Learning Ontology-Aware Classifiers

Jun Zhang, Doina Caragea, and Vasant Honavar

Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1040, USA
{jzhang, dcaragea, honavar}@cs.iastate.edu

Abstract. Many practical applications of machine learning in data-driven scientific discovery commonly call for the exploration of data from multiple points of view that correspond to explicitly specified ontologies. This paper formalizes a class of problems of learning from ontology and data, and explores the design space of learning classifiers from attribute value taxonomies (AVTs) and data. We introduce the notion of AVT-extended data sources and partially specified data. We propose a general framework for learning classifiers from such data sources. Two instantiations of this framework, AVT-based Decision Tree classifier and AVT-based Naïve Bayes classifier are presented. Experimental results show that the resulting algorithms are able to learn robust high accuracy classifiers with substantially more compact representations than those obtained by standard learners.

1 Introduction

Current advances in machine learning have offered powerful approaches to exploring complex, a-priori unknown relationships or discovering hypotheses that describe potentially interesting regularities from data. 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 [22]. This is particularly true in scientific discovery where specific ontological and representational commitments often reflect prior knowledge and working assumptions of scientists [8][27].

Hierarchical taxonomies over attribute values or classes are among the most common type of ontologies in practice. Examples of such ontologies include: Gene Ontology [3] that is a hierarchical taxonomy for describing many aspects of macromolecular sequence, structure and function; Hierarchical taxonomy built for features of intrusion detection [25]; Hierarchical groupings of attribute values for Semantic Web [5]; Hierarchies defined over data attributes in e-commerce applications of data mining [16].

Making ontological commitments (that are typically implicit in a data set) *explicit* enables users to explore data from different points of view, and at different levels of abstraction. Each point of view corresponds to a set of ontological (and representational) commitments regarding the domain of interest. In scientific discovery, there is no single perspective that can serve all purposes, and it is always helpful to analyze data in different contexts and from alternative representations. Hence, there is a need for ontology-aware learning algorithms to facilitate the exploration of data from multiple points of view.

Exploring ontology-aware learning algorithms can provide us with a better understanding of the interaction between data and knowledge. The availability of user-supplied ontologies (e.g., taxonomies) presents the opportunity to learn classification rules that are expressed in terms of familiar hierarchically related concepts leading to simpler, easier-to-comprehend rules [26]. Moreover, learning algorithms that exploit hierarchical taxonomies can potentially perform a built-in regularization and thereby yielding robust classifiers [27].

Against this background, it is of significant practical interest to precisely formulate the problem of learning from ontologies (as a form of background knowledge or working assumptions) and data, and to explore the design space of algorithms for data-driven knowledge acquisition using *explicitly specified* ontologies (such as taxonomies). In this paper, we formalize the problem of learning pattern classifiers from Attribute Value Taxonomies, and propose a general learning framework that takes into account the tradeoff between the complexity and the accuracy of the predictive models. According to this general framework, we presents two well-founded AVT-based variants of machine learning algorithms, including Decision Tree and Naïve Bayes classifiers. We present our experimental results, and conclude with summary and discussion.

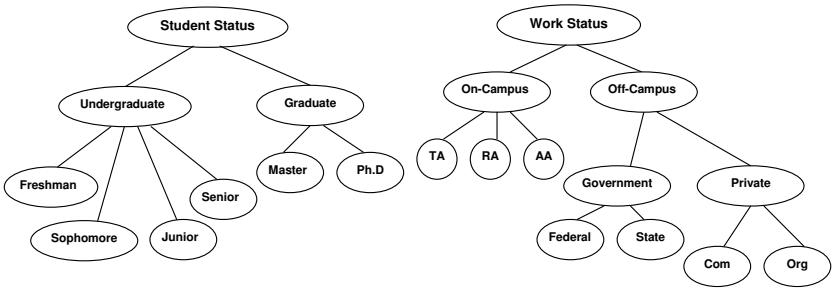
2 Problem Formulation

2.1 Ontology-extended Data Source

In supervised classification learning problems, the data to be explored are typically available as a set of labelled training instances $\{(X_p, c_{X_p})\}$ where X_p is an instance in instance space I, and c_{X_p} is the class label from $C = \{c_1, c_2, \cdots, c_M\}$, a finite set of mutually disjoint classes. Assume that D is the data set represented using an ordered set of attributes $A = \{A_1, A_2, \cdots, A_N\}$, and $O = \{\Lambda_1, \Lambda_2, \cdots, \Lambda_N\}$ be an ontology associated with the data set. The element $\Lambda_i \in O$ corresponds to the attribute A_i, and describes the type of that particular attribute. In general, the type of an attribute can be a standard type (e.g., Integer or String) or a hierarchical type, which is defined as an ordering of a set of terms (e.g., attribute values). The schema S of the data set D is given by the set of attributes $\{A_1, A_2, \cdots, A_N\}$ used to describe the data together with their respective types $\{\Lambda_1, \Lambda_2, \cdots, \Lambda_N\}$ described by the ontology O. Caragea et al [8] defined *ontology-extended data source* to be expressed as $\mathcal{D} = \langle D, S, O \rangle$, where D is the data set, S is the schema of the data and O is the ontology

associated with the data source. The instance space I where D is sampled can be defined as $I = \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_N$

In the discussion that follows, we focus on hierarchical ontologies in the form of attribute value taxonomies (AVTs). Typically, attribute values are grouped into a hierarchical structure to reflect actual or assumed similarities among the attribute values in the domain of interest. We use $T = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$ to represent the ordered set of attribute value taxonomies associated with attributes $A_1, A_2, \ldots, A_N$. Thus, an AVT defines an abstraction hierarchy over values of an attribute. Figure 1 shows an example of two AVTs, together with a sample data set collected by a university department based on the corresponding AVTs.



Student ID	Student Status	Work Status	Hourly Income	Internship
60-421	Freshman	Org	$10/hr.	No
73-727	Master	Com	$30/hr.	Yes
81-253	Ph.D	RA	$20/hr.	No
75-455	Graduate	On-Campus	$20/hr.	No
32-719	Sophomore	AA	$15/hr.	No
42-139	Senior	Government	$25/hr.	Yes
66-338	Undergraduate	Federal	$25/hr.	Yes
.....	.....	.....	.....	.....

Fig. 1. Two attribute value taxonomies on student status and work status and a sample data set based on the two corresponding AVTs.

Specifically, we use *AVT-extended data source* $\mathcal{D} = \langle D, S, T \rangle$ to refer to the special case of ontology-extended data source where ontology is a set of attribute value taxonomies.

2.2 AVT-Induced Instance Space

In many real world application domains, the instances from AVT-extended data sources are often specified at different levels of precision. The value of a particular

attribute or the class label associated with an instance or both are specified at different levels of abstraction with regard to the hierarchical taxonomies, leading to *partially specified instances* [27]. Partially specified data require us to extend our definition of instance space. We give formal definitions on partially specified data and AVT-induced instance space in the following.

Attribute value taxonomies enable us to specify a level of abstraction that reflects learner's perspective on the domain.

Definition 1 (Cut [14]). *A cut γ_i is a subset of elements in $Nodes(\mathcal{T}_i)$ satisfying the following two properties:* (1) *For any leaf $m \in Leaves(\mathcal{T}_i)$, either $m \in \gamma_i$ or m is a descendant of an element $n \in \gamma_i$; and* (2) *For any two nodes $f, g \in \gamma_i$, f is neither a descendant nor an ancestor of g.*

Definition 2 (Global Cut). *Let Δ_i be the set of all valid cuts in $\mathcal{T}_i$ of attribute A_i, and $\Delta = \times_{i=1}^{N} \Delta_i$ be the cartesian product of the cuts through the individual AVTs. $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_N\}$ defines a global cut through $T = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N\}$, where each $\gamma_i \in \Delta_i$ and $\Gamma \in \Delta$.*

Any global cut Γ in Δ specifies a level of abstraction for $\mathcal{D} = \langle D, S, T \rangle$. We use AVT frontier to refer to a global cut that is specified by the learning algorithm. In terms of a certain level of abstraction (i.e., a global cut Γ), we can precisely define fully specified instance and partially specified instance:

Definition 3 (Partially Specified Instance [27]). *If Γ represents the current level of abstraction in learner's AVT and $X_p = (v_{1p}, v_{2p}, ..., v_{Np})$ is an instance from D, then X_p is:*

- *Fully specified with respect to Γ, if $\forall i$, v_{ip} is on or below the cut Γ.*
- *Partially specified with respect to Γ, if $\exists v_{ip} \in X_p$, v_{ip} is above the cut Γ.*

When attribute value v_{ip} is below the specified cut Γ, it is fully specified because there is always a corresponding value on the cut that can replace the current value in the current level of abstraction. However, when v_{ip} is above the cut, there are several descendant values on the cut. It is uncertain which value will be the true attribute value, and hence partially specified. A particular attribute value can dynamically switch between being a fully specified value and being a partially specified value when the level of abstraction changes. For example, the shaded instances in Figure 1 are partially specified if the global cut Γ chooses to be all primitive attribute values in the corresponding AVTs.

The original instance space I is an instance space relative to a global cut Γ_ϕ with a domain of all primitive attribute values (all leaf-nodes in AVTs). Because any choice Γ defines a corresponding instance space I_Γ that is an abstraction of the original instance space I_{Γ_ϕ}, we can formally define AVT-induced instance space as follows.

Definition 4 (AVT-Induced Instance Space [28]). *A set of AVTs $T = \{T_1 \cdots T_N\}$ associated with a set of attributes $A = \{A_1 \cdots A_N\}$ induces an instance space $I_T = \cup_{\Gamma \in \Delta} I_\Gamma$ (the union of instance spaces induced by all of the cuts through the set of AVTs T).*

Therefore, a partially specified data set D_T is a collection of instances drawn from I_T where each instance is labeled with the appropriate class label from C. Thus, $D_T \subseteq I_T \times C$. Taking into account partially specified data, AVT-extended data source becomes $\mathcal{D} = \langle D_T, S, T \rangle$.

2.3 Learning Classifiers from Ontology-extended Data Source

The problem of learning classifiers from data can be described as follows: Given a data set D, a hypothesis class H, and a performance criterion P, the classifier learner L generates a hypothesis in the form of a function $h : I \rightarrow C$, where $h \in H$ optimizes P. For example, we search for a hypothesis h that is most likely given the training data D.

Learning classifiers from an ontology-extended data set is a generalization of learning classifiers from data. The typical hypothesis class H has been extended to H_O, where the original hypothesis language has been enriched by ontology O. The resulting hypothesis space H_O is a much larger space. In the case where the ontology is a set of attribute value taxonomies, the hypothesis space changes to H_T, a collection of hypothesis classes $\{H_\Gamma | \Gamma \in \Delta\}$. Each H_Γ corresponds a hypothesis class with regard to a global cut Γ in the AVTs. Because partial ordering exists among global cuts, it is obvious that the resulting hypothesis space H_T also has partial ordering structure.

The problem of learning classifiers from AVT-extended data can be stated as follows: Given a user-supplied set of AVTs T and a data set D_T of (possibly) partially specified labeled instances, construct a classifier $h : I_T \rightarrow C$ for assigning appropriate class labels to each instance in the instance space I_T. It is the structure of the hypothesis space H_T that makes it possible to search the space efficiently for a hypothesis h that could be both concise and accurate.

3 AVT-Based Classifier Learners

We describe in the following a general framework for designing algorithms to learn classifiers from AVT-extended data sources. Base on this framework, we demonstrate our approach by extending standard decision tree classifier and Naïve Bayes classifier.

3.1 A General Learning Framework

There are essentially three elements in learning classifiers from AVT-extended data sources: (1) A procedure for identifying estimated sufficient statistics on AVTs from data; (2) A procedure for building and refining hypothesis; (3) A performance criterion for making tradeoff between complexity and accuracy of the generated classifiers. In what follows, we discuss each element in details.

(1) Identifying Estimated Sufficient Statistics

Building a classifier only needs certain *statistics* (i.e., a function of data). A statistic $\mathcal{S}(D)$ is called a *sufficient statistic* for a parameter θ if $\mathcal{S}(D)$ provides all the information needed for estimating the parameter θ from data D. We can formally define sufficient statistic for a learning algorithm.

Definition 5 (Sufficient Statistic for a Learning Algorithm [7]). *We say that $\mathcal{S}_L(D)$ is a sufficient statistic for learning the hypothesis h using a learning algorithm L if there exists a procedure that takes $\mathcal{S}_L(D)$ as input and outputs h.*

For many learning algorithms, sufficient statistics are frequency counts or class conditional frequency counts for attribute values. Given a hierarchical structured AVT, we can define a tree of frequency counts or class conditional frequency counts as the sufficient statistics for the learning algorithms. More specifically, with regard to an attribute value taxonomy $\mathcal{T}_i$ for attribute A_i, we define a tree of class conditional frequency counts $CCFC(\mathcal{T}_i)$ (and similarly, a tree of frequency counts $FC(\mathcal{T}_i)$).

If all the instances are fully specified in AVT-extended data source, the class conditional frequency counts associated with a non leaf node of $CCFC(\mathcal{T}_i)$ should correspond to the aggregation of the corresponding class conditional frequency counts associated with its children. $CCFC(\mathcal{T}_i)$ can be computed in one upward pass. When data are partially specified in AVT-extended data source, we can use a 2-step process for computing $CCFC(\mathcal{T}_i)$ [28]: First we make an upward pass aggregating the class conditional frequency counts based on the specified attribute values in the data set; Then we propagate the counts associated with partially specified attribute values down through the tree, augmenting the counts at lower levels according to the distribution of values along the branches based on the subset of the data for which the corresponding values are fully specified. This procedure can be seen as a special case of EM (Expectation Maximization) algorithm [11] to estimate sufficient statistics for $CCFC(\mathcal{T}_i)$.

(2) Building and Refining Hypothesis

As we have mentioned earlier, for a particular global cut Γ there is a corresponding hypothesis class H_Γ, and we can learn a hypothesis $h(\theta|\Gamma)$ with parameters θ from this hypothesis class H_Γ using a learning algorithm L. The total number of global cuts $|\Delta|$ grows exponentially with the scale of AVTs, and so does the number of possible hypotheses. Since an exhaustive search over the complete hypothesis space $\{H_\Gamma|\Gamma \in \Delta\}$ is computationally infeasible, we need a strategy to search through the resulting hypothesis space.

Following the definition of cut, we can define a refinement operation on cut as follows:

Definition 6 (Cut Refinement [28]). *We say that a cut $\hat{\gamma}_i$ is a refinement of a cut γ_i if $\hat{\gamma}_i$ is obtained by replacing at least one attribute value $v \in \gamma_i$ by its descendant attribute values. A global cut $\hat{\Gamma}$ is a refinement of a global cut Γ if at least one cut in $\hat{\Gamma}$ is a refinement of a cut in Γ.*

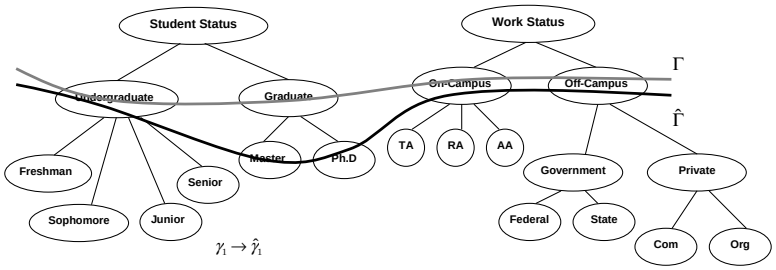


Fig. 2. Cut refinement. The cut γ_1={*Undergraduate*, *Graduate*} in the *student status* attribute has been refined to $\hat{\gamma}_1$={*Undergraduate*, *Master*, *Ph.D*}, and the global cut Γ has been refined to $\hat{\Gamma}$.

Figure 2 shows a demonstrative cut refinement process based on the AVTs shown in Figure 1. When $\hat{\Gamma}$ is a cut refinement of Γ, the corresponding hypothesis $h(\hat{\Gamma})$ is a *hypothesis refinement* of $h(\Gamma)$. Hypothesis refinements in AVT-based learning are conducted through cut refinements in AVTs.

Based on gathered sufficient statistics, our goal is to search for the optimal hypothesis $h(\Gamma^*)$ from $\{H_\Gamma | \Gamma \in \Delta\}$, where Γ^* is an optimal level of abstraction (i.e., an optimal cut) that is decided by the learning algorithm L using certain performance measurement P.

We use a top-down refinement on the global cut to greedily explore the design space of the corresponding classifier. Our general strategy is to start by building a classifier that is based on the most abstract global cut and successively refine the classifier (hypothesis) by cut refinement. Therefore, the learning algorithm L generates a sequence of cut refinements $\Gamma_0, \Gamma_1, \cdots, \Gamma^*$, which corresponds to a sequence of hypothesis refinements $h(\Gamma_0), h(\Gamma_1), \cdots, h(\Gamma^*)$, until a final optimal cut Γ^* and an optimal classifier $h(\Gamma^*)$ is obtained.

(3) Trading off the Complexity against the Error

For almost every learning algorithm L, there is a performance measurement P that is explicitly or implicitly optimized by L. For example, some performance measurements include predictive accuracy, statistical significance tests, and many information criteria. However, the lack of good performance measurement makes the learning algorithm to build over complex model as the classifier that shows excellent performance on training data but poor performance on test data. This problem is called overfitting, which is a general problem that many learning algorithms seek to overcome.

Of particular interest to us are those criteria that can make tradeoffs between the accuracy and the complexity of the model [2][21], thereby having a built-in mechanism to overcome overfitting. For example, Minimum Description Length (MDL) principle [21] is to compress the training data D and encode it by a hypothesis h such that it minimizes the length of the message that encodes both

h and the data D given h. By making this tradeoff, we are able to learn classifiers that is both compact and accurate.

In order to perform hypothesis refinements effectively, we need a performance criterion P that can decide if we need to make a refinement from $h(\Gamma)$ to $h(\hat{\Gamma})$. Also this criterion should be able to decide whether we should stop making refinement and output a final hypothesis as the classifier.

The performance criterion P is applied in the calculation of sufficient statistics for hypothesis refinement that is defined as follows.

Definition 7 (Sufficient Statistics for Hypothesis Refinement[7]). *We denote $\mathcal{S}_L(D, h_i \rightarrow h_{i+1})$ as the sufficient statistic for hypothesis refinement from h_i to h_{i+1}, if the learner L accepts h_i and a sufficient statistic $\mathcal{S}_L(D, h_i \rightarrow h_{i+1})$ as inputs and outputs an updated hypothesis h_{i+1}.*

Different learning algorithms may use different performance criteria, and thus may have different formats and expressions of refinement sufficient statistics.

By combining the three elements of AVT-based classifier learners, we can write the following procedure to show this general learning framework.

---

1. Identify estimated sufficient statistics $\mathcal{S}_L(D)$ for AVTs as counts $\{CCFC(\mathcal{T}_i) | i = 1, ..., N\}$ or $\{FC(\mathcal{T}_i) | i = 1, ..., N\}$.
2. Initialize the global cut Γ to the most abstract cut Γ_0.
3. Based on the estimated sufficient statistic, generate a hypothesis $h(\Gamma)$ corresponding to the current global cut Γ and learn its parameters.
4. Generate a cut refinement $\hat{\Gamma}$ on Γ, and construct hypothesis $h(\hat{\Gamma})$
5. Calculate $\mathcal{S}_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma}))$ for hypothesis refinement from $h(\Gamma)$ to $h(\hat{\Gamma})$.
6. Based on performance criterion P, if stopping criterion is met, then output $h(\Gamma)$ as the final classifier; else if the condition for hypothesis refinement is met, set current hypothesis to $h(\hat{\Gamma})$ by replacing Γ with $\hat{\Gamma}$, else keep $h(\Gamma)$, and goto step 4;

Next, we discuss two instantiations of this learning framework and identify their corresponding elements within the same framework.

3.2 AVT-Based Naïve Bayes Learner (AVT-NBL)

AVT-NBL [28] is an extension of the standard Naïve Bayes learning algorithm that effectively exploits user-supplied AVTs to construct compact and accurate Naïve Bayes classifier from partially specified data. We can easily identify the three elements in the learning framework for AVT-NBL as follows:

(1) The sufficient statistics $\mathcal{S}_L(D)$ for AVT-NBL is the class conditional frequency counts $\{CCFC(\mathcal{T}_i) | i = 1, ..., N\}$.

(2) The hypothesis refinements strictly follow the procedure of cut refinements in the framework. When a global cut Γ is specified, there is a corresponding Naïve Bayes classifier $h(\Gamma)$ that is completely specified by a set of class conditional probabilities for the attribute values on Γ. Because each attribute is assumed to be independent of others given the class, the search for the AVT-based Naïve Bayes classifier (AVT-NBC) can be performed efficiently by optimizing the criterion independently for each attribute.

(3) The performance criterion that AVT-NBL optimizes is the Conditional Minimum Description Length (CMDL) score suggested by Friedman et al [12]. CMDL score can be calculated as follows:

$$CMDL(h(\Gamma)|D) = \left(\frac{\log|D|}{2}\right) size(h(\Gamma)) - CLL(h(\Gamma)|D)$$
$$where,\ CLL(h(\Gamma)|D) = |D|\sum_{p=1}^{|D|} \log P_h(c_{X_p}|v_{1p}, \cdots, v_{Np})$$

where, $P_h(c_{X_p}|v_{1p}, \cdots, v_{Np})$ is the class conditional probability, $size(h(\Gamma))$ is the number of parameters used by $h(\Gamma)$, $|D|$ the size of the data set, and $CLL(h(\Gamma)|D)$ is the conditional log likelihood of the hypothesis $h(\Gamma)$ given the data D. In the case of a Naïve Bayes classifier, $size(h(\Gamma))$ corresponds to the total number of class conditional probabilities needed to describe $h(\Gamma)$.

The sufficient statistics for hypothesis refinement in AVT-NBL can be quantified by the difference between their respective CMDL scores: $s_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma})) = CMDL(h(\hat{\Gamma})|D) - CMDL(h(\Gamma)|D)$. If $s_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma})) > 0$, $h(\Gamma)$ is refined to $h(\hat{\Gamma})$. This refinement procedure terminates when no further refinement can make improvement in the CMDL score (i.e., the stoping criterion).

3.3 AVT-Based Decision Tree Learner (AVT-DTL)

AVT-DTL [27] implements a top-down AVT-guided search in decision tree hypothesis space, and is able to learn compact and accurate decision tree classifier from partially specified data. Similarly, we can identify the three elements in the learning framework for AVT-DTL as follows:

(1) The sufficient statistics $\mathcal{S}_L(D)$ for AVT-DTL is the frequency counts $\{FC(\mathcal{T}_i)|i = 1, ..., N\}$.

(2) The hypothesis refinement is incorporated into the process of decision tree construction. The cut refinement is done by keeping track of "pointing vectors" in the AVTs. Each "pointing vector" is a set of pointers, and each pointer points to a values in an AVT. As an example, in Figure 3, the pointing vector points to two high-level attribute values in the two corresponding taxonomies.

The union of the set of pointing vectors at all leaves of a partially constructed decision tree corresponds to a global cut in AVTs. Obviously, any global cut in the constructed decision tree has a corresponding global cut in AVTs. At each stage of decision tree construction, we have a current set of pointing vectors as the global cut Γ being explored, and a corresponding partially constructed decision tree to be the hypothesis $h(\Gamma)$. AVT-DTL indirectly makes refinement on Γ by updating each pointing vector, and hence makes hypothesis refinement

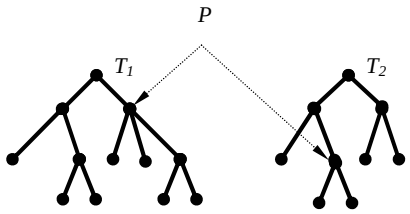


Fig. 3. Illustration of a Pointing Vector P

on $h(\Gamma)$ and grows the decision tree accordingly. AVT-DTL does not have the independent assumption on attributes given the class, the search is conducted globally to make refinements on possible cuts.

(3) The performance criterion that AVT-DTL uses is the standard information gain or gain ratio [20]. The sufficient statistic for hypothesis refinement is exactly the information criterion: $s_L(D, h(\Gamma) \rightarrow h(\hat{\Gamma})) = info(\Gamma \rightarrow \hat{\Gamma})$, where $info(\Gamma \rightarrow \hat{\Gamma})$ is the information gain (or gain ratio) when current decision tree $h(\Gamma)$ has been extended to $h(\hat{\Gamma})$. The stopping criterion for AVT-DTL is the same for standard decision tree. For example, such stopping criterion can be χ^2 test to test statistical significance on further split.

4 Experiments and results

We summarize below, results of experiments that compare the performance of standard learning algorithm (DTL, NBL) with that of their AVT-based counterparts (AVT-DTL/AVT-NBL) as well as the standard learning algorithms applied to a propositionalized version of the data set (PROP-DTL/PROP-NBL) [27]. In propositionalized method, the data set is represented using a set of Boolean attributes obtained from $\mathcal{T}_i$ of attribute A_i by associating a Boolean attribute with each node (except the root) in $\mathcal{T}_i$. Thus, each instance in the original data set defined using N attributes is turned into a Boolean instance specified using $\tilde{N}$ Boolean attributes where $\tilde{N} = \sum_{i=1}^{N}(|Nodes(\mathcal{T}_i)| - 1)$.

The data sets used in our experiments [27][28] were based on benchmark data sets available in the UC-Irvine repository. AVTs were supplied by domain experts for some of the data sets. For the remaining data sets, the AVTs were generated using AVT-Learner, a Hierarchical Agglomerative Clustering (HAC) algorithm for constructing AVTs [15].

Table 1 shows the estimated error rates of the Naïve Bayes classifiers generated by the AVT-NBL, NBL, and PROP-NBL on benchmark data sets [28]. Table 2 shows the estimated error rates of the decision tree classifiers generated by the AVT-DTL, C4.5 [20], and PROP-C4.5 on the same benchmark data sets.

Experiments were also run with synthetic data sets with different pre-specified percentages of totally or partially missing attribute values generated from the original benchmark data sets. Table 3 compares the estimated error rates of

AVT-NBL with that of NBL and PROP-NBL in the presence of varying percentages of partially missing attribute values and totally missing attribute values [28].

Table 1. Comparison of error rate and size of classifier generated by NBL, PROP-NBL and AVT-NBL on benchmark data

% Error rates using 10-fold cross validation with 90% confidence interval; The size of the classifiers for each data set is constant for NBL and Prop-NBL, and for AVT-NBL, the size shown represents the average across the 10-cross validation experiments.

DATA SET	NBL		PROP-NBL		AVT-NBL	
	ERROR	*SIZE*	ERROR	*SIZE*	ERROR	*SIZE*
Audiology	26.55 (±5.31)	*3696*	27.87 (±5.39)	*8184*	23.01 (±5.06)	*3600*
Breast-Cancer	28.32 (±4.82)	*84*	27.27 (±4.76)	*338*	27.62 (±4.78)	*62*
Car	14.47 (±1.53)	*88*	15.45 (±1.57)	*244*	13.83 (±1.50)	*80*
Dermatology	2.18 (±1.38)	*876*	1.91 (±1.29)	*2790*	2.18 (±1.38)	*576*
Mushroom	4.43 (±1.30)	*252*	4.45 (±1.30)	*682*	0.14 (±0.14)	*202*
Nursery	9.67 (±1.48)	*135*	10.59 (±1.54)	*355*	9.67 (±1.48)	*125*
Soybean	7.03 (±1.60)	*1900*	8.19 (±1.72)	*4959*	5.71 (±1.45)	*1729*
Zoo	6.93 (±4.57)	*259*	5.94 (±4.25)	*567*	3.96 (±3.51)	*245*

Table 2. Comparison of error rate and size of classifier generated by C4.5, PROP-C4.5 and AVT-DTL on benchmark data. No pruning is applied.

% Error rates using 10-fold cross validation with 90% confidence interval; The size of the classifier for each data set represents the average across the 10-cross validation experiments.

DATA SET	C4.5		PROP- C4.5		AVT- DTL	
	ERROR	*SIZE*	ERROR	*SIZE*	ERROR	*SIZE*
Audiology	23.01 (±5.06)	*37*	23.01 (±5.06)	*26*	21.23 (±4.91)	*30*
Breast-Cancer	33.91 (±5.06)	*152*	32.86 (±5.03)	*58*	29.37 (±4.87)	*38*
Car	7.75 (±1.16)	*297*	1.79 (±0.58)	*78*	1.67 (±0.57)	*78*
Dermatology	6.83 (±2.38)	*71*	5.74 (±2.20)	*19*	5.73 (±2.19)	*22*
Mushroom	0.0 (±0.00)	*26*	0.0 (±0.00)	*10*	0.0 (±0.00)	*10*
Nursery	3.34 (±0.90)	*680*	1.75 (±0.66)	*196*	1.21 (±0.55)	*172*
Soybean	9.81 (±2.06)	*175*	8.20 (±1.90)	*67*	7.75 (±1.85)	*90*
Zoo	7.92 (±4.86)	*13*	8.91 (±5.13)	*9*	7.92 (±4.86)	*7*

Our main results can be summarized as follows: (1) AVT-DTL and AVT-NBL are able to learn robust high accuracy classifiers from data sets consisting of partially specified data comparing to those produced by their standard counterparts on original data and propositionalized data. (2) Both AVT-DTL and AVT-NBL yield substantially more compact and comprehensible classifiers than standard version and propositionalized version of standard classifiers.

Table 3. Comparison of error rates on data with 10%, 30% and 50% partially or totally missing values. The error rates were estimated using 10-fold cross validation, and we calculate 90% confidence interval on each error rate.

DATA		PARTIALLY MISSING			TOTALLY MISSING		
METHODS		NBL	PROP-NBL	AVT-NBL	NBL	PROP-NBL	AVT-NBL
MUSHROOM	10%	4.65(±1.33)	4.69(±1.34)	0.30(±0.30)	4.65(±1.33)	4.76(±1.35)	1.29(±071)
	30%	5.28 (±1.41)	4.84(±1.36)	0.64(±0.50)	5.28 (±1.41)	5.37(±1.43)	2.78(±1.04)
	50%	6.63(±1.57)	5.82(±1.48)	1.24(±0.70)	6.63(±1.57)	6.98(±1.61)	4.61(±1.33)
NURSERY	10%	15.27(±1.81)	15.50(±1.82)	12.85(±1.67)	15.27(±1.81)	16.53(±1.86)	13.24(±1.70)
	30%	26.84(±2.23)	26.25(±2.21)	21.19(±2.05)	26.84(±2.23)	27.65(±2.24)	22.48(±2.09)
	50%	36.96(±2.43)	35.88(±2.41)	29.34(±2.29)	36.96(±2.43)	38.66(±2.45)	32.51(±2.35)
SOYBEAN	10%	8.76(±1.76)	9.08(±1.79)	6.75(±1.57)	8.76(±1.76)	9.09(±1.79)	6.88(±1.58)
	30%	12.45(±2.07)	11.54(±2.00)	10.32(±1.90)	12.45(±2.07)	12.31(±2.05)	10.41(±1.91)
	50%	19.39(±2.47)	16.91(±2.34)	16.93(±2.34)	19.39 (±2.47)	19.59(±2.48)	17.97(±2.40)

5 Summary and Discussion

5.1 Summary

Ontology-aware classifier learning algorithms are needed to explore data from multiple points of view, and to understand the interaction between data and knowledge. By exploiting ontologies in the form of attribute value taxonomies in learning classifiers from data, we are able to construct robust, accurate and easy-to-comprehend classifiers within a particular domain of interest.

We provide a general framework for learning classifiers from attribute value taxonomies and data. We illustrate the application of this framework in the case of AVT-based variants of decision tree and Naïve Bayes classifiers. However, this framework can be used to derive AVT-based variants of other learning algorithms, such as nonlinear regression classifiers, support vector machines, etc.

5.2 Related Work

Several authors have explored the use of attribute value taxonomies in learning classifiers from data. [1][9][10][13][17][23][27][28]. The use of prior knowledge or domain theories specified in first order logic or propositional logic to guide learning from data has been explored in ML-SMART [4], FOCL [19], and KBANN [24] systems. However, the work on exploiting domain theories in learning has not focused on the effective use of AVT to learn classifiers from partially specified data. McClean et al [18] proposed aggregation operators defined over partial values in databases. Caragea et al have explained the use of ontologies in learning classifiers from semantically heterogeneous data [8]. The use of multiple independent sets of features has led to "multi-view" learning [6]. However, our work focuses on exploring data with associated AVTs at multiple levels of abstraction, which corresponds to multiple points of view of the user.

In this paper, we have described a general framework for deriving ontology-aware algorithms for learning classifiers from data when ontologies take the form of attribute value taxonomies.

5.3 Future Work

Some promising directions for future work in ontology-guided data-driven learning include:

(1) Design of AVT-based variants of other machine learning algorithms. Specifically, it would be interesting to design AVT and CT-based variants of algorithms for construction Bag-of-words classifiers, Bayesian Networks, Nonlinear Regression Classifiers, and Hyperplane classifiers (Perceptron, Winnow Perceptron, and Support Vector Machines).
(2) Extensions that incorporate richer classes of AVT. Our work has so far focused on tree-structured taxonomies defined over nominal attribute values. It would be interesting to extend this work in several directions motivated by the natural characteristics of data: (a) Hierarchies of Intervals to handle numerical attribute values; (b) Ordered generalization Hierarchies where there is an ordering relation among nodes at a given level of a hierarchy (e.g., hierarchies over education levels); (c) Tangled Hierarchies that are represented by directed acyclic graphs (DAG) and Incomplete Hierarchies which can be represented by a forest of trees or DAGs.

Acknowledgments

This research was supported in part by grants from the National Science Foundation (NSF IIS 0219699) and the National Institutes of Health (GM 066387).

References

1. Almuallim H., Akiba, Y., Kaneda, S.: On Handling Tree-Structured Attributes. Proceedings of the Twelfth International Conference on Machine Learning (1995)
2. Akaike, H.: A New Look at Statistical Model Identification. IEEE Trans. on Automatic Control, AU-19:716-722. (1974)
3. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. 25(1). (2000)
4. Bergadano, F., Giordana, A.: Guiding Induction with Domain Theories. In: Machine Learning - An Artificial Intelligence Approach. Vol. 3, pp 474-492, Morgan Kaufmann. (1990)
5. Berners-Lee, T., Hendler, J. and Lassila, O.: The semantic web. Scientific American, May. (2001)
6. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. Annual Conference on Computational Learning Theory. (COLT-1998)
7. Caragea, D., Silvescu, A., and Honavar, V.: A Framework for Learning from Distributed Data Using Sufficient Statistics and its Application to Learning Decision Trees. International Journal of Hybrid Intelligent Systems. Vol. 1 (2004)
8. Caragea, D., Pathak, J., and Honavar, V.: Learning Classifiers from Semantically Heterogeneous Data. In 3rd International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. (2004)

9. A. Clare, R. King.: Knowledge Discovery in Multi-label Phenotype Data. In: Lecture Notes in Computer Science. Vol. 2168. (2001)
10. W. Cohen.: Learning Trees and Rules with Set-valued Features. In. Proceedings of the Thirteenth National Conference on Artificial Intelligence. AAAI Press, (1996)
11. Dempster A., Laird N., Rubin D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B, 39(1), pp 1-38. (1977)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning, Vol: 29. (1997)
13. 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)
14. Haussler, D.: Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. Artificial Intelligence, 36. (1988)
15. D. Kang, A. Silvescu, J. Zhang, and V. Honavar. Generation of Attribute Value Taxonomies from Data for Data-Driven Construction of Accurate and Compact Classifiers. To appear: Proceedings of The Fourth IEEE International Conference on Data Mining, 2004.
16. Kohavi, R., Provost, P.: Applications of Data Mining to Electronic Commerce. Data Mining and Knowledge Discovery, Vol. 5. (2001)
17. D. Koller, M. Sahami.: Hierarchically classifying documents using very few words. In: Proceedings of the 14th Int'l Conference on Machine Learning. (1997)
18. McClean S., Scotney B., Shapcott M.:Aggregation of Imprecise and Uncertain Information in Databases. IEEE Trans. on Knowledge and Data Engineering Vol. 13(6), pp 902-912. (2001)
19. Pazzani M., Kibler D.: The role of prior knowledge in inductive learning. Machine Learning, Vol. 9, pp 54-97. (1992)
20. Quinlan, J. R.: C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann (1992)
21. Rissanen, J.: Modeling by shortest data description. Automatica, vol. 14. (1978)
22. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. New York: PWS Publishing. (1999)
23. 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)
24. Towell, G., Shavlik, J.: Knowledge-based Artificial Neural Networks. Artificial Intelligence, Vol. 70. (1994)
25. Undercoffer, J., et al.: A Target Centric Ontology for Intrusion Detection: Using DAML+OIL to Classify Intrusive Behaviors. To appear, Knowledge Engineering Review - Special Issue on Ontologies for Distributed Systems, Cambridge University Press. (2004)
26. Zhang, J., Silvescu, A., Honavar, V.: Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction. Proceedings of Symposium on Abstraction, Reformulation, and Approximation. Lecture Notes in Computer Science 2371. (2002)
27. Zhang, J., Honavar, V.: Learning Decision Tree Classifiers from Attribute Value Taxonomies and Partially Specified Instances. In: Proceedings of the 20th Int'l Conference on Machine Learning. (2003)
28. Zhang, J., Honavar, V.: AVT-NBL: An Algorithm for Learning Compact and Accurate Naïve Bayes Classifiers from Attribute Value Taxonomies and Data. In: Proceedings of the Fourth IEEE International Conference on Data Mining. (2004)