Using Ontologies in Semantic Data Mining with g-SEGS and Aleph

Anže Vavpetič[1], Nada Lavrač[1,2]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[2] University of Nova Gorica, Nova Gorica, Slovenia
{anze.vavpetic, nada.lavrac}@ijs.si

Abstract. The paper describes a prototype semantic data mining system called g-SEGS, which uses ontologies as background knowledge in the learning process. The system is a generalization of an existing system SEGS, which was successfully used in the field of functional genomics, but cannot be applied to other fields. g-SEGS is implemented as a web service and integrated into Orange4WS, a visual programming environment for data mining. In addtion, the paper describes how to formulate the problem of semantic data mining in the inductive logic programming system Aleph. Both approaches are experimentally evaluated on two real-life biological domains.

1 Introduction

The knowledge discovery process can significantly benefit from the domain (background) knowledge, as successfully exploited in relational data mining and Inductive Logic Programming (ILP). Additional means of providing more information to the learner is by providing semantic descriptors to the data.

Usually, there is abundant empirical data, while the background knowledge is scarce. However, with the expanding of the Semantic Web and the availability of numerous ontologies which provide domain background knowledge and semantic descriptors to the data, the amount of *semantic data* (e.g., ontologies and annotated data collections) is rapidly growing[1]. The data mining community is now faced with a paradigm shift: instead of mining the abundance of empirical data supported by the background knowledge, the new challenge is to mine the abundance of knowledge encoded in domain ontologies, constrained by the heuristics computed from the empirical data collection. This paper uses the term *semantic data mining* to denote this new data mining challenge and approaches in which semantic data are mined.

In this paper we describe g-SEGS, a prototype semantic data mining system implemented in the novel service-oriented data mining environment Orange4WS [7]. System g-SEGS is a successor of SEGS, a system for Searching of Enriched Gene Sets [10] designed specifically for functional genomics tasks. While SEGS is a special purpose system for analyzing microarray data with biological ontologies

[1] See the Linked Data site http://linkeddata.org/

as background knowledge, g-SEGS is a general purpose semantic data mining system.

The described semantic data mining task requires a level of expressiveness that cannot be adequately represented in propositional logic. The reason is that ontologies can encode extremely complex relations. Since this clearly becomes a relational data mining problem, we find it instructing to employ an inductive logic programming system Aleph[2] to this task as well. In this paper we describe the procedure to formulate the semantic data mining task in Aleph.

In order to empirically compare both approaches we evaluated them on two real-life problems from the field of functional genomics.

The paper is organized as follows. We provide the related work in Section 2. In Section 3 we present the prototype system g-SEGS. Section 4 describes the problem formulation in Aleph. In Section 5 we present the experimental results. Section 6 concludes the paper and gives some ideas for further work.

2 Related work

The idea of using hierarchies is not new. It was already proposed by Michalski in 1983 [6], where a methodology which enables the use of hiearchies for generalizing terms in inductive rule learning is described. In [4], the use of taxonomies (where the leaves of the taxonomy correspond to attributes of the input data) on paleontological data is studied. In [1], background knowledge is presented in the standard inheritance network notation and the KBRL algorithm performs a general-to-specific heuristic search for a set of conjuctive rules that satisfy user-defined rule evaluation criteria. A domain specific system that uses ontologies and other hierarchies as background knowledge for data mining is SEGS [10]. Given labelled gene expression data and several biomedical ontologies as input, the SEGS system finds groups of differentially expressed genes, called *enriched gene sets*[3].

The main differences of system g-SEGS, described in this paper, compared to the related approaches is that these (1) use non-standard ontology formats [4, 10], (2) are domain specific [4, 10], (3) are not implemented as web services [1, 4] and (4) perform non-symbolic classification tasks [4].

3 g-SEGS

This section describes a prototype semantic data mining system, called g-SEGS, which can be used to discover subgroup descriptions for labelled or ranked data with the use of input OWL ontologies as background knowledge. The ontologies are exploited in a similar manner as in SEGS (i.e. ontological concepts are used

[2] http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/aleph.html

[3] A gene set is enriched if the genes that are members of this gene set are statistically significantly differentially expressed compared to the rest of the genes.

as terms that form rule conjuncts), with the important difference that they can be (1) from any domain and (2) in a standard OWL format.

The following paragraphs describe the main parts of our prototype system g-SEGS: the input data, the hypothesis language, the rule construction algorithm, the rule selection and evaluation principles and its implementation.

Input. Apart from various parameters (e.g. for controlling minimum support criteria, maximal rule length, etc) the main inputs are: (1) *background knowledge* in the form of OWL ontologies or in the legacy SEGS format, (2) *training data* which is a list of class-labelled or ranked examples and (3) an *example-to-ontology map* which associates each example with a number of concepts from the given ontologies. We say that an example is *annotated* with these concepts.

Hypothesis language. The hypothesis language are rules $class(X) \leftarrow Conditions$, where $Conditions$ is a logical conjuction of terms which represent ontological concepts. If we put this into a more illustrative context, a possible rule could have the following form: $class(X) \leftarrow doctor(X) \wedge germany(X)$. Both *doctor* and *germany* are terms which represent the ontological concepts *doctor* and *germany*. If our input examples are people, we can say that this rule describes a subgroup of people who are doctors and live in Germany.

Rule construction. A set of rules, which satisfy the size constraints (minimal support and maximal number of rule terms), is constructed using a top-down bounded *exhaustive* search algorithm, which enumerates all such possible rules by taking one term from each ontology. Note that because of the properties of the $subClassOf$ relation between concepts in the ontologies, the algorithm can prune all the rules, which would be generated by further specializing a term, which - if conjuctively added to the current rule - fails to meet the size constraints.

Additionally, the user can specify another relation between the input examples - the *interacts* relation. Two examples are in this relation, if they interact between each other in some way. For each concept, which the algorithm tries to conjuctively add to the rule, it also tries to add its interacting counterpart, e.g. the antecedent of the rule of the form $class(X) \leftarrow c_1(X) \wedge interacts(X, Y) \wedge c_2(Y)$ can be interpreted as: all the examples which are annotated by concept c_1 and interact with examples annotated by concept c_2.

Rule selection. As the number of generated rules can be large, uninteresting and overlapping rules have to be filtered out. In g-SEGS, rule filtering is performed using *wWRAcc* (Weighted Relative Accuray with example weights) [5], which uses example weights as means for considering different parts of the example space when selecting the best rules in rules postprocessing by a weighted covering algorithm used for rule selection.

Implementation. g-SEGS is implemented as a web service in the Orange4WS [7] environment which upgrades the freely available Orange [3] data mining environment. Additionally we developed an easy-to-use user interface for our system in

Orange, allowing simple experimentation with g-SEGS by using it in workflows together with the existing Orange widgets.

4 Problem formulation in Aleph

In order to solve similar semantic data mining tasks in Aleph as with g-SEGS, we need to encode (1) the ontologies, (2) the given examples and (3) the example-to-ontology map (annotations).

Each ontological concept `c`, with child concepts `c1,c2,..,cm`, is encoded as a recursively defined unary predicate `c/1`:

```
c(X) :- c1(X) ; c2(X) ; ... ; cm(X).
```

Each child concept is defined in the same way. To encode the whole ontology, one needs to start this procedure at the root concept. All these predicates are allowed to be used in the hypothesis body and are tabled for faster execution.

Each example is encoded as an atom defined for the concepts with which it is annotated. If the k-th example is annotated by concepts `c1,c2,..,cm` (this is defined by the example-to-ontology map), we encode it as:

```
instance(ik). c1(ik). c2(ik). ... cm(ik).
```

Additional relations (if available) can also be trivially added to the background knowledge.

In order to encode the input examples, we transform the ranked or labelled problem into a two-class problem (the positive class is the class which interests the user) and split the examples accordingly.

5 Experimental results

We tested both approaches on two publicly available[4] biological microarray datasets: *acute lymphoblastic leukemia* (ALL) [2] and *human mesenchymal stem cells* (hMSC) [11]. Both datasets encode gene expression data for two classes and the challenge is to produce descriptions of sets of differentially expressed genes involved in the process of each domain.

First, we preprocessed the datasets by following the SegMine [8] methodology. Genes were first ranked using the ReliefF [9] algorithm and then filtered using the logarithm of expression fold change (logFC). All genes with $|logFC| < 0.3$ were removed from the set, resulting in 9,001 genes in the ALL domain and 20,326 genes in the hMSC domain.

The ranked genes were annotated by Gene Ontology and KEGG (Kyoto Encyclopedia of Genes and Genomes) concepts by using the ENTREZ database to map between gene indentifiers and the ontology concepts. The top 300 were

[4] http://segmine.ijs.si

used as the positive class and from the remaining examples we have randomly selected 300 examples, which were labelled as negative.

Table 1 present the performance of both approaches on the two domains. The discovered rule sets were evaluated using the descriptive measures of rule interestingness as proposed in [5]: $AvgCov$ - the average rule coverage, $AvgSup$ - the overall support, $AvgSig$ - the average significance, $AvgWRAcc$ - the average unusualness and AUC - the area under the convex hull (method 1). Additionally we also measured the execution time t.

ALL

System	$AvgCov$	$AvgSup$	$AvgSig$	$AvgWRAcc$	AUC	t[s]
g-SEGS	0.043	0.477	**14.145**	0.0135	0.564	**5.79**
Aleph	**0.108**	**0.967**	6.969	**0.0156**	**0.592**	159.68

hMSC

System	$AvgCov$	$AvgSup$	$AvgSig$	$AvgWRAcc$	AUC	t[s]
g-SEGS	0.047	0.387	**2.354**	0.0039	**0.530**	**4.40**
Aleph	**0.086**	**0.963**	1.767	**0.0057**	**0.530**	132.53

Table 1. Experimental results.

The results show that g-SEGS produces more significant rules. Other measures indicate that a rule discovered by Aleph (on average) covers a higher portion of positive examples and is more unusual and thus potentially more interesting for the domain expert. Aleph's rule set also covers a higher percentage of the positive examples. An interesting fact is also that g-SEGS produces the resulting rule set approximately thirty times faster than Aleph, which is due to the fact that g-SEGS exploits the hierarchical properties of the ontologies. By defining the `refine/2` predicate, Aleph could also benefit from the hierarchical properties and we plan to implement this in future work. Of course we need to take into account that it is not necessary that these measures reflect a better rule set, which would in fact provide novel and interesting knowledge for the domain expert. Such an analysis with a domain expert is planned in future work.

6 Conclusion

This paper presents g-SEGS, a general purpose semantic data mining system, based on the successful system SEGS, designed exclusively for functional genomics. The paper also shows how to solve a similar task with the general purpose ILP system Aleph. Both approaches were experimentally evaluated on two real-life biological domains. The evaluation shows that, although g-SEGS produces more significant rules, Aleph clearly uses some concepts which are successful for this type of tasks and its potential should necessarily be further explored in building of the system for semantic data mining of linked data, planed in our future work.

Acknowledgments

The research presented in this paper was supported by the Slovenian Ministry of Higher Education, Science and Technology (grant no. P-103) and the EU-FP7 projects e-LICO and BISON.

References

[1] J.M. Aronis, F.J. Provost, and B.G. Buchanan. Exploiting background knowledge in automated discovery. In *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 355–358, 1996.

[2] Sabina Chiaretti, Xiaochun Li, Robert Gentleman, Antonella Vitale, Marco Vignetti, Franco Mandelli, Jerome Ritz, and Robin Foa. Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103:2771–2778, 2004.

[3] J. Demšar, B. Zupan, and G. Leban. Orange: From experimental machine learning to interactive data mining, white paper (www.ailab.si/orange). Faculty of Computer and Information Science, University of Ljubljana, 2004.

[4] G.C. Garriga, A. Ukkonen, and H. Mannila. Feature selection in taxonomies with applications to paleontology. In *Proc. of the 11th International Conference on Discovery Science*, DS '08, pages 112–123. Springer-Verlag, 2008.

[5] N. Lavrač, B. Kavšek, P.A. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[6] Ryszard S. Michalski. Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):349–361, 1980.

[7] V. Podpečan, M. Juršič, M. Žakova, and N. Lavrač. Towards a service-oriented knowledge discovery platform. In V. Podpečan and N. Lavrač, editors, *Third-generation data mining: towards service-oriented knowledge discovery*, pages 25–36, 2009.

[8] Vid Podpečan, Nada Lavrač, Igor Mozetič, Petra Kralj Novak, Igor Trajkovski, Laura Langohr, Kimmo Kulovesi, Hannu Toivonen, Marko Petek, Helena Motaln, and Kristina Gruden. SegMine workflows for semantic microarray data analysis in Orange4WS. *In preparation.*

[9] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53:23–69, October 2003.

[10] I. Trajkovski, N. Lavrač, and J. Tolar. SEGS: Search for enriched gene sets in microarray data. *Journal of Biomedical Informatics*, 41(4):588–601, 2008.

[11] Wolfgang Wagner, Patrick Horn, Mirco Castoldi, Anke Diehlmann, Simone Bork, Rainer Saffrich, Vladimir Benes, Jonathon Blake, Stefan Pfister, Volker Eckstein, and Anthony D. Ho. Replicative senescence of mesenchymal stem cells: A continuous and organized process. *PLoS ONE*, 3(5):e2213, 05 2008.