Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction

Jun Zhang, Adrian Silvescu, and Vasant Honavar

Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1040
USA
{jzhang, silvescu, honavar}@cs.iastate.edu
http://www.cs.iastate.edu/~honavar/aigroup.html

Abstract. Most learning algorithms for data-driven induction of pattern classifiers (e.g., the decision tree algorithm), typically represent input patterns at a single level of abstraction – usually in the form of an ordered tuple of attribute values. However, in many applications of inductive learning – e.g., scientific discovery, users often need to explore a data set at multiple levels of abstraction, and from different points of view. Each point of view corresponds to a set of ontological (and representational) commitments regarding the domain of interest. The choice of an ontology induces a set of representatios of the data and a set of transformations of the hypothesis space. This paper formalizes the problem of inductive learning using ontologies and data; describes an ontology-driven decision tree learning algorithm to learn classification rules at multiple levels of abstraction; and presents preliminary results to demonstrate the feasibility of the proposed approach.

1 Introduction

Inductive learning algorithms (e.g., decision tree learning) offer a powerful approach to data-driven discovery of complex, a-priori unknown relationships (e.g., classifiers) from data. Most learning algorithms for data-driven induction of pattern classifiers (e.g., the decision tree algorithm), typically represent input patterns at a single level of abstraction – usually in the form of an ordered tuple of attribute values. They typically assume that each pattern belongs to one of a set of disjoint classes. Thus, any relationships might exist between the different values of an attribute or relationships between classes (e.g., a hierarchically nested class structure) are ignored. In contrast, data-driven knowledge discovery in practice, occurs within a *context*, or under certain *ontological commitments* on the part of the learner. The learner's ontology (i.e., assumptions concerning *things* that exist in the *world*) determines the choice of *terms* and *relationships* among terms (or more generally, *concepts*) that are used to describe the domain of interest and their intended correspondence with objects and properties of the world [11]. This is particularly true in scientific

S. Koenig and R. Holte (Eds.): SARA 2002, LNAI 2371, pp. 316–323, 2002.

applications of machine learning where specific ontological and representational commitments often reflect prior knowledge and working assumptions of scientists. When several independently generated and managed data repositories are to be used as sources of data in a learning task, the ontological commitments that are implicit in the design of the data repositories may or may not correspond to those of the user (typically a scientist familiar with the domain e.g., a biologist) in a given context [6], [7]. For example, in one context, the scientist may not consider it necessary to distinguish between different subfamilies of a family of proteins or different types of sequence patterns or structural features of proteins. In other cases, such distinctions may be desirable. In computational characterization of protein sequence-structure-function relationships, it is often useful to consider alternative representations of protein sequences and different notions of protein function [2]. In scientific discovery applications, because users often need to examine data in *different contexts from different perspectives* and *at different levels of abstraction*, there is no single universal ontology that can serve all users, or for that matter, even a single user, in every context. Hence, methods for learning from ontologies and data are needed to support knowledge acquisition from heterogeneous distributed data.

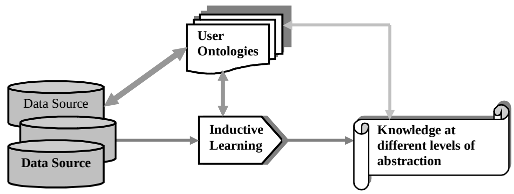

Fig. 1. Ontology-driven inductive learning

Making ontological commitments (that are typically implicit in a data set) explicit enables users to explore data from multiple perspectives, and at different levels of abstraction. Some aspects of ontology guided learning have received attention in the literature. Walker [13] first used the concept taxonomies in information retrieval from large database. Han et al. [4] proposed attribute-oriented induction of multi-level classification rules using background knowledge in the form of concept hierarchies. They also proposed a method to discover association rules at multiple levels of abstraction. Quinlan [9] suggested pre-processing approaches to deal with tree-structured attributes (ISA hierarchy), re-encoding the training examples in terms of an equivalent set of purely nominal attributes. Almuallim [1] proposed handling tree-structured attributes directly by routing examples in hierarchies, which count the class frequency of every concept node, then apply decision tree learning algorithm to score and find the best concept node in the hierarchy to build the decision tree. Taylor et al [12] proposed an algorithm for rule learning using taxonomies and data.

Against this background, it is of interest to formalize the problem of learning from ontologies and data and to explore the design space of algorithms for data-driven knowledge acquisition using *explicitly specified* ontologies. In this paper, we formalize the problem of inductive learning using ontologies (as a form of background knowledge or working assumptions) and data. We present an ontology-driven decision tree learning algorithm to learn classification rules at multiple levels of abstraction. We present some preliminary results to demonstrate the feasibility of the proposed approach. We briefly examine several variations of the proposed approach and a general strategy for transforming traditional inductive learning algorithms into ontology-guided inductive learning algorithms for data-driven discovery of relationships at multiple levels of abstraction.

2 Role of Ontologies in Learning from Data

An ontology specifies the terms or concepts and relationships among terms and their intended correspondence to objects and entities that exist in the world [3], [11]. A formal ontology is specified by a collection of names for concept and relation types organized in a partial ordering by the type-subtype relation [11]. In philosophy, an ontology corresponds to a nomenclature of all things (entities, properties of entities, and relations among entities) that exist and as such, there is no room for multiple ontologies. However, such a view is untenable in practice. Consequently, we adopt the position that an ontology corresponds to a particular conceptualization of the world from a specific point of view.

Syntactically, given a logical language L, an ontology is a tuple $< V, A >$, where the vocabulary $V \subset S_p$ is some subset of the predicate symbols of$L$ and the axioms $A \subset W$ are a subset of the well-formed formulas of L [5]. Taxonomies that specify hierarchical relationships among concepts in a domain of interest are among some of the most commonly used ontologies. Normally, we would draw this mapping in the form of a tree, or a directed acyclic graph (DAG).

As is usually the case in formulations of inductive learning problems, we will assume that each instance is described by a tuple of attribute values and that a concept corresponds to a set of instances that satisfy specific constraints on the values of their attributes. Note that there is a certain duality between attributes and concepts. For instance, instances in a particular domain may be described in terms of two attributes: *color* and *size*. Now, *blue*, a possible value for the attribute *color* is itself a concept (composed out of all the instances that have color blue). Thus, it is possible for each attribute to have an ontology associated with it.

In what follows, we consider several cases (ordered by their complexity) in which ontologies may play a role in learning from data.

a) Each attribute has associated with it, an ontology in the form of a hierarchy. A hierarchical ontology is the simplest form of ontology, and is typically represented by a tree. Every node in the tree represents a concept. The topmost root concept is the name of the corresponding attribute. Every link in the tree represents an interrelationship between two nodes. The interrelationship between concepts could be the relation of ISA, Instance-Of, or Part-Of. For an attribute with a finite domain of possible values, each value corresponds to a primitive concept (or a leaf node) in a hierarchical ontology. An attribute without a hierarchically structured ontology corresponds to a single level taxonomy with attribute name as root, and the attribute values as the children leaf nodes of root.

b) Each attribute has a single ontology associated with it. However, concepts that appear in ontologies associated with different attributes can be related. This results in ontologies that can be represented using directed acyclic graphs (DAGs). (this would yield more complicated ontologies than a set of competing hierachies).

The next section describes an ontology-driven decision tree construction algorithm for the first of the four cases outlined above.

3 Ontology-Guided Decision Tree Learning Algorithm

We consider decision tree learning in a scenario in which each attribute has a single hierarchically structured ontology (e.g., a concept taxonomy) associated with it and each instance is labeled with one of m disjoint class labels. Ontology-driven Decision Tree (ODT) learning algorithm is a top-down multi-level ontology (concept hierarchy) guided search in a hypothesis space of decision trees.

Recall that the basic decision tree algorithm recursively selects at each step, an attribute from a set of candidate attributes based on an information gain criterion [9]. Thus, each node in a partially constructed decision tree has associated with it, a set of candidate attributes to choose from for growing the tree rooted at that node.

In our case, each attribute has associated with it, a hierarchically structured taxonomy over possible values of the attribute. Thus, the learning algorithm has to choose not just a particular attribute, but also an appropriate level of abstraction in the taxonomy. The basic idea behind the algorithm is to start with abstract attributes (i.e., groupings of attribute values that corresponds to nodes that appear at higher levels of a hierarchically structured attribute value taxonomy). Thus, each node of a partially constructed decision tree has associated with it, a set of candidate attributes drawn from the taxonomies associated with each of the individual attributes. The algorithm maintains for each node in the partially constructed decision tree, a set of pointers to nodes on the frontier, computes information gain for the corresponding attributes, and selects from the set of candidate attributes under consideration, one with the largest information gain.

It is useful to introduce some notation to help describe our algorithm. Let the set of attributes used to describe instances in the data set be $A = \{A_1, A_2, ..., A_n\}$. Let the set of class labels be $O = \{O_1, O_2, ..., O_m\}$. Each attribute Ai, has associated with it, a corresponding taxonomy T_i. The set of ontologies is denoted by $T = \{T_1, T_2, ..., T_n\}$. Note that the root node of taxonomy T_i is A_i. Let $\Psi(c)$ denote the children of a node c. The training data set is denoted by S. Each leaf node of a partially constructed decision tree has associated with it, a subset of the training data set. We will associate with each such set of examples,

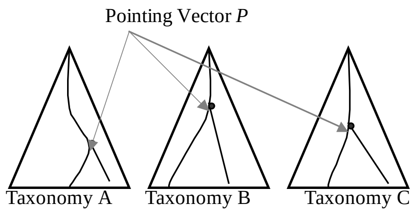

Fig. 2. Vector of Pointers: concept frontier across different attribute taxonomies

n pointers that point to n concepts in n taxonomies (one for each attribute). Let $P = \{p_1, p_2, ..., p_n\}$ denote such a vector of pointers where p_i is concept in taxonomy T_i. A vector of pointers is called an *ending vector* if each p_i is a leaf node in the corresponding taxonomy T_i. We use $\Phi(P) = true$ to denote that P is an ending vector. Note that $\Phi(P) = true$ if and only if $\forall p_i \in P$, $\Psi(p_i) = \{\}$.

Our current implementation selects an attribute from a set of candidate attributes (specified by the vector of pointers) that yields the maximum reduction in entropy as the splitting criterion to partition the dataset [9]. However, other splitting criteria could also be adopted directly in our algorithm (e.g. Gini Index, one-sided purity, one-sided extremes, etc).

ODT algorithm:

ODT (Examples S, Attributes A, Taxonomies T, Class labels O, Vector of Pointers P, Default Df)

1. If decision tree is $NULL$ Then create a root node for decision tree, set all examples to S, and set $P = \{A_1, A_2, ..., A_n\}$, and set $Df = Majority\_Class(S)$
2. If S is empty Then assign the label Df to the node.
   Else If every instance in S has the same class label o Then Return(o)
   Else If $\Phi(P)$ is true Then assign the label Df to the node.
3. Calculate the best attribute B_j and best concept b by calling function *Choose-Best(P, S, A, O)*
4. Set the best partition value set $Bvalue = \Psi(b)$
5. Partition the examples S using the concepts in $Bvalue$
   For each value V_i in Bvalue Do
      S_i = subset of S with concept V_i
      j = order of B_j in A
      update the Pointing Vector P to P' by substituting p_j for V_i
      construct the subtree by calling $ODT(S_i, A, O, P', Majority\_Value(S_i))$
      add new branch with label V_i and connect to its subtree.
   End
6. Return the Decision Tree

Choose-Best(PointingVectors P, Examples S, Attributes A, Class labels O)
/* Returns the attribute whose expansion will yield the best information gain, selected from P = $\{p_1, ...p_n\}$ */
Return $\operatorname*{argmax}_i \; Gain(S, p_i)$

The ontology-driven decision tree algorithm can be viewed as a best-first search through the hypothesis space of decision trees defined with respect to a set of attribute taxonomies.

3.1 Illustration of the Working of the ODT Algorithm

The preliminary test of our algorithm is based on a simple customer purchase database that used in Taylor's paper [12]. Each of the attributes used to describe instances in this data set has a taxonomy associated with it. The two taxonomies are ISA hierarchies for Beverage and Snack. For concepts in the Beverage taxonomy, there are three different levels of abstraction, and in the Snack taxonomy, we have two different levels of abstraction. The class has three values: Young, Middle-aged, and Old. Figure 3 shows the two taxonomies, and Figure 4 shows the dataset and the induced tree.

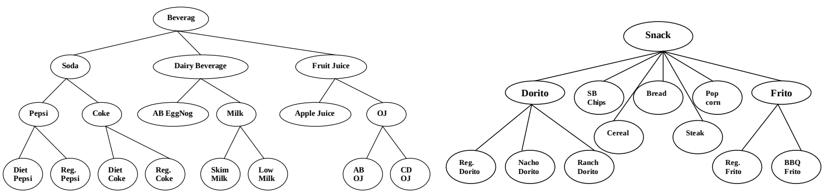

Fig. 3. Two taxonomies defined over two attributes

Customer	Item1	Item 2	Class
1	Diet Coke	Ranch Dorito	Young
2	AB OJ	CD Cereal	Old
3	Reg Coke	Reg Dorito	Young
4	Reg Coke	SB Chips	Mid-Aged
5	Diet Coke	Nacho Dorito	Young
6	Diet Pepsi	BBQ Frito	Mid-Aged
7	Reg Pepsi	Reg Frito	Mid-Aged
8	Skim Milk	CD Cereal	Old
9	Reg Pepsi	BBQ Frito	Mid-Aged
10	CD OJ	Bread	Old
11	Reg Pepsi	Popcorn	Young
12	AB Egg Nog	CD Steak	Old

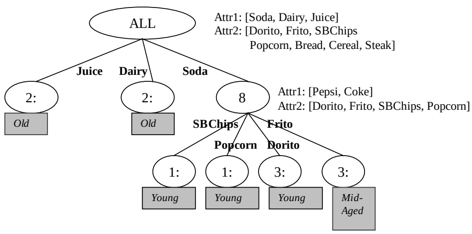

Fig. 4. Sample customer purchase database and the induced decision tree

We start our search with the Pointing Vectors pointing to the root of both taxonomies. The information gain associated with the attribute Beverage is higher than that associated with Snack. Consequently, the attribute corresponding to the root of the Beverage hierarchy is selected to partition the original data. This yields a 3-way split at the root of the decision tree. Two examples are classified as Old on the basis of the attribute corresponding to the concept "Dairy Beverage", and two examples are classified as Old on the basis of the attributed corresponding to the concept concept "Fruit Juice". The remaining eight examples need to be partitioned further. The first element p_1 in Pointing Vector for these eight examples changes to "Soda" in the Beverage Taxonomy, and hence the possible choice of attribute values will be [Pepsi, Coke]. While the second element p_2 continues to the root of the Snack Taxonomy, the possible attribute value set will include all the available concepts in the first level of this taxonomy.

This time p_2 yields a better value for information gain and all eight examples are correctly classified. The resulting decision tree corresponds to the rules: *If* Soda and Dorito *Then* Young; *If* Soda and Frito *Then* Midle-aged; *If* Dairy *Then* Old; *If* Juice *Then* Old. Note that many of the rules discriminate among instances belonging to the three classes using attributes that correspond to concepts that reside at higher levels of the corresponding attribute taxonomies.

4 Summary and Discussion

Many practical applications of machine learning (e.g., data-driven scientific discovery in computational biology [2] call for exploration of a data set from multiple perspectives (that correspond to multiple ontologies). Different ontologies induce different representations of the data and transformations of the hypothesis space. For example, hierarchically structured taxonomies over values of attributes facilitate discovery of classifiers at different levels of abstraction. The work described in this paper represents a tentative first step toward formulating the problem of ontology-guided data-driven knowledge discovery. We have demonstrated an extension of the standard decision tree learning algorithm that can exploit user-supplied ontologies to induce classification rules at higher levels of abstraction.

Work in progress is aimed at:

a) Systematic experimental evaluation of the proposed algorithm on real-world data sets that are encountered in computational biology (e.g., data-driven characterization of macromolecular sequence-structure-function relationships), text classification, among others.
b) Extensions of the proposed approach to accommodate use of multiple hierarchically structured ontologies for each attribute, as well as DAG-structured (as opposed to tree-structured) ontologies.

In related work, we are exploring ontology-guided learning algorithms for domains in which:

a) Each class (target attribute) has a tree-structured concept hierarchy (e.g., a taxonomy) associated with it. For example, in a pattern classification task, it may be necessary to classify an instance at different levels of abstraction (e.g., a soft drink into Pepsi, Cola, carbonated beverage). Higher level concepts correspond to generalizations of the basic level target concepts.
b) Each instance may belong to more than one class. (For instance, an individual may be classified as a parent, student, wife, friend. A protein may have multiple not necessarily mutually exclusive functions) [10].
c) There is a need to integrate information from multiple heterogeneous, autonomous, distributed data and knowledge sources from different ontological view points [6],[7] (e.g., in scientific discovery environments).

Acknowledgements. This research was supported in part by grants from the National Science Foundation (9982341, 0087152, 9972653) and Pioneer Hi-Bred to Vasant Honavar and the Iowa State University (ISU) Graduate College. This research has benefited from discussions with members of the ISU Artificial Intelligence Research Laboratory.

References

1. Almuallim H., Akiba, Y., Kaneda, S.: On Handling Tree-Structured Attributes. Proceedings of the Twelfth International Conference on Machine Learning (1995)
2. Andorf, C., Dobbs, D., Honavar, V.: Discovering Protein Function Classification Rules from Reduced Alphabet Representations of Protein Sequences. Proceedings of the Conference on Computational Biology and Genome Informatics. Durham, North Carolina (2002)
3. Gruber T. R.: A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199-220, (1993)
4. Han, J., Fu, Y.: Exploration of the Power of Attribute-Oriented Induction in Data Mining. U.M. Fayyad, el al. (eds.), Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996)
5. Heflin, J., Hendler, J., Luke, S.: Coping with Changing Ontologies in a Distributed Environment. Ontology Management. Papers from the AAAI Workshop. WS-99-13. AAAI Press (1999)
6. Honavar, V., Silvescu, A., Reinoso-Castillo, J., Andorf, C., Dobbs, D.: Ontology-Driven Information Extraction and Knowledge Acquisition from Heterogeneous, Distributed Biological Data Sources. Proceedings of the IJCAI-2001 Workshop on Knowledge Discovery from Heterogeneous, Distributed, Autonomous, Dynamic Data and Knowledge Sources. (2001)
7. Reinoso-Castillo, J.: An Ontology-Driven Query-Centric Approach to Information Integration from Heterogeneous, Distributed, Autonomous Data Sources. Masters Thesis. Artificial Intelligence Research Laboratory, Iowa State University, June 2002.
8. Kamber, M., Winstone, L., Gong, W., Cheng, S., Han, J.: Generalization and Decision Tree Induction: Efficient Classification in Data Mining, Proc. of 1997 Int'l Workshop on Research Issues on Data Engineering (RIDE'97) Birmingham, England, April (1997)
9. Quinlan, J. R.: C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann (1992)
10. Silvescu, A., Caragea, D., and Honavar, V.: Learning Decision Tree Classifiers When Classes are not Mutually Exclusive. To appear. (2002)
11. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA (2000)
12. Taylor, M., Stoffel, K., Hendler, J.: Ontology-based Induction of High Level Classification Rules. SIGMOD Data Mining and Knowledge Discovery workshop proceedings. Tuscon, Arizona (1997)
13. Walker, A.: On retrieval from a small version of a large database. VLDB Conference (1980)