Learning Naïve Bayes Classifiers From Attribute Value Taxonomies and Partially Specified Data

Jun Zhang
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1040, USA
Email: jzhang@cs.iastate.edu

Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1040, USA
Email: honavar@cs.iastate.edu

Abstract— **Partially specified data are commonplace in many practical applications of machine learning where different instances are described at different levels of precision relative to an attribute value taxonomy (AVT). This paper describes AVT-NBL - an extension of the Naïve Bayes Learning algorithm that effectively exploits user-supplied attribute value taxonomies to construct compact and accurate Naïve Bayes classifiers from partially specified data. Our experiments with benchmark data sets and AVTs show that AVT-NBL yields classifiers that are substantially more accurate and more compact than those obtained using the standard Naïve Bayes learner.**

I. Introduction

Current data-driven scientific discovery processes commonly call for the explorations of data from multiple points of view. For many pattern classification tasks, it is often the case that the instances to be classified are specified at different levels of precision. That is, the value of a particular attribute, or the class label associated with an instance, or both are specified at different levels of precision in different instances, leading to *partially specified instances* [21]. Algorithms for learning from AVT and partially specified data are of significant practical interest for several reasons:

a. Partially specified data are quite common in many application domains including medical diagnosis, scientific discovery, electronic commerce, and security informatics. For example, in a medical diagnosis task, different cases may be described in terms of symptoms or results of diagnostic tests at different levels of precision e.g., a patient may be described as having cardiac arrhythmia without specifying the precise type of arrhythmia.
b. Partially specified data are unavoidable in knowledge acquisition scenarios which call for integration of information from semantically heterogeneous information sources [17]. Semantic differences between information sources arise as a direct consequence of differences in ontological commitments [2]. Increasing need for data sharing between autonomous organizations and groups have led to major efforts aimed at construction of taxonomies (e.g., AVT). Examples include gene ontology (www.geneontology.org) [1], and ontology for intrusion detection [19].
c. An important goal of machine learning is to discover comprehensible, yet accurate and robust classifiers [16]. The availability of AVT presents the opportunity to learn classification rules that are expressed in terms of *abstract* attribute values leading to simpler, easier-to-comprehend rules that are expressed in terms of familiar hierarchically related concepts [20]. Kohavi and Provost [9] have noted the need to be able to incorporate background knowledge in the form of hierarchies over data attributes in e-commerce applications of data mining.
d. When training data are limited, there is a risk of generating classifiers that over fit the training data. A common approach used by statisticians when estimating from small samples involves *shrinkage* [13] to estimate the relevant statistics with adequate confidence. Learning algorithms that exploit AVT can potentially perform *shrinkage* automatically thereby yielding robust classifiers.

Against this background, this paper introduces AVT-NBL, an AVT-based generalization of the standard algorithm for learning Naïve Bayes classifiers from partially specified data. The rest of the paper is organized as follows: Section 2 formalizes the notion of AVT taxonomies, partially specified data, and learning classifiers from AVT and partially specified data; Section 3 discuss briefly on approaches to learning from partially specified data; Section 4 presents the AVT-NBL algorithm; Section 5 describes our experimental results and Section 6 concludes with summary and discussion.

II. Preliminaries

In what follows, we formally define AVT, introduce the notions of a partially missing value and partially specified instances, and formalize the problem of learning from AVT and partially specified data.

A. Attribute Value Taxonomies

Let $\mathbf{A} = \{A_1, A_2, ..., A_N\}$, be an ordered set of attributes and $\mathbf{C} = \{c_1, c_2, ..., c_M\}$ a finite set of mutually disjoint classes. Let $Values(A_i)$ denote the set of values (the domain) of attribute A_i. Instances are represented using ordered tuples of attribute values. Each instance belongs to a class in $\mathbf{C}$.

Let T_i be an Attribute Value Taxonomy $AVT(A_i)$ defined over the possible values of attribute A_i. We use T_i and $AVT(A_i)$ interchangeably to represent AVT for attribute A_i. Let $Nodes(T_i)$ represent the set of all values in T_i, and $Root(T_i)$ stand for the root of T_i. The set of leaves of the tree, $Leaves(T_i) = Values(A_i)$, corresponds to the set of *primitive values* of attribute A_i. The internal nodes of the tree correspond to *abstract values* of attribute A_i. For example, Figure 1 shows two attributes with corresponding AVTs for describing students in terms of their *student status* and *work status*.

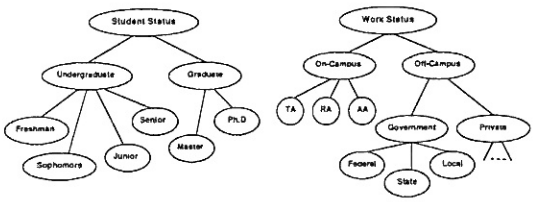

Fig. 1. Illustrative taxonomies on student status and work status

We define two operations on AVT T_i associated with an attribute A_i.

- $depth(T_i, v(A_i))$ returns the length of the path from root to an attribute value $v(A_i)$ in the taxonomy;
- $leaf(T_i, v(A_i))$ return a Boolean value indicating if $v(A_i)$ is a leaf node in T_i, that is if $v(A_i) \in Leaves(T_i)$.

After Haussler [8], we define a cut γ_i for $AVT(A_i)$.

Definition 1 (Cut): A *Cut* γ_i is a subset of elements in $AVT(A_i)$ satisfying the following two properties: (1) For any leaf $l \in Leaves(T_i)$, either $l \in \gamma_i$ or l 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.

A cut γ_i induces a partition of elements of $Values(A_i)$. For example in Figure 1, $\{$*On-Campus*, *Government*, *Private*$\}$ defines a partition over the primitive values of the *work status* attribute.

Let $\mathbf{T} = \{T_1, T_2, ..., T_N\}$ denote the ordered set of AVTs associated with $A_1 \cdots A_N$. Let $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_N\}$ denote a set of cuts through $T_1 \cdots T_N$ accordingly.

Let $\psi(v, T_i)$ be the set of descendants of a node corresponding to value v in the AVT T_i; $\pi(v, T_i)$, the set of all children (direct descendants) of a node with value v in T_i; $\Lambda(v, T_i)$ the list of ancestors, including the root, for v in T_i.

Let $\mathbf{\Gamma_A} = \times_i \Gamma_{A_i}$ denote the cartesian product of the cuts through the individual AVTs.

Definition 2 (Refinements): 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 descendants $\psi(v, T_i)$. Conversely, γ_i is an abstraction of $\hat{\gamma}_i$. We say that a set of cuts $\hat{\mathbf{\Gamma}}_\mathbf{A}$ is a refinement of a set of cuts $\mathbf{\Gamma_A}$ if at least one cut in $\hat{\mathbf{\Gamma}}_\mathbf{A}$ is a refinement of a cut in $\mathbf{\Gamma_A}$. Conversely, the set of cuts $\mathbf{\Gamma_A}$ is an abstraction of the set of cuts $\hat{\mathbf{\Gamma}}_\mathbf{A}$.

Figure 2 illustrates a refinement process. The cut $\gamma_2^{(1)} = \{A, B, C, D\}$ in T_2 has been refined to $\gamma_2^{(2)} = \{A, B_1, B_2, C, D\}$ by replacing B with its two children B_1, B_2. Therefore, $\mathbf{\Gamma}^{(2)} = \{\gamma_1, \gamma_2^{(2)}, \gamma_3\}$ is a refinement of $\mathbf{\Gamma}^{(1)} = \{\gamma_1, \gamma_2^{(1)}, \gamma_3\}$.

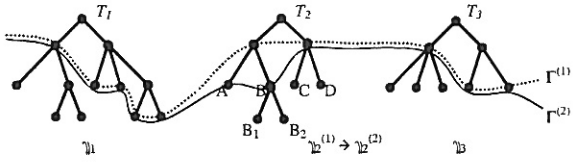

Fig. 2. A demonstrative refinement process

Definition 3 (Instance Space): Any choice of $\mathbf{\Gamma_A} = \times_i \Gamma_{A_i}$ defines an *instance space* $\mathbf{I}_{\mathbf{\Gamma}_{\mathbf{A}_i}}$. When $\exists \Gamma_{A_i}$ of $\mathbf{\Gamma_A}$ such that $\Gamma_{A_i} \neq Leaves(A_i)$, the resulting instance space is an abstraction of the original instance space $\mathbf{I}$. The original instance space is given by $\mathbf{I} = \times_i Values(A_i) = \times_i Leaves(A_i)$, that is, the cartesian product of the primitive values of the attributes $A_1 \cdots A_N$.

B. Partially Specified Data

Definition 4 (Partially Specified Data): An instance X_j is represented by a tuple $= (v_{1j}, v_{2j}, ..., v_{Nj})$. X_j is:

- a completely specified instance if $\forall i\, v_{ij} \in Leaves(T_i)$
- a partially specified instance if one or more of its attribute values are not primitive: $\exists v_{ij} \in X_j$, $depth(T_i, v_{ij}) \geq 0 \wedge \neg leaf(T_i, v_{ij})$

Thus, a partially specified instance is an instance in which at least one of the attributes is partially specified. Relative to the AVT shown in Figure 1, the instance (*Senior*, *TA*) is a fully specified instance. Some examples of partially specified instances are: (*Undergraduate*, *RA*), (*Freshman*, *Government*), (*Graduate*, *Off-Campus*).

Definition 5 (Instance Space Induced by a Set of AVTs): A set of AVTs $\mathbf{T} = \{T_1 \cdots T_N\}$ associated with a set of attributes $\mathbf{A} = \{A_1 \cdots A_N\}$ induces an instance space $\mathbf{I_A} = \cup_\Gamma \mathbf{I}_\Gamma$ (the union of instance spaces induced by all of the the cuts through the set of AVTs $\mathbf{T}$).

Definition 6 (A Partially Specified Data Set): A partially specified data set $\mathbf{D_T}$ (relative to a set $\mathbf{T}$ of attribute value taxonomies) is a collection of instances drawn from $\mathbf{I_A}$ where each instance is labelled with the appropriate class label from $\mathbf{C}$. Thus, $\mathbf{D_T} \subseteq \mathbf{I_A} \times \mathbf{C}$.

C. Learning Classifier from AVT and Partially Specified Data

The problem of learning classifiers from AVT and partially specified data is a natural generalization of the problem of learning classifiers from (fully specified) data. A classifier is a hypothesis in the form of a function $h : \mathbf{I} \rightarrow \mathbf{C}$, whose domain is the instance space $\mathbf{I}$ and whose range is the set of classes $\mathbf{C}$. A hypothesis space $\mathbf{H}$ is a set of hypotheses that can be represented in some hypothesis language or by a parameterized family of functions (e.g., Naive Bayes classifiers, SVM, etc.). The task of learning classifiers from a fully specified data set entails identifying a hypothesis $h \in \mathbf{H}$ that satisfies some criteria (e.g., a hypothesis that is most likely given the training data).

The problem of learning classifiers from partially specified data can be stated as follows: Given a user-supplied set of

AVTs $\mathbf{T}$ and a data set $\mathbf{D_T}$ of (possibly) partially specified labelled instances, construct a classifier $h_T : \mathbf{I_A} \rightarrow \mathbf{C}$ for assigning appropriate class labels to each instance in the instance space $\mathbf{I_A}$. Of special interest is the case in which the resulting hypothesis space $\mathbf{H_T}$ has structure that makes it possible to search it efficiently for a hypothesis that is both concise as well as accurate. This is true in the case of learning Naive Bayes classifiers from AVT and partially specified data.

III. Approaches to Learning Classifiers from Partially Specified Data

We can envision three approaches to learning from Partially Specified Data:

(1) Approaches that Treat Partially Specified Attribute Values as if they were Totally Missing: Each partially specified (and hence partially missing) attribute value is treated as if it were (totally) missing, and the resulting data set with missing attribute values is handled using standard approaches for dealing with missing attribute values in learning classifiers. A main advantage of this approach is that it requires no modification to the learning algorithm.

(2) AVT-Based Propositionalization Methods: The data set is represented using a set of Boolean attributes obtained from $Nodes(T_i)$ of attribute A_i by associating a Boolean attribute with each node (except the root) in 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 $\tilde{N} = \sum_{i=1}^{N} |Nodes(A_i)|$. The Boolean attributes that correspond to descendants of the partially specified attribute value are treated as unknown.

Note that the Boolean features created by the propositionalization technique described above are not independent given the class. A Boolean attribute that corresponds to any node in an AVT is necessarily correlated with Boolean attributes that correspond to its descendants as well as its ancestors in the tree. Thus, a Naïve Bayes classifier that would be optimal in the Maximal a Posteriori sense [11] would no longer be optimal because of the strong dependencies among the Boolean attributes derived from an AVT.

(3) AVT Guided Variants of Standard Learning algorithms: We can extend standard learning algorithms in principled ways so as to exploit the information provided by AVT [20]. AVT-DTL [21] which extends the standard decision tree learning algorithm and the AVT-NBL algorithm described in this paper which extends the standard algorithm for learning Naïve Bayes classifiers are examples of this class of algorithms.

IV. AVT-Based Naïve Bayes Learner (AVT-NBL)

A. Naïve Bayes Learner (NBL)

Suppose each attribute A_i takes a value from a finite set of values $V(A_i)$. An instance X_p to be classified is represented as a tuple of attribute values $(v_{1p}, v_{2p}, \cdots, v_{Np})$ where each $a_{ip} \in V(A_i)$. The Bayesian approach to classifying X_p is to assign it the most probable class $c_{MAP}(X_p)$. Naïve Bayes classifier operates under the assumption that each attribute is independent of others given the class. Hence, we have:

$$\begin{aligned} c_{MAP}(X_p) &= \underset{c_j \in C}{\operatorname{argmax}}\, P(v_{1p}, v_{2p}, \cdots, v_{Np}|c_j)p(c_j) \\ &= \underset{c_j \in C}{\operatorname{argmax}}\, p(c_j) \prod_i P(v_{ip}|c_j) \end{aligned}$$

The standard algorithm (NBL) for learning a Naïve Bayes classifier simply estimates a class conditional probability table for each attribute from a data set D of training examples. The class conditional probability table for attribute A_i has $|V(A_i)||C|$ entries.

B. AVT-Guided Naïve Bayes Learner (AVT-NBL)

Given a user-supplied ordered set of AVTs $\mathbf{T} = \{T_1, \cdots, T_N\}$ corresponding to the attributes $A_1 \cdots A_N$ and a data set $D = \{(X_p, c_p)\}$ of labelled examples of the form (X_p, c_p) where $X_p \in \mathbf{I_A}$ is a partially or fully specified instance and $c_p \in \mathbf{C}$ is the corresponding class label, the task of AVT-NBL is to construct a Naïve Bayes classifier for assigning a partially specified instance X_p to its most probable class $c_{MAP}(X_p)$. As in the case of NBL, we assume that each attribute is independent of other attributes given the class.

Let $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_N\}$ be a set of cuts where, γ_i stands for a cut through T_i. A Naive Bayes classifier defined on the instance space $\mathbf{I}_\Gamma$ is completely specified by a set of class conditional probabilities for each value of each attribute. Suppose we denote the table of class conditional probabilities associated with values in γ_i by $CPT(\gamma_i)$. Then the Naive Bayes classifier defined over the instance space $\mathbf{I}_\Gamma$ is specified by $h(\Gamma) = \{CPT(\gamma_1), CPT(\gamma_2), \cdots, CPT(\gamma_N)\}$.

We start with the Naïve Bayes Classifier that is based on the most abstract value of each attribute (the most general hypothesis in $\mathbf{H_T}$) and successively refine the classifier (hypothesis) using a criterion that is designed to tradeoff between the accuracy of classification and the complexity of the resulting Naïve Bayes classifier.

1) Calculating Class Conditional Frequency Counts: Let $\sigma_i(v|c_j)$ be the frequency count of value v of attribute A_i given class label c_j in a training set D and $p_i(v|c_j)$, the estimated class conditional probability of value v of attribute A_i given class label c_j in a training set D.

Given an attribute value taxonomy T_i for attribute A_i, we can define a tree of class conditional frequency counts $CCFC(A_i)$ such that there is an one-to-one correspondence between the nodes of the AVT T_i and the nodes of the corresponding $CCFC(A_i)$. It follows that the class conditional frequency counts associated with a non leaf node of $CCFC(A_i)$ should correspond the aggregation of the corresponding class conditional frequency counts associated with its children. Because each cut through an AVT T_i corresponds to a partition of the set of possible values $Nodes(A_i)$ of the attribute A_i, the corresponding cut through $CCFC(A_i)$ specifies a valid class conditional probability table for the attribute A_i.

When all of the instances in the data set D are fully specified, estimation of $CCFC(A_i)$ for each attribute is straightforward: We simply estimate the class conditional frequency counts associated with each of the primitive values of A_i from the data set D and use them recursively to compute the class conditional frequency counts associated with the non-leaf nodes of $CCFC(A_i)$. When some of the data are partially specified, we can use a 2-step process for computing $CCFC(A_i)$: 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. The procedure is shown below.

1. Calculate frequency counts $\sigma_i(v|c_j)$ for each node v in T_i using the class conditional frequency counts associated with the specified values of attribute A_i in training set D.
2. For each attribute value v in T_i which received non-zero counts as a result of step 1, aggregate the counts upward from each such node v to its ancestors $\Lambda(v, T_i)$: $\sigma_i(w|c_j)_{w\in\Lambda(v,T_i)} \leftarrow \sigma_i(w|c_j) + \sigma_i(v|c_j)$
3. Starting from the root, recursively propagate the counts corresponding to partially specified instances at each node v downward according to the observed distribution among its children to obtain updated counts for each child $u_l \in Children(v, T_i)$:

$$\sigma_i(u_l|c_j) \leftarrow \sigma_i(u_l|c_j)\left(1+\frac{\sigma_i(v|c_j)-\sum_{k=1}^{|\pi(v,T_i)|}\sigma_i(u_k|c_j)}{\sum_{k=1}^{|\pi(v,T_i)|}\sigma_i(u_k|c_j)}\right)$$
$$\text{If } \sum_{k=1}^{|\pi(v,T_i)|}\sigma_i(u_k|c_j)\neq 0;$$
$$\sigma_i(u_l|c_j) \leftarrow \left(\frac{\sigma_i(v|c_j)}{|\pi(v,T_i)|}\right) \text{ Otherwise.}$$

Let $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_N\}$ be a set of cuts where, γ_i stands for a cut through $CCFC(A_i)$. The estimated conditional probability table $CPT(\gamma_i)$ associated with the cut γ_i can be calculated from $CCFC(A_i)$ using Laplacian estimator.

$$p_i(v|c_j)_{v\in\gamma_i} \leftarrow \frac{1+\sigma_i(v|c_j)}{|\gamma_i| + \sum_{u\in\gamma_i}\sigma_i(u|c_j)}$$

Recall that the Naïve Bayes Classifier $h(\Gamma)$ is completely specified by: $h(\Gamma) = \{CPT(\gamma_1), CPT(\gamma_2), \cdots, CPT(\gamma_N)\}$.

2) Searching for a Compact Naïve Bayes Classifier: The scoring function that we use to evaluate a candidate AVT-guided refinement of a Naïve Bayes Classifier is based on a variant of the minimum description length (MDL) score [18] which captures the tradeoff between the complexity and accuracy of the model. Friedman et al [7] suggested the use of a conditional MDL (CMDL) score in the case of hypotheses that are used for classification (as opposed to modelling the joint probability distribution of a set of random variables) to capture this tradeoff. In general, computation of CMDL score is not feasible for Bayesian networks with arbitrary structure. However, in the case of Naive Bayes classifiers induced by a set of AVT, as shown below, it is possible to efficiently calculate the CMDL score.

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

Here, $P_h(c_p|v_{1p}, \cdots, v_{Np})$ denotes the conditional probability assigned to the class $c_p \in C$ associated with the training sample $X_p = (v_{1p}, v_{2p}, \cdots, v_{Np})$ by the classifier h, $size(h)$ is the number of parameters used by h, $|D|$ the size of the data set, and $CLL(h|D)$ is the conditional log likelihood of the data D given a hypothesis h. In the case of a Naïve Bayes classifier h, $size(h)$ corresponds to the total number of class conditional probabilities needed to describe h. Because each attribute is assumed to be independent of the others given the class in a Naïve Bayes classifier, we have:

$$CLL(h|D) = |D|\sum_{p=1}^{|D|}\log\left(\frac{P(c_p)\prod_i P_h(v_{ip}|c_p)}{\sum_{j=1}^{|C|}P(c_j)\prod_i P_h(v_{ip}|c_j)}\right)$$

where $P(c_p)$ is the prior probability of the class c_p which can be estimated from the observed class distribution in data D.

There are two cases in the calculation of the conditional likelihood $CLL(h|D)$ when D contains partially specified instances. When a partially specified value of attribute A_i for an instance lies on the cut γ through $CCFC(A_i)$ or corresponds to one of the descendants of the nodes in the cut, we can treat that instance as though it were fully specified relative to the Naïve Bayes classifier based on the cut and use the class conditional probabilities associated with the cut γ to calculate its contribution to $CLL(h|D)$. The second case is when a partially specified value (say v) of A_i is an ancestor of a subset (say λ) of the nodes in γ. In this case, $p(v|c_j) = \sum_{u_i\in\lambda}p(u_i|c_j)$, such that we can aggregate the class conditional probabilities of the nodes in λ to calculate the contribution of the corresponding instance to $CLL(h|D)$.

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. This results in a hypothesis h that intuitively trades off the complexity of Naïve Bayes classifier (in terms of the number of parameters used to describe the relevant class conditional probabilities) against accuracy of classification. The algorithm terminates when none of the candidate refinements of the classifier yield statistically significant improvement in the CMDL score. The procedure is outlined below.

1. Initialize each γ_i in $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_N\}$ to $\{Root(T_i)\}$.
2. Estimate probabilities that specify the hypothesis $h(\Gamma)$.
3. For each cut γ_i in $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_N\}$:
   A. Set $\delta_i \leftarrow \gamma_i$
   B. Until there are no updates to γ_i
      i. For each $v \in \delta_i$
         a. Generate a refinement γ_i^v of γ_i by replacing v with $\pi(v, T_i)$, and refine Γ accordingly to obtain Δ. Construct corresponding hypothesis $h(\Delta)$
         b. If $CMDL(h(\Delta)|D) < CMDL(h(\Gamma)|D)$, replace Γ with Δ and γ_i with γ_i^v
      ii. $\delta_i \leftarrow \gamma_i$
4. Output $h(\Gamma)$

TABLE I

COMPARISON OF ERROR RATES OF NBL, PROP-NBL AND AVT-NBL ON BENCHMARK DATA SETS.

% Error rates using 10-fold cross validation with 90% confidence interval							
DATA		PARTIALLY MISSING			TOTALLY MISSING		
METHODS		NBL	PROP-NBL	AVT-NBL	NBL	PROP-NBL	AVT-NBL
MUSHROOM	0%	4.43(±1.30)	4.45(±1.30)	1.36(±0.73)	–	–	–
	10%	4.65(±1.33)	4.69(±1.34)	1.46(±0.76)	4.65(±1.33)	4.76(±1.35)	2.21(±0.93)
	30%	5.28 (±1.41)	4.84(±1.36)	1.57(±0.79)	5.28 (±1.41)	5.37(±1.43)	3.06(±1.09)
	50%	6.63(±1.57)	5.82(±1.48)	2.06(±0.90)	6.63(±1.57)	6.98(±1.61)	4.24(±1.27)
NURSERY	0%	9.67(±1.48)	10.59(±1.54)	9.67(±1.48)	–	–	–
	10%	15.27(±1.81)	15.50(±1.82)	12.97(±1.68)	15.27(±1.81)	16.53(±1.86)	14.05(±1.74)
	30%	26.84(±2.23)	26.25(±2.21)	21.27(±2.06)	26.84(±2.23)	27.65(±2.24)	23.79(±2.14)
	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	0%	7.03(±1.60)	8.19(±1.72)	6.44(±1.53)	–	–	–
	10%	11.12(±1.97)	11.13(±1.97)	8.49(±1.74)	11.12(±1.97)	11.71(±2.01)	9.37(±1.82)
	30%	12.45(±2.07)	11.78(±2.02)	8.99(±1.79)	12.45(±2.07)	12.75(±2.09)	9.81(±1.86)
	50%	17.42(±2.37)	14.91(±2.23)	12.35(±2.06)	17.42(±2.37)	18.59(±2.43)	14.32(±2.19)

V. EXPERIMENTS AND RESULTS

A. Experiments

Our experiments were designed to explore the performance of AVT-NBL relative to that of the standard Naïve Bayes algorithm (NBL) and a Naïve Bayes Learner applied to a propositionalized version of the data set (PROP-NBL).

Although partially specified data and hierarchical AVT are common in many application domains, at present, there are few standard benchmark data sets of partially specified data and the associated AVT. Hence, we describe results of experiments with several data sets (*Mushroom*, *Soybean*, and *Nursery*) from the UC Irvine Repository. In the case of *Mushroom*, AVT was supplied by a botanist. For the rest data sets, the AVTs were specified based on our understanding of the domain.

The first set of experiments compares the performance of AVT-NBL, NBL, and PROP-NBL on the original (fully specified) data. The second set of experiments explores the performance of the three algorithms on data sets with different percentages of totally missing and partially missing attribute values. Data sets with a pre-specified percentage (10%, 30%, or 50%) of totally or partially missing attribute values were generated by assuming that the missing values are uniformly distributed on the nominal attributes [21]. In each case, the error rate and the size (as measured by the number of class conditional probabilities used to specify the learned classifier) were estimated using 10-fold cross-validation, and we calculate 90% confidence interval on the error rate.

B. Results

AVT-NBL yields significantly lower error rates than NBL and PROP-NBL on the original fully specified data. Table 1 shows the estimated error rates of the classifiers generated by the AVT-NBL, NBL, and PROP-NBL on three benchmark data sets. The error rate of AVT-NBL is substantially smaller than that of NBL and PROP-NBL, with the difference in error rates being most pronounced in the case of *Mushroom* data. It is worth noting that PROP-NBL (NBL applied to a transformed data set using Boolean features that correspond to nodes of the AVTs) generally produces classifiers that have substantially higher error rates than AVT-NBL. This can be explained by the fact that the Boolean features generated from an AVT are generally not independent given the class. This argues for the investigation of algorithms such as AVT-NBL based on principled ways of exploiting supplied AVT in generating classifiers.

AVT-NBL yields significantly lower error rates than NBL and PROP-NBL on partially specified data and data with totally missing value. Table 1 compares the estimated error rates of AVT-NBL with that of NBL and PROP-NBL in the presence of varying percentages (10%, 30% and 50%) of partially missing attribute values and totally missing attribute values. Naïve Bayes classifiers generated by AVT-NBL have substantially lower error rates than those generated by NBL and PROP-NBL, with the differences being more pronounced at higher percentages of partially (or totally) missing attribute values.

TABLE II

COMPARISON OF THE COMPLEXITY OF THE CLASSIFIERS AS MEASURED BY THE NUMBER OF CLASS CONDITIONAL PROBABILITIES

	NBL		PROP-NBL		AVT-NBL	
DATA SET	Percentage of Partially Missing Values					
	0%	50%	0%	50%	0%	50%
MUSHROOM	252	252	682	682	192	194
NURSERY	135	135	355	355	125	125
SOYBEAN	1900	1900	4959	4959	1653	1723

AVT-NBL yields classifiers that are substantially more compact than those generated by PROP-NBL and NBL. Table 2 compares the total number of class conditional probabilities needed to specify the classifiers produced by AVT-NBL, NBL, and PROP-NBL when 0% and 50% of the attribute values are partially specified. The results show that AVT-NBL is effective in exploiting the information supplied by the AVT to generate accurate yet compact classifiers. Thus, AVT-guided learning

algorithms offer an approach to compressing class conditional probability distributions that is different from the statistical independence-based factorization used in Bayesian Networks.

VI. Summary and Discussion

A. Summary

In this paper, we have presented AVT-NBL, an algorithm for learning Naïve Bayes Classifiers using attribute value taxonomies from partially specified data. AVT-NBL is a natural generalization of the standard algorithm (NBL) for learning Naïve Bayes Classifiers.

Experimental results presented in the paper show that:

(1) AVT-NBL is able to learn substantially more accurate Naïve Bayes classifiers than those produced by NBL and PROP-NBL from data sets with varying percentages of partially specified attribute values (including data sets with no partially specified attribute values).
(2) Classifiers generated by AVT-NBL are substantially more compact than those generated by NBL and PROP-NBL.

B. Related Work

There is some work in the machine learning community on the problem of learning classifiers from attribute value taxonomies (sometimes called tree-structured attributes) and fully specified data in the case of decision trees and rules (see [21] for a review) desJardins et al [6] suggested the use of Abstraction-Based Search (ABS) to learn Bayesian networks with compact structure. Zhang and Honavar [21] describe AVT-DTL, an efficient algorithm for learning decision tree classifiers from AVT and partially specified data. With the exception of AVT-DTL, to the best of our knowledge, there are no algorithms for learning classifiers from AVT and partially specified data.

There has been some work on the use of class taxonomy (CT) in the learning of classifiers in scenarios where class labels correspond to nodes in a predefined class hierarchy [5][10].

. Chen et al. [4] proposed database models to handle imprecision using partial values and associated probabilities where a partial value refers to a set of possible values for an attribute. McClean et al [14] proposed aggregation operators defined over partial values. While this work suggests ways to aggregate statistics so as to minimize information loss, it does not address the problem of learning from AVT and partially specified data.

C. Future Work

Some directions for future work include:

(1) Development AVT-based variants of other machine learning algorithms for construction of classifiers from partially specified data from distributed, semantically heterogeneous data sources [17][3].
(2) Extension of the algorithms like AVT-DTL and AVT-NBL to handle taxonomies defined over ordered and numeric attribute values.
(3) Further experimental evaluation of AVT-NBL, AVT-DTL, and related learning algorithms on a broad range of data sets in scientific knowledge discovery applications e.g., computational biology.

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] M. Ashburner, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. 25(1), 2000.
[2] T. Berners-Lee, J. Hendler and O. Lassila. The semantic web. Scientific American, May 2001.
[3] D. Caragea, A. Silvescu, and V. Honavar. 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.
[4] A. Chen, J. Chiu, and F. Tseng. Evaluating aggregate operations over imprecise data. IEEE Trans. On Knowledge and Data Engineering, 8, 1996.
[5] A. Clare, R. King. Knowledge Discovery in Multi-label Phenotype Data. In: Lecture Notes in Computer Science. Vol. 2168, 2001.
[6] M. desJardins, L. Getoor, D. Koller. Using Feature Hierarchies in Bayesian Network Learning. Lecture Notes in Artificial Intelligence 1864, 2000.
[7] N. Friedman, D. Geiger. Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning, Vol: 29, 1997.
[8] D. Haussler. Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. Artificial Intelligence, 36, 1988.
[9] R. Kohavi, P. Provost. Applications of Data Mining to Electronic Commerce. Data Mining and Knowledge Discovery, Vol. 5, 2001.
[10] D. Koller, M. Sahami. Hierarchically classifying documents using very few words. In: Proceedings of the 14th Int'l Conference on Machine Learning, 1997.
[11] P. Langley, W. Iba, K. Thompson. An analysis of Bayesian classifiers Proceedings of the Tenth National Conference on Artificial Intelligence, 1992.
[12] F. Martin, E. Plaza. SOID: A Simple Ontology for Intrusion Detection. Seventh International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence 2773, 2003.
[13] A. McCallum, R. Rosenfeld, T. Mitchell, A. Ng. Improving Text Classification by Shrinkage in a Hierarchy of Classes. Proceedings of the 15th Int'l Conference on Machine Learning, 1998.
[14] S. McClean, B. Scotney, M. Shapcott. Aggregation of Imprecise and Uncertain Information in Databases. IEEE Transactions on Knowledge and Data Engineering (6), 2001.
[15] T. Mitchell. Machine Learning. New York: Addison-Wesley, 1997.
[16] M. Pazzani, S. Mani, W. Shankle. Beyond concise and colorful: Learning Intelligible Rules. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, 1997.
[17] J. Reinoso-Castillo, A. Silvescu, D. Caragea, J. Pathak, and V. Honavar. A Federated Query Centric Approach to Information Extraction and Integration from Heterogeneous, Distributed, and Autonomous Data Sources. In: The 2003 IEEE International Conference on Information Reuse and Integration, 2003.
[18] J. Rissanen. Modeling by shortest data description. Automatica, vol. 14, 1978.
[19] J. Undercoffer, 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.
[20] J. Zhang, A. Silvescu, and V. Honavar. Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction. Proceedings of Symposium on Abstraction, Reformulation, and Approximation 2002. Lecture Notes in Artificial Intelligence 2371, 2002.
[21] J. Zhang, V. Honavar. Learning From Attribute Value Taxonomies and Partially Specified Instances. In: Proceedings of the 20th Int'l Conference on Machine Learning, 2003.