Towards Semantic Data Mining

Haishan Liu

Department of Computer and Information Science, University of Oregon,
Eugene, OR, 97401, USA
ahoyleo@cs.uoregon.edu

Abstract. Incorporating domain knowledge is one of the most challenging problems in data mining. The Semantic Web technologies are promising to offer solutions to formally capture and efficiently use the domain knowledge. We call data mining technologies powered by the Semantic Web, capable of systematically incorporating domain knowledge, the *semantic data mining*. In this paper, we identify the importance of *semantic annotation*—a crucial step towards realizing semantic data mining by bringing meaning to data, and propose a learning-based semantic search algorithm for annotating (semi-) structured data.

Keywords: Ontology, Data mining, Semantic annotation

1 Introduction

It has been widely stated that one of the most important and challenging problems in data mining is incorporation of domain knowledge. Fayyad et al. [4] contended that the use of domain knowledge is important in all stages of the knowledge discovery process. When both data and domain knowledge are available, it is worthwhile to explore the fusion of them. In practice, however, users of data mining systems are mostly encouraged to express domain knowledge in a application-specific form, scope and granularity. The ad hoc manner of the representation hinders efficient usage of the codified knowledge.

At the same time, research in the area of the Semantic Web has led to quite mature standards for modeling and codifying knowledge. Today, Semantic Web ontologies become a key technology for intelligent knowledge processing, providing a framework for sharing conceptual models about a domain. Moreover, the large and continuously growing amount of interlinked Semantic Web data have provided perfect applications for novel data mining methods. Such methods focus on relations between objects in addition to features/attributes of objects [6]. We propose to exploit the advances of the Semantic Web technologies to formally represent domain knowledge including structured collection of prior information, inference rules, knowledge enriched datasets etc, and thus develop frameworks for systematic incorporation of domain knowledge in an intelligent data mining environment. We call this technology the *semantic data mining*. Our ongoing NEMO project has made a first step to formally representing knowledge in the ERP domain [5], and produced considerable amount of data in RDF. How to

utilize the knowledge and linked data and perform efficient mining becomes the major research question that motivates our research on semantic data mining.

The majority of data underpinning a wide spectrum of data mining applications are stored in structured sources such as relational databases (RDB) with their proven track record of scalability and reliability, or in semi-structured sources such as spreadsheets with their advantage of low maintenance and cheaper overheads. Hence the problem of how to impart knowledge encoded in Semantic Web ontologies to (semi-)structured data becomes a major challenge in realizing the semantic data mining. *Semantic annotation* aims at addressing this challenge by assigning semantic descriptions to elements of data. To ease the burden of common users that are not familiar with the Semantic Web, we propose, in this paper, a learning-based semantic search algorithm to automatically suggest appropriate semantic descriptions for annotation.

In the next section, we first examine the field of semantic annotation and then propose the learning-based solution.

2 Automatic Annotation by Semantic Search

Semantic Annotation aims at assigning to the basic element of information links to formal semantic descriptions [7]. Such elements should constitute the semantics of their source. Semantic annotation is crucial in realizing semantic data mining by bringing meanings to data. Annotating unstructured data (e.g., text) has been studied more extensively than annotating (semi-)structured data due to the proliferation of information extraction techniques that facilitate automatic entity recognition from text. Since large amount of data for knowledge discovery applications are stored in (semi-)structured sources, we focus on studying annotating (semi-)structured data in this paper.

The annotation process can be generally divided into two steps. The first is to establish mappings between existing Semantic Web terms and those need to be annotated in data. The second step is to come up with a local ontological structure constituting the semantic web terms to model the data. Most of previous work in annotating (semi-)structured data focus on the second step. Some skip the first step and bootstrap the ontological terms and structure from the local data itself. For example, a number of systems that map data in RDB to RDF format leverage a set of rules such as "table to class and column to predicate".

We argue that such syntactical translation alone without referencing to existing semantic descriptions does not lend itself well to aiding semantic data mining. The automatically constructed self-contained local ontology may be applicable to describe a specific dataset but is most likely too rough to capture the full domain semantics that is necessary to express meaningful domain knowledge. Moreover, with the advent of the Semantic Web and pervasive connectivity, an increasing number of ontologies has been made widely available for reuse. These ontologies are created by thorough knowledge engineering process and should serve as better models for annotation. However, on the other hand, the sheer number of Semantic Web ontologies and lack of effective search functionality

can lead to a huge hidden barrier for common users. Choosing proper Semantic Web ontologies and terms (classes and properties) requires familiarity with appropriate ontologies and the terms they define. There is very few system that is able to provide automatic suggestions. To solve this problem, we propose a learning-based semantic search algorithm to suggest proper Semantic Web terms and ontologies for annotation given semantically related words and general domain and context information.

2.1 Proposed Learning-based Semantic Search Algorithm

In order to suggest suitable Semantic Web terms and ontologies for users to annotate their data, we propose a learning-based semantic search algorithm. We first submit a list of terms appeared in the schema of (semi-)structured data to our semantic search algorithm and then use the returned results for annotation. In a fully automatic setting, the search algorithm is configured to return the top-1 hit; while in an interactive setting, the search algorithm returns ordered top-k search results for users to decide. Previous semantic search algorithms leverage a variety of measures, including lexical and structural similarities (see details below) to rank Semantic Web documents according to how likely they can be semantically matched to the search terms. However, using any single measure alone may not be sufficient to achieve the optimal result. We propose to combine various measures to a weighted feature-based search model, where the weights are learned from training data. We believe the incorporation of learning techniques will improve the semantic search result.

Feature-based Semantic Search Model. Consider a set of ontologies $\mathbf{O} = \{O_1...O_m\}$ returned as the search result for a specific search term. Let $\mathbf{\Phi} = \{\phi_1(O_i)...\phi_m(O_m)\}$ be a vector of real-valued feature functions $\phi : O \mapsto \mathbb{R}$ that compute rank indicating how ontologies should be ordered in the search result. The one with the highest rank is the top-hit for a specific search. Let $\mathbf{W} = \{w_1...w_m\}$ be a vector of real-valued weights associated with each feature. A score is computed for each ontology O_i by taking the dot product of the features and weights: $\tau(O_i, \mathbf{W}) = \mathbf{\Phi} \cdot \mathbf{W}$. We can leverage a variety of ranking methods proposed in the literature as the feature functions $\mathbf{\Phi}$. For example, Alan et al. [1] proposed four types of measurements to evaluate the ranks for ontologies given a list of search terms; The Swoogle search engine [2][3] weighs different types of links between Semantic Web data and rank them using link-based algorithms at three levels of granularity: documents, terms and RDF graphs; Maedche et al. [9] described a two level similarity measure considering both lexical and conceptual comparisons, etc.

Training Set. Our algorithm to determine the weight vector $\mathbf{W}$ requires a *training set* of known top hit in the search result (chosen by human): $\mathcal{T} = \{< \mathbf{O}^1, l_1 > ... < \mathbf{O}^n, l_n >\}$, where each set of ontology namespaces $\mathbf{O}^i = \{O_1...O_k\}$ is associated with label $l_i \in \{1...k\}$, indicating which of the ontologies should

be selected for annotating the specific term t_i (i.e., $O_{l_i} \in O$ is the true ontology selected by human as the best choice for annotating the term). There are several ways to estimate $\mathbf{W}$ from the training set $\mathcal{T}$ as described below.

Subgradient Descent. We can view the weight learning as maximum margin structured learning problem. Given a training set and loss function, the learned $\mathbf{W}$ should score each known top-hit result O_{l_i} higher than all other O by at least $\mathcal{L}(O_{l_i}, O)$, where $\mathcal{L}$ is the loss function. Mathematically, this constraint is

$$\forall i, O \in \{\mathbf{O}^i \backslash O_{l_i}\}, \mathbf{W} \cdot \mathbf{\Phi}(O_{l_i}) \geq \mathbf{W} \cdot \mathbf{\Phi}(O) + \mathcal{L}(O_{l_i}, O) \ ,$$

where $\{\mathbf{O}^i \backslash O_{l_i}\}$ is the set of possible ontologies returned by a specific search query excluding the gold standard ontology O_{l_i} chosen by human. We can express this constraint as the following convex program:

$$\min_{\mathbf{W}, \zeta^i} \frac{\lambda}{2}\|\mathbf{W}\|^2 + \frac{1}{d}\sum_{i=1}^{d} \zeta^i \ . \ s.t. \forall i, O \in \mathbf{O}, \mathbf{W} \cdot \mathbf{\Phi}(O_{l_i}) + \zeta_i \geq \mathbf{W} \cdot \mathbf{\Phi}(O) + \mathcal{L}(O_{l_i}, O) \ ,$$

where λ is a regularization term that prevents overfitting. We can rearrange the convex program to show that the optimal $\mathbf{W}$ minimizes

$$c(\mathbf{W}) = \frac{1}{d}\sum_{i=1}^{d} r^i(\mathbf{W}) + \frac{\lambda}{2}\|\mathbf{W}\|^2 \ , \tag{1}$$

where $r_i(\mathbf{w}) = \max_{O \in \{O_{l_i}\}}(\mathbf{W} \cdot \mathbf{\Phi}(O) + \mathcal{L}(O_{l_i}, O)) - \mathbf{W} \cdot \mathbf{\Phi}(O_{l_i})$. This objective function is convex but nondifferentiable. We can therefore minimize it with subgradient descent, an extension of gradient descent to nondifferentiable objective functions. The subgradient of equation 1 can be computed iteratively [10] .

Logistic Regression. The second method is based on logistic regression (sometimes called maximum entropy classification). We modify the traditional logistic regression loss function to rank, rather than classify, instances. Let the binary random variable C_i be 1 if and only if ontology O_i is the gold standard chosen by human. Given $\mathbf{W}$ and $\mathbf{\Phi}$, we can compute the probability of C_i as follows:

$$p\left(C_i = 1 | \mathbf{O}, \mathbf{W}\right) = \frac{e^{\tau(O_i, \mathbf{W})}}{\sum_{O_j \in \mathbf{O}} e^{\tau(O_j, \mathbf{W})}} \ ,$$

where the score for ontology O_i is normalized by the scores for every other ontologies. We can estimate $\mathbf{W}$ from the training set $\mathcal{T}$ by minimizing the negative log-likelihood of the data given $\mathbf{W}$:

$$\mathcal{L}\left(\mathbf{W}, \mathcal{T}\right) = -\sum_{O^i \in \mathcal{T}} \log p\left(C_{l_i} | \mathbf{O}, \mathbf{W}\right) \ . \tag{2}$$

We can find the setting of $\mathbf{W}$ that minimizes Equation 2 using limited-memory BFGS, a gradient ascent method with a second-order approximation [8].

3 Conclusion and Future Work

In this paper, we introduce semantic data mining, an area we envision emerging as the solution to systematic incorporation of domain knowledge in data mining with the help of the Semantic Web technologies. We also recognize that vast amount of information stored in (semi-)structured sources is calling for attention to develop innovative approaches to solve following challenges: 1) how to impart knowledge encoded in ontologies into the (semi-) structured data, and 2) exploration of more meaningful ways to utilize the knowledge. We believe semantic annotation is the solution to the first challenge. In this paper, we propose a learning-based semantic search algorithm to suggest appropriate Semantic Web terms and ontologies.

4 Acknowledgement

This work is supported by the NIH funded NEMO project (Grant No. R01EB007684 NIH/NIBIB). I am grateful to Dr. Dejing Dou and Dr. Daniel Lowd for helpful discussion and comments.

References

1. H. Alani and C. Brewster. Ontology ranking based on the analysis of concept structures. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, pages 51–58, New York, NY, USA, 2005. ACM.
2. L. Ding, T. Finin, A. Joshi, Y. Peng, R. Pan, and P. Reddivari. Search on the semantic web. *IEEE Computer*, 10, 2005.
3. L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. In *In Proceedings of the 4th International Semantic Web Conference*, pages 156–170, 2005.
4. U. Fayyad, G. Piatetsky-shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.
5. G. Frishkoff, P. LePendu, R. Frank, H. Liu, and D. Dou. Development of Neural Electromagnetic Ontologies (NEMO): Ontology-based Tools for Representation and Integration of Event-related Brain Potentials. In *Proceedings of the International Conference on Biomedical Ontology (ICBO)*, pages 31–34, 2009.
6. C. Kiefer, A. Bernstein, and A. Locher. Adding data mining support to sparql via statistical relational learning methods. pages 478–492. 2008.
7. A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *J. Web Sem.*, 2(1):49–79, 2004.
8. D. C. Liu, J. Nocedal, and D. C. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
9. A. Maedche and S. Staab. Measuring similarity between ontologies. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 251–263, London, UK, 2002. Springer-Verlag.
10. N. D. Ratliff, J. Bagnell, and M. Zinkevich. Subgradient methods for structured prediction. In *Eleventh International Conference on Artificial Intelligence and Statistics (AIStats)*, 2007.