AVT-NBL: An Algorithm for Learning Compact and Accurate Naïve Bayes Classifiers from Attribute Value Taxonomies and Data

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

Abstract

In many application domains, there is a need for learning algorithms that can effectively exploit attribute value taxonomies (AVT) - hierarchical groupings of attribute values - to learn compact, comprehensible, and accurate classifiers from data - including data that are partially specified. *This paper describes AVT-NBL, a natural generalization of the Naïve Bayes learner (NBL), for learning classifiers from AVT and data. Our experimental results show that AVT-NBL is able to generate classifiers that are substantially more compact and more accurate than those produced by NBL on a broad range of data sets with different percentages of partially specified values. We also show that AVT-NBL is more efficient in its use of training data: AVT-NBL produces classifiers that outperform those produced by NBL using substantially fewer training examples.*

1. Introduction

Synthesis of accurate and compact pattern classifiers from data is one of the major applications of data mining. In a typical inductive learning scenario, instances to be classified are represented as ordered tuples of attribute values. However, attribute values can be grouped together to reflect assumed or actual similarities among the values in a domain of interest or in the context of a specific application. Such a hierarchical grouping of attribute values yields an attribute value taxonomy (AVT). Such AVT are quite common in biological sciences. For example, the Gene Ontology Consortium is developing hierarchical taxonomies for describing many aspects of macromolecular sequence, structure, and function [1]. Undercoffer et al. have developed a hierarchical taxonomy which captures the features that are observable or measurable by the target of an attack or by a system of sensors acting on behalf of the target [22]. Several ontologies being developed as part of the Semantic Web related efforts [2] also capture hierarchical groupings of attribute values. Kohavi and Provost 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 [11]. Against this background, algorithms for learning from AVT and data are of significant practical interest for several reasons:

a. An important goal of machine learning is to discover comprehensible, yet accurate and robust classifiers [18]. The availability of AVT presents the opportunity to learn classification rules that are expressed in terms of *abstract* attribute values leading to simpler, accurate and easier-to-comprehend rules that are expressed using familiar hierarchically related concepts [24] [11].

b. Exploiting AVT in learning classifier can potentially perform regularization to minimize overfitting when learning from relatively small data sets. A common approach used by statisticians when estimating from small samples involves *shrinkage* [15] to estimate the relevant statistics with adequate confidence. Learning algorithms that exploit AVT can potentially perform *shrinkage* automatically thereby yielding robust classifiers and minimizing over-fitting.

c. Presence of explicitly defined AVT allows specification of data at different levels of precision, giving rise to *partially specified instances* [25]. The attribute value of a particular attribute can be specified at different levels of precision in different instances. For example, the medical diagnostic test results given by different institutions are presented at different levels of precision. Partially specified data are unavoidable in knowledge acquisition scenarios which call for in-

tegration of information from semantically heterogeneous information sources [4]. Semantic differences between information sources arise as a direct consequence of differences in ontological commitments [2]. Hence, algorithms for learning classifiers from AVT and partially specified data are of great interest.

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 notions on learning classifiers with AVT taxonomies; Section 3 presents the AVT-NBL algorithm; Section 4 discusses briefly on alternative approaches; Section 5 describes our experimental results and Section 6 concludes with summary and discussion.

2 Preliminaries

In what follows, we formally define AVT, and its induced instance space. We introduce the notion of partially specified instances, and formalize the problem of learning from AVT and data.

2.1 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*.

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;

- $leaf(T_i, v(A_i))$ returns a Boolean value indicating if $v(A_i)$ is a leaf node in T_i, that is if $v(A_i) \in Leaves(T_i)$.

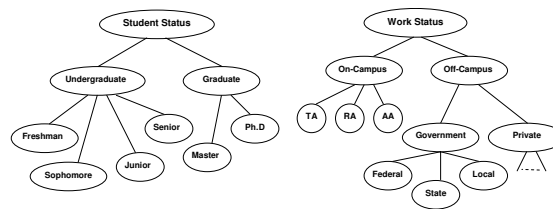After Haussler [9], we define a cut γ_i for $AVT(A_i)$ as follows.

Figure 1. Illustrative taxonomies on student status and work status

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 $A_1, A_2, \cdots A_N$. For each T_i, define Δ_i to be the set of all valid cuts in T_i. Let $\mathbf{\Delta} = \times_i \Delta_i$ denote the cartesian product of the cuts through the individual AVTs. Let $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_N\}$ be an ordered set that defines a global cut through $T_1, T_2, \cdots T_N$ accordingly, where $\gamma_i \in \Delta_i$ and $\Gamma \in \mathbf{\Delta}$.

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.

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 $\hat{\Gamma}$ is a refinement of a set of cuts Γ if at least one cut in $\hat{\Gamma}$ is a refinement of a cut in Γ. Conversely, the set of cuts Γ is an abstraction of the set of cuts $\hat{\Gamma}$.*
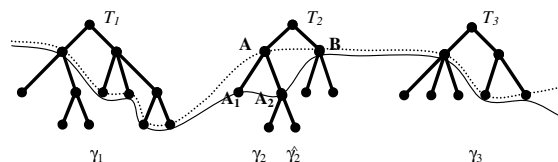

Figure 2. A demonstrative refinement process

Figure 2 illustrates a refinement process. $\gamma_2 = \{A, B\}$ in T_2 has been refined to $\hat{\gamma}_2 = \{A_1, A_2, B\}$ by replacing A with its two children A_1, A_2. Therefore, $\hat{\Gamma} = \{\gamma_1, \hat{\gamma}_2, \gamma_3\}$ is a refinement of $\Gamma = \{\gamma_1, \gamma_2, \gamma_3\}$.

2.2 AVT-Induced Instance Space

Definition 3 (Abstract Instance Space) *Any choice Γ of $\boldsymbol{\Delta} = \times_i \Delta_i$ defines an abstract instance space $\mathbf{I}_\Gamma$. When $\exists i \, \gamma_i \in \Gamma$ such that $\gamma_i \neq Leaves(T_i)$, the resulting instance space is an abstraction of the original instance space $\mathbf{I}$. The original instance space is given by $\mathbf{I} = \mathbf{I}_{\Gamma_0}$, where $\forall i \, \gamma_i \in \Gamma_0$, $\gamma_i = Values(A_i) = Leaves(T_i)$, that is, the primitive values of the attributes $A_1 \cdots A_N$.*

Definition 4 (AVT-Induced Abstract Instance Space) *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 $\mathbf{I_A} = \cup_{\Gamma \in \boldsymbol{\Delta}} \mathbf{I}_\Gamma$ (the union of instance spaces induced by all of the the cuts through the set of AVTs $\mathbf{T}$).*

2.3 Partially Specified Data

Definition 5 (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 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, $\mathbf{D_T} \subseteq \mathbf{I_A} \times \mathbf{C}$.*

2.4 Learning Classifiers from Data

The problem of learning classifiers from AVT and data is a natural generalization of the problem of learning classifiers from data without AVT. The original data set D is simply a collection of labelled instances of the form (X_j, c_j) where $X_j \in \mathbf{I} = \times_i Values(A_i) = \times_i Leaves(T_i)$, and $c_j \in \mathbf{C}$ is a class label. A classifier is a hypothesis in the form of a function $h : \mathbf{I} \to \mathbf{C}$, 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., decision trees, Naive Bayes classifiers, SVM, etc.). The task of learning classifiers from the original data set D entails identifying a hypothesis $h \in \mathbf{H}$ that

satisfies some criteria (e.g., a hypothesis that is most likely given the training data D).

The problem of learning classifiers from AVT and 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 $h_T : \mathbf{I_A} \to \mathbf{C}$ for assigning appropriate class labels to each instance in the instance space $\mathbf{I_A}$. Of special interest are the cases 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.

3 AVT-Based Naïve Bayes Learner

3.1 Naïve Bayes Learner (NBL)

Suppose each attribute A_i takes a value from a finite set of values $Values(A_i)$. An instance X_p to be classified is represented as a tuple of attribute values $(v_{1p}, v_{2p}, \cdots, v_{Np})$ where each $v_{ip} \in Values(A_i)$. 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:

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

Hence, the task of the Naive Bayes Learner (NBL) is to estimate $\forall c_j \in \mathbf{C}$ and $\forall v_{i_k} \in Values(A_i)$, relevant class probabilities $p(c_j)$ and the class conditional probabilities $P(v_{i_k}|c_j)$ from training data D. These probabilities, which completely specify a Naive Bayes classifier, can be estimated from D using standard probability estimation methods [17] based on relative frequencies of the corresponding classes and attribute value and class label cooccurrences observed in D. These relative frequencies summarize *all* the information relevant for constructing a Naive Bayes classifier from a training set D, and hence constitute *sufficient statistics* for NBL [3, 4].

3.2 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 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 the 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 $h(\Gamma) = \{CPT(\gamma_1), CPT(\gamma_2), \cdots, CPT(\gamma_N)\}$.

If each cut $\gamma_i \in \Gamma$ is chosen to correspond to the primitive values of the respective attribute i.e., $\forall i\, \gamma_i = Leaves(T_i)$, $h(\Gamma)$ is simply the standard Naïve Bayes Classifier based on the attributes $A_1, A_2, \cdots, A_N$. If each cut $\gamma_i \in \Gamma$ is chosen to pass through the root of each AVT, i.e., $\forall i\, \gamma_i = \{Root(T_i)\}$, $h(\Gamma)$ simply assigns each instance to the class that is a priori most probable.

AVT-NBL starts 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 refines the classifier (hypothesis) using a criterion that is designed to trade-off between the accuracy of classification and the complexity of the resulting Naïve Bayes classifier. Successive refinements of Γ correspond to a partial ordering of Naive Bayes classifiers based on the structure of the AVTs in $\mathbf{T}$. For example, in Figure 2, $\hat{\Gamma}$ is a refinement of Γ, and hence corresponding hypothesis $h(\hat{\Gamma})$ is a refinement of $h(\Gamma)$

3.2.1 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 a 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. This procedure is a simplified case of EM (Expectation Maximization) algorithm to estimate expected sufficient statistics for $CCFC(A_i)$. 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 $u_l \in \pi(v, T_i)$:

$$\sigma_i(u_l|c_j) \leftarrow \sigma_i(u_l|c_j)\left(1 + \frac{\sigma_i(v|c_j) - \sum_{k=1}^{|\pi(v, T_i)|} \sigma_i(u_k|c_j)}{\sum_{k=1}^{|\pi(v, T_i)|} \sigma_i(u_k|c_j)}\right)$$

$$If \sum_{k=1}^{|\pi(v, T_i)|} \sigma_i(u_k|c_j) \neq 0;$$

$$\sigma_i(u_l|c_j) \leftarrow \left(\frac{\sigma_i(v|c_j)}{|\pi(v, T_i)|}\right) Otherwise.$$

Let $\Gamma = \{\gamma_1, \cdots, \gamma_N\}$ 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 Laplace estimates [17, 12].

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

Recall that the Naïve Bayes Classifier $h(\Gamma)$ based on a chosen set of cuts Γ is completely specified by the conditional probability tables associated with the cuts in Γ: $h(\Gamma) = \{CPT(\gamma_1), \cdots, CPT(\gamma_N)\}$.

3.2.2 Searching for a Compact Naïve Bayes Classifier

We use a variant of the minimum description length (MDL) principle [20] to capture the tradeoff between the complexity and accuracy of Naive Bayes classifiers that correspond to different choices of cuts through the AVTs. Friedman et al [8] 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

IEEE
COMPUTER
SOCIETY

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.

$$CMDL(h|D) = \left(\frac{\log|D|}{2}\right) size(h) - CLL(h|D)$$

$$where, \; CLL(h|D) = |D| \sum_{p=1}^{|D|} \log P_h(c_p|v_{1p}, \cdots, v_{Np})$$

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 the data D.

There are two cases in the calculation of the conditional likelihood $CLL(h|D)$ when D contains partially specified instances. The first case is 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. In this case, we can treat that instance as though it were fully specified relative to the Naïve Bayes classifier based on the cut γ of $CCFC(A_i)$ 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 $\hat{\Gamma}$. Construct corresponding hypothesis $h(\hat{\Gamma})$

         b. If $CMDL(h(\hat{\Gamma})|D) < CMDL(h(\Gamma)|D)$, replace Γ with $\hat{\Gamma}$ and γ_i with γ_i^v

      ii. $\delta_i \leftarrow \gamma_i$

4. Output $h(\Gamma)$

4 Alternative Approaches to Learning Classifiers from AVT and Data

Besides AVT-NBL, we can envision two alternative approaches to learning classifiers from AVT and data:

The first approach is to treat each partially specified (and hence partially missing) attribute value as if it were (totally) missing, and handle the resulting data set with missing attribute values 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 NBL.

A second approach to learn classifiers from AVT and data uses AVT to construct a set of Boolean attributes from each (original) attribute A_i, a set of boolean attributes corresponds to nodes 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. For example, the boolean attribute (Student Status = *Undergraduate*) is correlated with (Student Status = *Junior*). (Indeed, it is this correlation that enables us to exploit the information provided by AVT in learning classifiers from AVT and data). Thus, a Naïve Bayes classifier that would be optimal in the Maximal a Posteriori sense [14] when the original attributes are independent given class, would no longer be optimal when applied to *propositionalized* data sets because of the strong dependencies among the Boolean attributes derived from an AVT.

IEEE
COMPUTER
SOCIETY

5 Experiments and Results

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. We select 8 data sets (with only nominal attributes) from the UC Irvine Machine Learning Repository. For three of them (i.e., *Mushroom*, *Soybean*, and *Nursery*), AVTs were supplied by domain experts. For the rest data sets, the AVTs were generated using AVT-Learner, a Hierarchical Agglomerative Clustering (HAC) algorithm to construct AVTs [10].

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 algorithms on data sets with different percentages of totally missing and partially missing attribute values. Three 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 [25]. 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.

A third set of experiments were designed to investigate the performance of classifiers generated by AVT-NBL, Prop-NBL, and NBL as a function of the training set size. We divided each data set into two disjoint parts: a training pool and a test pool. Training sets of different sizes, corresponding to 10%, 20%, ..., 100% of the training pool, were sampled and used to train Naïve Bayes classifier using AVT-NBL, Prop-NBL, and NBL. The resulting classifiers were evaluated on the entire test pool. The experiment was repeated 9 times for each training set size. The entire process was repeated using 3 different random partitions of data into training and test pools. The accuracy of the learned classifiers on the examples in the test pool were averaged across the 9×3=27 runs.

5.1 Results

AVT-NBL yields 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 8 original 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, Soybean, Audiology* and *Zoo* 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 higher error rates than NBL. This can be explained by the fact that the Boolean features generated from an AVT are generally not independent given the class.

Table 1. Comparison of error rate and size of classifiers 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 ERROR	SIZE	PROP-NBL ERROR	SIZE	AVT-NBL 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 rates on data with partially or totally missing values

% Error rates using 10-fold cross validation with 90% confidence interval

DATA METHODS	PARTIALLY MISSING NBL	PROP-NBL	AVT-NBL	TOTALLY MISSING 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)

AVT-NBL yields classifiers that are substantially more compact than those generated by PROP-NBL and NBL. The shaded columns in Table 1 compare the total number of class conditional probabilities needed to specify the classifiers produced by AVT-NBL, NBL, and PROP-NBL on original data. 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.

AVT-NBL yields significantly lower error rates than NBL and PROP-NBL on partially specified data and data with totally missing values. Table 2 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.

AVT-NBL produces more accurate classifiers than NBL and Prop-NBL for a given training set size. Figure 3 shows the plot of the accuracy of the classifiers learned as a function of training set size for *Audiology* data. We obtained similar results on other benchmark data sets used in this study. Thus, AVT-NBL is *more efficient* than NBL and Prop-NBL in its use of training data.
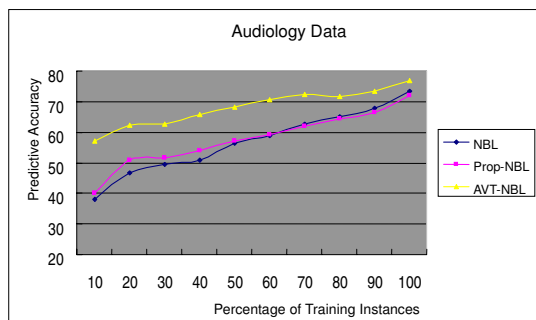

Figure 3. Classifier accuracy as a function of training set size

6 Summary and Discussion

6.1 Summary

In this paper, we have described AVT-NBL [1], an algorithm for learning classifiers from attribute value taxonomies (AVT) and data. Our experimental results show that AVT-NBL is able to generate classifiers that are substantially more compact and accurate than those produced by NBL on a broad range of data sets with different percentages of partially specified values. We also show that AVT-NBL is more efficient in its use of training data: AVT-NBL produces classifiers that outperform those produced by NBL using substantially fewer training examples. Thus, AVT-NBL offers an effective approach to learning compact (hence more comprehensible) accurate classifiers from data - including data that are *partially specified*. AVT-guided learning algorithms offer a promising approach to knowledge acquisition from autonomous, semantically heteroge-

---

[1] A Java implementation of AVT-NBL and the data sets and AVTs used in this study are available at:
http://www.cs.iastate.edu/˜jzhang/ICDM04/index.html

neous information sources, where domain specific AVTs are often available and data are often partially specified.

6.2 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 [25] for a review) desJardins et al [7] suggested the use of Abstraction-Based Search (ABS) to learn Bayesian networks with compact structure. Zhang and Honavar [25] 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 [6][13].

There is a large body of work on the use of domain theories to guide learning. AVT can be viewed as a restricted class of domain theories. 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.

Chen et al. [5] 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 [16] 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.

Automated construction of hierarchical taxonomies over attribute values and class labels is beginning to receive attention in the machine learning community. Examples include distributional clustering [19], extended FOCL and statistical clustering [23], information bottleneck [21]. Such algorithms provide a source of AVT in domains where none are available. The focus of work described in this paper is on algorithms that use AVT in learning classifiers from data.

6.3 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 [3][4].

(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

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] D. Caragea, J. Pathak, and V. Honavar. Learning Classifiers from Semantically Heterogeneous Data. In: Proceedings of the 3rd International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, ODBASE-2004.

[5] A. Chen, J. Chiu, and F. Tseng. Evaluating aggregate operations over imprecise data. IEEE Trans. On Knowledge and Data Engineering, 8, 1996.

[6] A. Clare, R. King. Knowledge Discovery in Multi-label Phenotype Data. In: Lecture Notes in Computer Science. Vol. 2168, 2001.

[7] M. desJardins, L. Getoor, D. Koller. Using Feature Hierarchies in Bayesian Network Learning. Lecture Notes in Artificial Intelligence 1864, 2000.

[8] N. Friedman, D. Geiger. Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning, Vol: 29, 1997.

[9] D. Haussler. Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. Artificial Intelligence, 36, 1988.

[10] 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.

[11] R. Kohavi, P. Provost. Applications of Data Mining to Electronic Commerce. Data Mining and Knowledge Discovery, Vol. 5, 2001.

[12] R. Kohavi, B. Becker, D. Sommerfield. Improving simple Bayes. Tech. Report, Data Mining and Visualization Group, Silicon Graphics Inc., 1997.

[13] D. Koller, M. Sahami. Hierarchically classifying documents using very few words. In: Proceedings of the 14th Int'l Conference on Machine Learning, 1997.

[14] P. Langley, W. Iba, K. Thompson. An analysis of Bayesian classifiers Proceedings of the Tenth National Conference on Artificial Intelligence, 1992.

[15] 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.

[16] S. McClean, B. Scotney, M. Shapcott. Aggregation of Imprecise and Uncertain Information in Databases. IEEE Transactions on Knowledge and Data Engineering (6), 2001.

[17] T. Mitchell. Machine Learning. Addison-Wesley, 1997.

[18] 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.

[19] F. Pereira, N. Tishby, L. Lee. Distributional clustering of English words. In: Proceedings of the Thirty-first Annual Meeting of the Association for Computational Linguistics, 1993.

[20] J. Rissanen. Modeling by shortest data description. Automatica, vol. 14, 1978.

[21] N. Slonim, N. Tishby. Document Clustering using Word Clusters via the Information Bottleneck Method. ACM SIGIR, 2000.

[22] 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.

[23] T. Yamazaki, M. Pazzani, C. Merz. Learning Hierarchies from Ambiguous Natural Language Data. In: Proceedings of the 12th Int'l Conference on Machine Learning, 1995.

[24] 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.

[25] 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.