Co-Occurrence-Embedding for Relational Data and Queries

Hannes Schulz

18. February 2009

Contents

1 Introduction

Inductive Logic Programming (ILP) is a scientific discipline at the intersection of machine learning and logic programming. The term was introduced by Stephen Muggleton [Muggleton and King, 1991, Muggleton, 1992]. ILP strives to explain a set of observations O by a theory H and the background knowledge B. O, H and B are specified using a more or less restricted subset of First Order Logic (FOL).

ILP inherits its choice of FOL from logic programming. Its advocates claim that FOL is expressive and intuitive. It is expressive, since very complex concepts including variables can be described [Dantsin et al., 2001]. Additionally, FOL rules are said to be intuitive because their semantics easily map to natural language sentences.

As inherited from its machine learning ancestry and in contrast to for example deductive reasoning, in ILP the theory H is inferred from background knowledge B and observations O by induction and is therefore not necessarily correct. In fact, there may be infinitely many possible H. An ILP learner therefore has to introduce a bias such as Occam's Razor [Blumer et al., 1987] or Plotkin's Relative Least General Generalization [Plotkin and of Edinburgh, 1971] when searching for and deciding on a theory H.

A "good" theory in ILP must be carefully chosen to balance between how much of the data is explained (coverage) and how complicated its description is (simplicity). The rationale here is that simpler explanations tend to have better generalization properties

and are therefore better fit for prediction purposes. However, for non-trivial datasets there is no obvious answer as to where to draw this line.

It is crucial to notice that the tradeoff between coverage and simplicity is not a purely technical matter. It also affects the intuitivity of the generated rules. A complicated theory has more and longer rules, which become increasingly hard to interpret, irrespective of the general intuitivity of FOL rules. Furthermore, complex datasets call for complex and possibly difficult to understand theories which are therefore hardly avoidable.

In this work, we argue that a FOL descriptions of a theory is not "by definition" intuitive and can sometimes better be understood using visual aids. The argument is not only valid to generated theories, however. The background knowledge B and the observations O are also specified in FOL and can, through their mere amount, defy human interpretation. In machine learning the choice of the correct algorithm and parameters depends heavily on the understanding of the data, and ILP may be no exception. In this work we consequently propose a visualization method for ILP datasets which provides such necessary insights. Our method is based on the idea to provide a structured and easily accessible visualization of what is true in which parts of the dataset, as discovered by ILP search algorithms.

2 Searching the Relational Lattice

The procedure searching for the theory H is at the heart of ILP. Since the space of possible FOL formulas is intractably large, one typically proceeds as follows. First, the search space is restricted by defining a language bias based on the available descriptions in the observations O and the background B. Second, an ordering, typically based on θ-subsumption is defined on possible theories [Dzeroski, 2007]. This ordering could be an ordering from general to specific or from specific to general and need not be strict nor ; it therefore forms a so-called lattice. Finally, the lattice is searched for good theories. As the branching factor is very large, heuristic or greedy search strategies are typically employed (ibd.).

FiXme: *strikt und ???*

We would like to utilize the knowledge discovered throughout the search process for the visualization of a dataset. A canonical choice would be to visualize the searched lattice itself. As we will point out in the following, this is not the best choice.

Firstly, although the notion of the lattice is intuitive, the lattice searched for a real dataset can not easily be visualized in a straight-forward way. As an example, consider the level in the lattice consisting only of all known atoms. The next more specific level in the lattice is the level consisting of all pairwise conjunctions of all atoms. Even setting aside the combinatorial explosion, we need to impose more structure onto the space of queries.

Secondly, the lattice search is determined by both, syntax and data. The syntactical relations between the theories upon which the ordering is defined are used to generate candidate theories, while their relation to the actual dataset, such as the coverage of the theory, is used for the heuristic. This interaction is at the core of ILP. The syntactical structure and the structure with respect to the data have very different

properties. For example, from the syntactical point of view it is certain that more specific theories will cover equal or less of the data than the general ones from which they are derived. Little can be said about theories for which the generality relation is not defined. In an actual dataset, however, we observe that very different theories can have a very similar coverage, i.e. they explain the same subset of the data. They need not be similar and a ordering relation between them need not be defined.

Thirdly, there may be many paths through the lattice which arrive at a selected theory. For trivial example, consider the theory

$$p(a) \wedge p(b) \wedge p(c).$$

It could be generated by either combining

$$(1) \quad p(a) \wedge p(b) \text{ and } p(c) \qquad \text{or} \qquad (2) \quad p(a) \text{ and } p(b) \wedge p(c)$$

However, for efficiency reasons, only one path will be pursued. Thus, even queries which are very similar with respect to the data and similar syntactically can be arrived by very different means.

To summarize, although the syntax-based lattice is crucial for the generation of theories, it has the drawbacks of being data-agnostic and redundant.

We propose instead to focus on the relation of the searched theories with the data. As we will see, among new insights this relation also reflects some important properties of the lattice.

3 Co-Occurrence Embedding for Relational Data

Machine Learning often deals with complex objects described by a possibly large set of features. Assuming that there is some structure in the data, it may be worth looking at a low-dimensional manifold within the high-dimensional data. Firstly, such a low-dimensional representation lacks irrelevant dimensions which pose a problem for many machine learning algorithms. Secondly, by choosing a favorable projection, interesting properties of the data can be emphasized. Finally, the low-dimensional representation can be optimized to respect the intrinsic structure of the data and can therefore be used to visualize the high-dimensional space.

Numerous unsupervised (PCA, ...) and supervised (...) dimensionality reduction algorithms have been proposed for this setting [Fodor, 2002]. However, often only similarities between the objects can be measured. Multidimensional Scaling (MDS, Cox and Cox [2001]), Locally Linear Embedding [Roweis and Saul, 2000]) and IsoMap [Tenenbaum et al., 2000] were introduced to embed objects in a n-dimensional space based on pairwise distances only.

A problem occurs, however, if the objects are of different types. Then it could be that they either cannot naturally be embedded in a common high-dimensional feature space in the first place or that the similarity relation between some of them is not defined. Recently, embedding techniques [Globerson et al., 2007, Iwata et al., 2007] have been developed for this setting. They derive their similarity measure from

statistics extracted from the data, such as joint or conditional probabilities of objects co-occurring. The information on how similarity is measured is then further exploited by deriving a gradient which can be used to guide a gradient descend search towards a locally optimal embedding.

For our purposes we will focus on the Euclidean Embedding of Co-Occurrence Data introduced by Globerson et al. [2007]. The authors suggest to place objects which often occur together near to each other, while objects which do not occur together are assigned positions far from each other. The embedding is therefore based on the empirical estimate of the joint probability distribution $p(x, y)$ of two random variables X and Y. The authors derive a least squares gradient descend algorithm which optimizes random initial positions of objects representing the values of X and Y such that their relations in space respect the empirical co-occurrence measure. Additionally, the authors introduce extensions to add further embedding constraints, such as co-occurrence-statistics within a random variable or embedding based on the co-occurrence statistics of more than two variables.

The co-occurrence statistics used in Globerson et al. [2007] are particularly interesting from the ILP perspective. An ILP learner is ultimately guided towards the induced theory by the frequencies of queries in the data. Instead of just focusing on the absolute number of examples covered, we can further determine which objects in our observations co-occur with which queries. The discovered co-occurrence statistics can be used to embed observations and queries in a common space.

References

Alselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Inf. Process. Lett.*, 24(6):377–380, 1987. URL `http://portal.acm.org/citation.cfm?id=31168.31174`.

T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* CRC Press, 2001.

Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001. doi: 10.1145/502807.502810. URL `http://portal.acm.org/citation.cfm?id=502810`.

Saso Dzeroski. *Introduction to statistical relational learning*, chapter 3, pages 57–92. MIT Press, 2007.

I.K. Fodor. A survey of dimension reduction techniques. *Manuscript*, 2002.

A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean Embedding of Co-occurrence Data. *The Journal of Machine Learning Research*, 8:2265–2295, 2007.

T. Iwata, K. Saito, N. Ueda, S. Stromsten, T.L. Griffiths, and J.B. Tenenbaum. Parametric Embedding for Class Visualization. *Neural Computation*, 19(9):2536–2556, 2007.

S. Muggleton and R. King. Predicting protein secondary-structure using inductive logic programming. Technical report, Turing Institute, Glasgow, Scotland, 1991.

S. H. Muggleton, editor. Academic Press, New York, NY, 1992.

G.D. Plotkin and University of Edinburgh. Automatic Methods of Inductive Inference. 1971.

S.T. Roweis and L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding, 2000.

J.B. Tenenbaum, V. Silva, and J.C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction, 2000.