solve a series of problems and make further optimizations in the setting of Web APIs, which have been significantly increasing recently. In particular, approaches integrating Web APIs and Linked Data, also referred to as Linked APIs, provide novel opportunities for the application of synergy approaches with KDD methods.
We give insights on several aspects that can be covered through such synergy approach, then focus, specifically, on the problem of API usage mining via statistical relational learning. We propose a Hidden Relational Model, which explores the usage of Web APIs to enable analysis and prediction. The benefit of such model lies on its ability to capture the relational structure of API requests. This approach might help not only to gain insights about the usage of the APIs, but most importantly to make active predictions on which APIs to link together for creating useful mashups, or facilitating API composition.

1 Introduction

Knowledge discovery is an interdisciplinary area that focuses on methodologies for identifying novel, potentially useful and meaningful patterns from data. Data mining is an important part of the field. The rapid growth of data on the Web and the widespread use of large databases have resulted in an increased demand for knowledge discovery and data mining (KDD) methods.

At the same time, the Web of Data has grown to one of the largest publicly available collections of structured data, spurred by the Linked Open Data initiative[1]. A more dynamic way to access such data is through APIs, which can provide access to a wealth of up-to-date information [12]. Recent approaches [13, 18,20] have investigated the integration of Web APIs, or data-providing services, with Linked Data, referred here as Linked APIs. The overall goal is to enable

[1] http://linkeddata.org/

the communication of APIs at a semantic level, so that they can consume and produce Linked Data. Table 1 illustrates an example of the invocation of the original GeoNames API[2], which finds nearest GeoNames feature to a given point and links a geographic point to nearby resources from DBpedia. The API is invoked via a HTTP request. If exposed as Linked API, its request would be in the Linked Data format, mapped to public vocabularies and serialized in RDF. The same goes for the response retrieved by the API invocation.

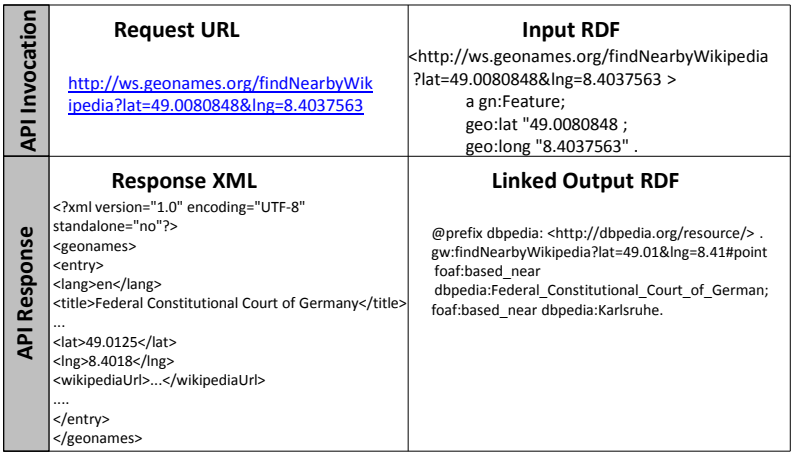

Fig. 1. An example of Web API request and response in different formats

As in the general case where the growing information on the Web necessitates the application of knowledge discovery, we argue that the setting of Linked APIs also demands such techniques to tackle a series of open problems. In this position paper, we aim to investigate the potential of a synergy between KDD methods, on one hand, and research on Web APIs and Linked Data integration, on the other hand. Based on state-of-the-art approaches and new insights, we discuss (1) how KDD methods can tackle problems related to Linked APIs, and (2) how Linked APIs can be used to leverage existing KDD methods. Our goal is to stimulate the interest of both communities to explore novel approaches for mutual research.

2 Synergy Approach

We discuss how KDD methods can tackle problems related to Linked APIs, and how Linked APIs can be used to leverage existing KDD methods. There are two main questions that we address: (1) *how can the KDD methods be leveraged and pushed forward using contributions from Linked APIs*, and (2) *how can we tackle the problems and main research questions of Linked APIs by applying KDD methods*?

[2] http://www.geonames.org/

In the following sections, we propose a few areas where a synergy is promising in each of these two directions.

2.1 Linked APIs for KDD

In the past few years, there has been a growing realization in the research community of data mining and knowledge discovery that semantics and structure can greatly enhance existing methods by boosting their performance. This issue has been investigated in research areas such as, among others, search and information retrieval, Web mining, recommender systems and social network analysis. While this field is very large and of high variety, we select here only a few concrete topics we think are promising and elaborate on some ideas how KDD methods can be enhanced via the use of Linked APIs.

Federated Search. Currently, search on the Web is going beyond the retrieval of textual Web sites, taking advantage of the growing amount of structured data. As an application of the broad field of information retrieval (IR), *federated search* allows users to submit a real-time search in parallel to multiple, distributed information sources and retrieve aggregated, ranked and deduplicated results. Recent focus of IR research has been entity search [2, 6, 17], where the units of retrieval are structured entities instead of textual documents. These entities reside in different sources and, instead of having a centralized solution, an investigated approach is to directly search entities over distributed data sources.

An interesting research ground would be the investigation of federated search, e.g. federated entity discovery, over distributed Web APIs. If the IR community shifts the interest on these APIs, they can find the needed setting where abudant data is offered in structured format and in distributed sources. From a research point of view, in such a setting they can address the problems of data completeness, ranking, or information redudancy via on-the-fly entity consolidation techniques.

Pattern Mining. Several works [1, 9, 10, 19, 25] investigate the effect of semantic information on mining frequent patterns. The common idea behind this research is to enable semantic (association) pattern mining based on ontology knowledge representation. The goal is to let machines provide the capability of understanding the semantics of text data, and learning and reasoning automatically. Generally, they indicate an increase in pattern quality when patterns are semantically enriched.

While the results that they show are very promising, it is noticable that these approaches are generally based on small datasets and toy or quite small, manually-built ontologies. This is an issue raised very recently within the data mining community [23], which addresses the problem of the lack of interesting large datasets. Most of their research is based on on old, solid evaluation benchmarks, but less compelling Big Data. On the other side, the Web of Data has grown to one of the largest publicly available collections of structured, cross-domain data sets. In our opinion, one very suitable way to reach these datasets is through APIs. As such, research community of data mining, in general, and of

pattern mining specifically, can greatly profit by getting more acquainted with the setting of Linked Data and Web APIs, whose scale and heterogeneity would certainly pose research challenges and novel contributions. The same idea applies also to social network mining, where current analytic approaches can be leveraged with data requested by social graph APIs, e.g. Facebook's Graph API to explore linked objects and connections in Facebook's social graph [22]. Current interesting theoretical works on social network mining with probabilistic relational models, which work with small datasets, can be extended with data acquired over social graph APIs.

Recommender Systems. The field of recommender systems is well established with solid practical developments in various fields during the last years. Based on preferences of the users and their browsing history on the Web, recommendation approaches are able to predict relevant items and pages to the users. Still, these systems deal with the limitations of preference sparsity and *cold start* problem. Another limitation is the lack of flexibility to incorporate contextual factors in the recommendation methods. To a great extent, these issues can be related to a limited description and exploitation of the semantics underlying both user and item representations. As such, a lot of research [3–5,11,15,16,21] is focused on harnessing the power of domain knowledge and semantic data, utilizing ontological concepts and relations, to provide more effective top-N recommendations.

As mentioned also earlier, one aspect how Linked APIs could help is through the semantic enrichment and plentiness of structured data that can be retrieved. Since these APIs provide a way to automatically produce semantic knowledge bases and item annotations from public sources, they yield an attractive and challenging setting for scalability evaluation. Furthemore, we believe an interesting research directions based upon these techniques is the extensions with recommendations of requests directed to Web APIs. With the growth of APIs, one can envision a shift from item/page recommendation to Web API request recommendation.

We have listed above only a few suggestions, with the goal of stimulating the interest of communities to explore novel approaches for mutual research.

2.2 KDD for Linked APIs

Very recently, researchers have argued that research on knowledge discovery, particularly machine learning (ML), can offer a large variety of methods applicable to different expressivity levels of Semantic Web knowledge bases [14]. In this section, we go a step further and elaborate on how KDD methods can be useful to tackle problems related to Web APIs, especially to the setting of Linked APIs.

Semantic Models of Web APIs While there are several benefits from integrating Web APIs with the Linked Data cloud, a key challenge is the difficulty of building the required semantic models to describe and deploy APIs, so that they directly consume Linked Data and generate RDF linked to the input data.

KDD methods can be helpful to alleviate this process, e.g. statistical models for pattern recognition and machine learning, or structure learning [8], can be

deployed in this case. Methods from inductive logic programming and ML have also been applied in the semantic Web for ontology learning. It is promising to see there adaptation for the API setting, e.g. to generate the semantic models. One such approach, more precisely Conditional Random Fields, is recently applied in the modeling process of Linked APIs [20], in order to construct the inputs/outputs worksheet from the invocation URLs, and further generate a formal model type of data by assigning semantic types to the data types. Another important data mining task that can be applicable to RDF data of API requests is the clustering of instances, also called group detection [7], to further help automate the process of building semantic models of APIs.

Web API usage mining Of special interest of a provider is to understand how the offered APIs are being used. In that respect, techniques of Web usage mining can be extended or adapted for Web API usage mining. The idea would be to apply KDD techniques upon Web APIs usage data. Particular methods of interest that we can mention here: (i) event detection and pattern discovery, e.g. to automatically detect anomalies or failures in Web APIs based on the analysis of logs of requests and responses, (ii) frequent pattern analysis, (iii) statistical relational learning, based on which we develop a model (Sec. 3) to show how research in this area can be helpful.

We elaborate in more details on the synergy approach of applying KDD methods to APIs usage data. The difference in this approach, when compared to typical usage mining, is that the requests/responses are structured and annotated (or if not fully annotated, they can still be semantically described with existing approaches). This is a powerful aspect to be used as the basis for the application of semantically-leveraged KDD techniques. This enables ways of increasing the expressivity of the query to be posed over the usage dataset for mining purposes. It also enables prediction based on relational ML methods, which are shown to be more effective than methods where no structural information is captured.

By discovering recurring patterns in the requests and responses directed to one or more Web APIs, one can detect possible problems or bottlenecks. Furthermore, the analytics power is used to make predictions on what will *most likely* occurr in the future, as such discovery potentials for **optimization** of a specific API or **composition** of several APIs.

3 Mining Web API requests with Statistical Relational Learning

3.1 API Requests Network

An API is normally invoked by sending HTTP requests, which can be broken down into the elements of which they are composed. Table 1 illustrates an example of a database of requests to different APIs that a provider may receive.

The elements of the request include the base URL, input variables (name and value), and output variables. When several requests are issued in sequence by the

Table 1. Web API Requests Database

Req.	Link	API	Description
r1	`http://maps.googleapis.com/maps/api/geocode/json?address=Karlsruhe&sensor=false`	Google GeoCoding API	Provides latitude and longitude for a given street address.
r2	`http://ws.geonames.org/findNearbyWikipedia?lat=49.0080848&lng=8.4037563`	GeoNames	Finds the nearest GeoNames feature to a given point and links a geographic point to resources from DBpedia that are nearby.
r3	`http://www.worldweatheronline.com/feed/search.ashx?query=Karlsruhe&num_of_results=3`	World Weather Online	Provides weather information on a given city name.

same user (i.e. identified by the same IP), they can be interlinked by a *nextTo* relationship. We propose a graphical representation for this rich collection of objects, which we refer to as *API requests network*. Note that this representation is related to the RDF graph, which makes it coherent with the beforementioned representation of Linked APIs. An example of such a network, based on the sample requests of Table. 1, is illustrated in Fig. 2.

3.2 Hidden Relational Model

Based on the *API requests network*, various applications of KDD techniques are interesting including relationship prediction, usage pattern discovery, API recommendations, etc. In this paper, we focus more on the task of relationship prediction and propose a model based on statistical relational learning (SRL) [7]. SRL is a prominent area of machine learning research, which combines formalisms of expressive knowledge representation with statistical approaches for performing probabilistic inference and learning on relational networks. An SLR approach is more appropriate for the task at hand, since we would like to explore the rich relational information embedded in the usage data of APIs.

Fig. 2 (right) illustrates a simple SRL model of the API requests network. We introduce a random variable, denoted as $R_{i,j}$, for each potential edge to describe its state. As such, there is binary variable associated with the edge between object request req_1 of the GeoNames API and object request req_1 of the Google GeoCoding API. The variable is 1 if the request is *nextTo* another request in the transaction, and 0 otherwise. The edge between an object (e.g. req_1) and object property (e.g. input is *address*) is also associated with a random variable denoted by G_i, whose value describes the profile of the API request. To infer whether a request is next to another request of an API, we learn the probabilistic dependencies between the random variables.

To explore non-local dependencies in the model, for each object request a hidden variable is introduced, denoted in the figure as Z_i. The state of the hidden variable represents unknown attributes of the request that still impact the relationship of interest. This model is referred to as the *hidden relational model* (HRM) [24]. In this case, we assume that the relation *nextTo* is conditioned

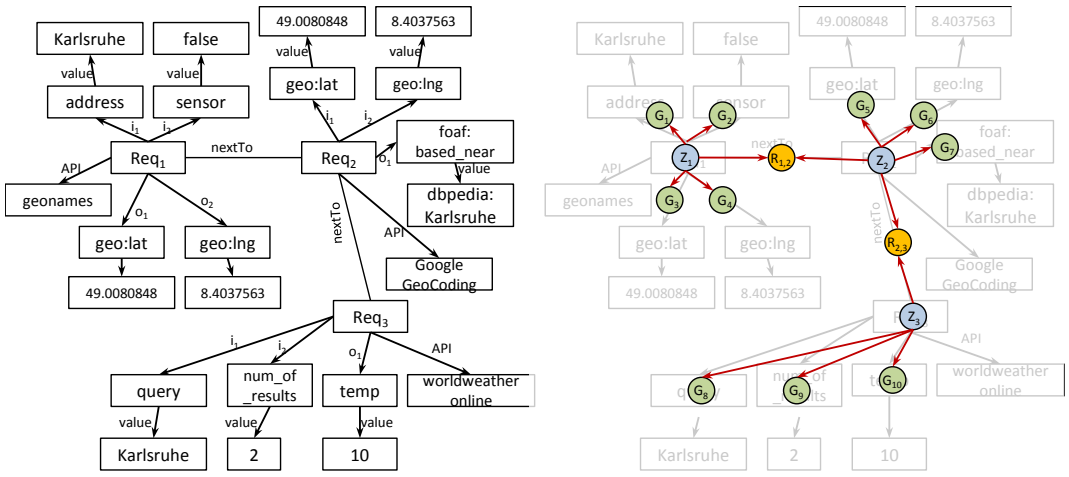

Fig. 2. Left: graph of Web API requests based upon API usage transactions database. Right: hidden relational model (HRM) of the graph. Each edge is associated with a random variable that determines the state of the edge. The directed arcs indicate direct probabilistic dependencies.

on the attributes of the API request (i.e. inputs and outputs). In Fig. 2, for simplicity, we introduce variables for the input names only. Information in the model can propagate via interconnected hidden variables.

To predict whether request Req_1 is next to another request Req_2 to the GeoCoding API, we need to predict the relationship $R_{1,2}$. The probability is computed on the evidence about (1)the attributes of the requests, i.e. $\{G_1,...,G_7\}$, (2) the known relationships associated with the objects of interests, i.e. the relations $R_{2,3}$ of request Req_2, and (3) information transferred biy hidden variables, i.g. information on G_8, G_9, G_{10} propagated via Z_3. Through the hidden variables, information is globally distributed in the ground network defined by the relational structure, which consists here of attribute variables exchanging information via a network of hidden variables.

The model provides also a cluster analysis of the API requests network. The hidden variables are drawn from a discrete probability distributions, thus they can be interpreted as cluster variables where similar API requests are grouped together. The cluster assignments (or hidden states) of the objects are decided not only by their attributes, but also by their relations.

We complete the model by introducing the parameters in Fig. 3 as in [24]. The state of Z_i specifies the cluster of a request req_i. With K denoting the number of clusters, Z follows a multinomial distribution with parameter vector π, specifying the probability of a request belonging to a cluster, i.e. $P(Z_i = k) = \pi_k$. It is drawn from a conjugated Dirichlet prior with hyperparameters α_0. The attributes of the requests are assumed to be discrete and multinomial variables, drawn from a multinomial distribution with parameters θ_k, also referred to as

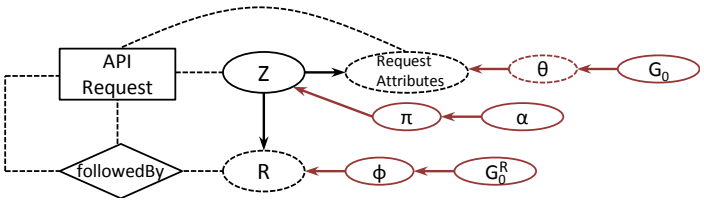

Fig. 3. Hidden Relation Model (HRM) of Web API requests

mixture component associated with the cluster k. These mixture components are independently drawn from a prior G_0 that, following [24], can be a conjugated Dirichlet prior with hyperparameters β. As the crucial components of this model, the relationships ($nextTo$) are also assocuated to variables and parameters, where relationship R is discrete with two states. Each $R_{i,j}$ is drawn from a binomial distribution with parameter $\phi_{k,l}$, where k and l denote cluster assignments of the request req_i and the request req_j, respectively. Each $\phi_{k,l}$ is independently drawn from the prior G_0^r, which can be defined as a a conjugated Beta distribution with hyperparameters β^r.

Inference. Given certain evidence of the ground network, the goal would then be to compute the probabilities of the relationships R for unobserved variables in the data. This is the inferential problem of computing the posterial probabilities, for which approximate inference methods, such as Markov chain Monte Carlo (MCMC) sampling, can be applied.

3.3 Practical Applications of HRM

As mentioned earlier, this relational model can be used for cluster analysis as well as relationship prediction. As such, based on past usage logs, we can explore which API makes more sense to request next given a specific coming request. Through the clustering analysis, we are able to group APIs together, based on how they have been frequently requested by agents.

In both cases, this not only helps to gain insights about the usage of the APIs, but most importantly can generate active knowledge on which APIs to link together to create useful mashups. Furthermore, we foresee the application of such predictive models to facilitate the automation of Web API compositions. In this case, the composition process will be founded on APIs matching driven by the respective usage behavior.

References

1. M. Adda, P. Valtchev, R. Missaoui, and C. Djeraba. A framework for mining meaningful usage patterns within a semantically enhanced web portal. In *Proceedings of the Third C* Conference on Computer Science and Software Engineering*, C3S2E '10, pages 138–147, New York, NY, USA, 2010. ACM.

2. K. Balog, R. Neumayer, and K. Nørvåg. Collection ranking and selection for federated entity search. In *Proceedings of the 19th international conference on String Processing and Information Retrieval*, SPIRE'12, pages 73–85, Berlin, Heidelberg, 2012. Springer-Verlag.
3. I. Cantador, P. Castells, and A. Bellogín. An enhanced semantic layer for hybrid recommender systems: Application to news recommendation. *Int. J. Semantic Web Inf. Syst.*, 7(1):44–78, 2011.
4. V. Codina and L. Ceccaroni. Taking advantage of semantics in recommendation systems. In *Proceedings of the 2010 conference on Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*, pages 163–172, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
5. M. Eirinaki, D. Mavroeidis, G. Tsatsaronis, and M. Vazirgiannis. Introducing semantics in web personalization: the role of ontologies. In *Proceedings of the 2005 joint international conference on Semantics, Web and Mining*, EWMF'05/KDO'05, pages 147–162, Berlin, Heidelberg, 2006. Springer-Verlag.
6. S. Endrullis, A. Thor, and E. Rahm. Entity search strategies for mashup applications. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 66–77, Washington, DC, USA, 2012. IEEE Computer Society.
7. L. Getoor, N. Friedman, D. Koller, A. Pferrer, and B. Taskar. *Probabilistic relational models*. MIT Press, 2007.
8. T. N. Huynh and R. J. Mooney. Online structure learning for markov logic networks. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part II*, ECML PKDD'11, pages 81–96, Berlin, Heidelberg, 2011. Springer-Verlag.
9. J. Jozefowska, A. Lawrynowicz, and T. Lukaszewski. The role of semantics in mining frequent patterns from knowledge bases in description logics with rules. *Theory Pract. Log. Program.*, 10(3):251–289, May 2010.
10. N. Lavrač, A. Vavpetič, L. Soldatova, I. Trajkovski, and P. K. Novak. Using ontologies in semantic data mining with segs and g-segs. In *Proceedings of the 14th international conference on Discovery science*, DS'11, pages 165–178, Berlin, Heidelberg, 2011. Springer-Verlag.
11. N. R. Mabroukeh and C. I. Ezeife. Ontology-based web recommendation from tags. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering Workshops*, ICDEW '11, pages 206–211, Washington, DC, USA, 2011. IEEE Computer Society.
12. M. Maleshkova, C. Pedrinaci, and J. Domingue. Investigating web apis on the world wide web. In *Web Services (ECOWS), 2010 IEEE 8th European Conference on*, pages 107 –114, dec. 2010.
13. B. Norton and R. Krummenacher. Consuming dynamic linked data. In *Proceedings of the First International Workshop on Consuming Linked Data (COLD), Shanghai, China, November 8, 2010*, 2010.
14. A. Rettinger, U. Lösch, V. Tresp, C. D'Amato, and N. Fanizzi. Mining the semantic web. *Data Min. Knowl. Discov.*, 24(3):613–662, May 2012.
15. M. Ruiz-Montiel and J. F. Aldana-Montes. Semantically enhanced recommender systems. In *Proceedings of the Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems: ADI, CAMS, EI2N, ISDE, IWSSA, MONET, OnToContent, ODIS, ORM, OTM Academy, SWWS, SEMELS, Beyond SAWSDL, and COMBEK 2009*, OTM '09, pages 604–609, Berlin, Heidelberg, 2009. Springer-Verlag.

16. P. Senkul and S. Salin. Improving pattern quality in web usage mining by using semantic information. *Knowl. Inf. Syst.*, 30(3):527–541, Mar. 2012.
17. M. Shokouhi and L. Si. Federated search. *Found. Trends Inf. Retr.*, 5(1):1–102, Jan. 2011.
18. S. Speiser and A. Harth. Integrating linked data and services with linked data services. In *Proceedings of the 8th Extended Semantic Web Conference on the semantic web: research and applications - Volume Part I*, ESWC'11, pages 170–184, Berlin, Heidelberg, 2011. Springer-Verlag.
19. V. Svátek, J. Rauch, and M. Ralbovský. Ontology-enhanced association mining. In *Proceedings of the 2005 joint international conference on Semantics, Web and Mining*, EWMF'05/KDO'05, pages 163–179, Berlin, Heidelberg, 2006. Springer-Verlag.
20. M. Taheriyan, C. A. Knoblock, P. A. Szekely, and J. L. Ambite. Rapidly integrating services into the linked data cloud. In P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, editors, *International Semantic Web Conference (1)*, volume 7649 of *Lecture Notes in Computer Science*, pages 559–574. Springer, 2012.
21. A. Thalhammer. Leveraging linked data analysis for semantic recommender systems. In *Proceedings of the 9th international conference on The Semantic Web: research and applications*, ESWC'12, pages 823–827, Berlin, Heidelberg, 2012. Springer-Verlag.
22. J. Weaver and P. Tarjan. Facebook linked data via the graph api. *Semantic Web Journal*, pages 1–6, 2012.
23. G. Weikum. Where's the data in the big data wave? `http://wp.sigmod.org/`. Accessed: 14/03/2013.
24. Z. Xu, V. Tresp, A. Rettinger, and K. Kersting. Social network mining with nonparametric relational models. In *Proceedings of the Second international conference on Advances in social network mining and analysis*, SNAKDD'08, pages 77–96, Berlin, Heidelberg, 2010. Springer-Verlag.
25. H. Yilmaz and P. Senkul. Using ontology and sequence information for extracting behavior patterns from web navigation logs. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 549–556, Washington, DC, USA, 2010. IEEE Computer Society.