

Categorizing Unsupervised Relational Learning Algorithms*

Hannah Blau Amy McGovern

Department of Computer Science
University of Massachusetts
Amherst, Massachusetts 01003-9264
{blau,amy}@cs.umass.edu

Abstract

We outline some criteria by which to compare unsupervised relational learning algorithms, and illustrate these criteria with reference to three examples: SUBDUE, relational association rules (WARMR), and Probabilistic Relational Models. For each algorithm we ask, What form of input data does it require? What form of output does it produce? Can the output be used to make predictions about unseen inputs? Categorizing the existing unsupervised relational learning algorithms helps us to understand how each algorithm relates to the others (no pun intended). We can identify important gaps in coverage that could be fruitful areas for future research.

1 What do we mean by *unsupervised*?

In this paper we outline some criteria by which to compare unsupervised relational learning algorithms. We begin by clarifying what we mean by an *unsupervised* learning algorithm. A *supervised* learning algorithm distinguishes one attribute of its input instances as the target and learns a model designed to predict the value of the target attribute for previously unseen inputs. The target attribute can be discrete, as in classification, or continuous. An *unsupervised* learning algorithm does not treat any particular attribute of its input instances as the target to be learned. There is no teacher who gives the correct answer; there *is* no one correct answer. In some cases, the model produced by an unsupervised learning algorithm can be used for prediction tasks even though it was not designed for such tasks. The distinction between supervised and unsupervised learning is a spectrum on which some algorithms are at the extremes and others are toward the middle. SUBDUE is clearly an unsupervised learning algorithm.

*This effort is supported by DARPA and AFRL under contract numbers F30602-00-2-0597 and F30602-01-2-0566, and by NSF under contract number EIA9983215. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, AFRL, NSF, or the U.S. Government.

It recognizes repeated substructures in a labeled graph, and can be used for graph compression [Cook and Holder, 1994] and for hierarchical clustering [Jonker *et al.*, 2001], but not prediction. Relational Markov Networks [Taskar *et al.*, 2002] are designed for discriminative training: they fall at the supervised end of the spectrum. Probabilistic Relational Models are more toward the middle. PRMs learn a dependency structure which can enhance a domain expert’s understanding of the data [Getoor *et al.*, 2001]. They can model uncertainty in the relational structure of the domain [Getoor *et al.*, 2002]. They can be used for classification and for clustering [Taskar *et al.*, 2001]. The underlying learning algorithm is the same, but the relational data structures given as input are adapted to the desired task.

2 Criteria of comparison

Unsupervised relational learning algorithms can be categorized along several different axes:

- What form of input data does the algorithm require?
- What form of output does it produce?
- Can the output be used to make predictions about unseen inputs?

To describe the input data configuration, we employ the terms *object*, *link*, and *attribute*. (We choose *link* instead of *relation* to avoid confusion with the terminology of relational database management systems.) In our framework, relational data consist of objects connected together by links. Both objects and links can have attributes. An attribute is a name-value pair.

The input for any learning algorithm that claims to be “relational” must have links as well as objects. A link can be represented explicitly by an edge in a graph, or implicitly by a pointer to the related object. The number of attributes allowed for each object or link can be none, exactly one, or many. The input database can consist of a single connected component, or a set of connected components.

The output of a relational learning algorithm is a pattern (using the term loosely) that expresses a generalization supported by the input data. The scale of the pattern might be a single object, or a structure consisting of a group of related objects and the links that connect them. All patterns produced by a relational learning algorithm are descriptive because they

capture regularities of the input data; some patterns can also be used to make predictions about unseen data.

Categorizing the existing unsupervised relational learning algorithms helps us to understand how each algorithm relates to the others (no pun intended). Our goals in developing this categorization are

- to establish a common vocabulary in which to express the similarities and differences of relational learning algorithms;
- to identify interesting areas of unsupervised relational learning that are currently underdeveloped.

3 Three example algorithms

We illustrate our multi-dimensional categorization of unsupervised relational learning algorithms by comparing three systems that differ widely in their input and output formats.

The WARMR algorithm [Dehaspe *et al.*, 1998; Dehaspe and Toivonen, 2001] finds relational association rules or, to use the vocabulary of the authors, *query extensions*. The algorithm takes as input a Prolog database and a specification (in the WARMODE language) that limits the format of possible query extensions. The output of WARMR is a set of query extensions, all of which refer to the object designated as the *key* parameter. The query extensions are not limited to attributes of the key object, but can include its links to other objects and their attributes.

The SUBDUE system [Cook and Holder, 1994] iteratively discovers repeated substructures in a graph and compresses the graph by replacing the repeated substructure with a single vertex. The algorithm takes as input a labeled graph and a set of rules intended to bias the search process toward structures that are deemed more interesting. SUBDUE returns as output the substructure selected at each iteration as the best to compress the graph.

Probabilistic Relational Models (PRM) reinterpret Bayesian networks in a relational setting. PRMs have been evolving rapidly over the past few years; we focus here on the version described in [Getoor *et al.*, 2002]. A PRM captures the probabilistic dependence between the attributes of interrelated objects. It can also model uncertainty about the link structure. *Reference uncertainty* means we know how many links there are in the graph, but we don't know what their endpoints are. *Existence uncertainty* means we don't know how many links there are and have to consider the possibility that any pair of objects (of the appropriate types) might be linked. The input to the PRM learning algorithm is a database schema (specifying objects, links, and attributes) and an instantiation of that schema (a set of relational tables).

4 Input criterion of comparison

The first criterion of comparison concerns the input to the unsupervised relational learning algorithm. Our three example algorithms have very different data representations, but conceptually we can view their input in terms of objects and links. For SUBDUE the mapping is straightforward: objects correspond to vertices in the graph, and links to edges. SUBDUE requires exactly one attribute on each object and link in the graph: a label.

In the Inductive Logic Programming approach of WARMR, the input data are a set of Prolog facts, describing both objects and links. The predicate name is the equivalent of a type attribute. For example (from [Dehaspe and Toivonen, 2001, p. 191]), a fact such as

customer(allen).

represents an object of type **customer** with identifier **allen**. A fact such as

parent(allen, bill).

represents a link of type **parent** between the **allen** object and the **bill** object. The WARMR data model allows both objects and links to have multiple attributes besides type, which would be represented by additional arguments to the **customer** and **parent** predicates.

Our use of the terms “object” and “link” does not coincide with the terminology of [Getoor *et al.*, 2002]. What we call an object corresponds to the instantiation of an *entity class* in the PRM. What we call a link corresponds to the instantiation of a *relationship class*. The *reference slots* of the relationship class tell us the endpoints of the link. Both entity classes and relationship classes can have *descriptive attributes*, which we would simply call attributes. So any object or link in the PRM input can have multiple attributes. Could there be a link with no attributes? No. Even if the relationship class has only reference slots and no descriptive attributes, we still say that the link has one type attribute because in the PRM we know to what class this link belongs. For example, the PRM for the citation domain has a class representing the “cites” relationship between one paper and another. This is equivalent in our vocabulary to a link of type “cites” going from the citing paper to the cited paper. Keeping this translation of terminology in mind, we conclude that every object and link in the PRM's input has at least one attribute, its type, and possibly more.

5 Output criterion of comparison

The second criterion of comparison concerns the output produced by the unsupervised relational learning algorithm. Does the algorithm discover patterns at the level of individual objects, or at the level of subgraphs? (By “subgraph” we mean a structure containing at least one link with its associated objects.) SUBDUE searches for repeated substructures using an approximate graph match, and at each iteration returns the substructure which achieves the maximum graph compression when it is collapsed to a vertex. These are certainly patterns at the subgraph level. PRMs also discover patterns at the subgraph level. The result of training a PRM is an estimate of the joint probability distribution of attribute values (and link structure, in the case of reference or existence uncertainty) over the entire network.

Relational association rules are in a gray area. The WARMR algorithm requires that some predicate be designated as the key. All query extensions must contain the key predicate. For example, if **customer** is the key then all the rules will be about customers. (A link predicate such as **parent** can also be designated the key.) The association rules mention other objects to which the customer is linked, and the

attributes of those related objects. So all the discovered patterns concern the key object (or link) but can draw upon the relational neighborhood surrounding the key.

6 Predictive criterion of comparison

Generally the goal of an unsupervised learning algorithm is descriptive. We hope that the discovered patterns capture the essential regularities of the input dataset. However, for some algorithms it is possible to make predictions about new inputs based on the patterns observed in the training data. Relational association rules could be applied to make predictions about the key object (or link). As noted in Section 1, PRMs can be used for classification [Taskar *et al.*, 2001; Getoor *et al.*, 2002]. SUBDUE's output cannot be exploited for prediction. There is no reason to assume the substructure that provides maximum compression in one input graph would do the same in another graph.

7 Conclusion

We have presented one approach to categorizing unsupervised relational learning algorithms, and applied it to three examples. These same criteria of comparison would be relevant for other algorithms we have not discussed, such as frequent subgraph discovery [Kuramochi and Karypis, 2001], and stochastic link and group detection [Kubica *et al.*, 2002]. We aim to establish a common vocabulary in which we can compare systems that have very different input/output specifications. Categorizing the current algorithms helps us identify important gaps in unsupervised relational learning that could be fruitful areas for future research.

Acknowledgments

We are indebted to the discussions of the Structuring Data working group of the Knowledge Discovery Laboratory at the University of Massachusetts Amherst. Members of this group are Andrew Fast, Lisa Friedland, Michael Hay, David Jensen, and Jen Neville, all of U Mass Amherst, and Pat Riddle of the University of Auckland.

References

- [Cook and Holder, 1994] Diane J. Cook and Lawrence B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.
- [Dehaspe and Toivonen, 2001] Luc Dehaspe and Hannu Toivonen. Discovery of relational association rules. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining*, pages 189–212. Springer-Verlag, 2001.
- [Dehaspe *et al.*, 1998] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. In R. Agrawal, P. Stolorz, and G. Piattetsky-Shapiro, editors, *4th International Conference on Knowledge Discovery and Data Mining*, pages 30–36. AAAI Press., 1998.

- [Getoor *et al.*, 2001] Lise Getoor, Nir Friedman, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.
- [Getoor *et al.*, 2002] Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707, 2002.
- [Jonyer *et al.*, 2001] Istvan Jonyer, Diane J. Cook, and Lawrence B. Holder. Graph-based hierarchical conceptual clustering. *Journal of Machine Learning Research*, 2:19–43, 2001.
- [Kubica *et al.*, 2002] Jeremy Kubica, Andrew Moore, Jeff Schneider, and Yiming Yang. Stochastic link and group detection. In *The Eighteenth National Conference on Artificial Intelligence*, pages 798–804, Jul 2002.
- [Kuramochi and Karypis, 2001] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *ICDM*, pages 313–320, 2001.
- [Taskar *et al.*, 2001] Benjamin Taskar, Eran Segal, and Daphne Koller. Probabilistic classification and clustering in relational data. In Bernhard Nebel, editor, *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, pages 870–878, Seattle, US, 2001.
- [Taskar *et al.*, 2002] Benjamin Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, Edmonton, Canada, 2002.