

# Generation of Attribute Value Taxonomies from Data for Data-Driven Construction of Accurate and Compact Classifiers

Dae-Ki Kang, Adrian Silvescu, Jun Zhang, and Vasant Honavar  
Artificial Intelligence Research Laboratory  
Department of Computer Science  
Iowa State University, Ames, IA 50011 USA  
{dkkang, silvescu, junzhang, honavar}@iastate.edu

## Abstract

Attribute Value Taxonomies (AVT) have been shown to be useful in constructing compact, robust, and comprehensible classifiers. However, in many application domains, human-designed AVTs are unavailable. We introduce AVT-Learner, an algorithm for automated construction of attribute value taxonomies from data. AVT-Learner uses Hierarchical Agglomerative Clustering (HAC) to cluster attribute values based on the distribution of classes that co-occur with the values. We describe experiments on UCI data sets that compare the performance of AVT-NBL (an AVT-guided Naive Bayes Learner) with that of the standard Naive Bayes Learner (NBL) applied to the original data set. Our results show that the AVTs generated by AVT-Learner are competitive with human-generated AVTs (in cases where such AVTs are available). AVT-NBL using AVTs generated by AVT-Learner achieves classification accuracies that are comparable to or higher than those obtained by NBL; and the resulting classifiers are significantly more compact than those generated by NBL.

## 1. Introduction

An important goal of inductive learning is to generate accurate and compact classifiers from data. 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). For example, Figure 1 shows a human-made taxonomy associated with the nominal attribute ‘Odor’ of the UC Irvine AGARICUS-LEPIOTA mushroom data set [5].

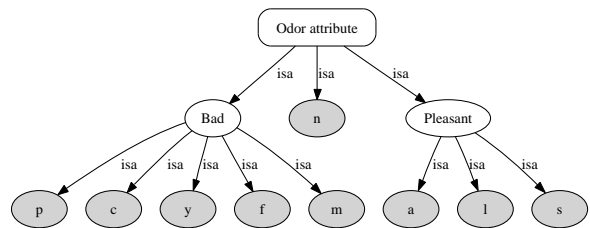


Figure 1. Human-made AVT from ‘odor’ attribute of UCI AGARICUS-LEPIOTA mushroom data set.

Hierarchical groupings of attribute values (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. [24] 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. Several ontologies being developed as part of the Semantic Web related efforts [4] also capture hierarchical groupings of attribute values. Kohavi and Provost [15] have noted the need to be able to incorporate background knowledge in the form of hierarchies over data attributes in electronic commerce applications of data mining.

There are several reasons for exploiting AVT in learning classifiers from data, perhaps the most important being a preference for comprehensible and simple, yet accurate and robust classifiers [18] in many practical applications of data mining. 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 hierarchically related values. Thus, the rule (*odor = pleasant*)  $\rightarrow$  (*class = edible*) is likely to be preferred over ((*odor =*

$a) \wedge (\text{color} = \text{brown}) \vee ((\text{odor} = l) \wedge (\text{color} = \text{brown})) \vee ((\text{odor} = s) \wedge (\text{color} = \text{brown})) \rightarrow (\text{class} = \text{edible})$  by a user who is familiar with the odor taxonomy shown in Figure 1.

Another reason for exploiting AVTs in learning classifiers from data arises from the necessity, in many application domains, for learning from small data sets where there is a greater chance of generating classifiers that over-fit the training data. A common approach used by statisticians when estimating from small samples involves *shrinkage* [7] or grouping attribute values (or more commonly class labels) into bins, when there are too few instances that match any specific attribute value or class label, to estimate the relevant statistics with adequate confidence. Learning algorithms that exploit AVT can potentially perform *shrinkage* automatically thereby yielding robust classifiers. In other words, exploiting information provided by an AVT can be an effective approach to performing regularization to minimize over-fitting [28].

Consequently, several algorithms for learning classifiers from AVTs and data have been proposed in the literature. This work has shown that AVTs can be exploited to improve the accuracy of classification and in many instances, to reduce the complexity and increase the comprehensibility of the resulting classifiers [6, 11, 14, 23, 28, 30]. Most of these algorithms exploit AVTs to represent the information needed for classification at different levels of abstraction.

However, in many domains, AVTs specified by human experts are unavailable. Even when a human-supplied AVT is available, it is interesting to explore whether alternative groupings of attribute values into an AVT might yield more accurate or more compact classifiers. Against this background, we explore the problem of automated construction of AVTs from data. In particular, we are interested in AVTs that are useful for generating accurate and compact classifiers.

## 2. Learning attribute value taxonomies from data

### 2.1. Learning AVT from data

We describe AVT-Learner, an algorithm for automated construction of AVT from a data set of instances wherein each instance is described by an ordered tuple of  $N$  nominal attribute values and a class label.

Let  $A = \{A_1, A_2, \dots, A_n\}$  be a set of nominal attributes. Let  $V_i = \{v_i^1, v_i^2, \dots, v_i^{m_i}\}$  be a finite domain of mutually exclusive values associated with attribute  $A_i$  where  $v_i^j$  is the  $j^{\text{th}}$  attribute value of  $A_i$  and  $m_i$  is the number possible number of values of  $A_i$ , that is,  $|V_i|$ . We say that  $V_i$  is the set of primitive values of attribute  $A_i$ . Let

$C = \{C_1, C_2, \dots, C_k\}$  be a set of mutually disjoint class labels. A data set is  $D \subseteq V_1 \times V_2 \times \dots \times V_n \times C$ .

Let  $T = \{T_1, T_2, \dots, T_n\}$  denote a set of AVT such that  $T_i$  is an AVT associated with the attribute  $A_i$ , and  $Leaves(T_i)$  denote a set of all leaf nodes in  $T_i$ . We define a cut  $\delta_i$  of an AVT  $T_i$  to be a subset of nodes in  $T_i$  satisfying the following two properties: (1) For any leaf  $l \in Leaves(T_i)$ , either  $l \in \delta_i$  or  $l$  is a descendant of a node  $n \in \delta_i$ ; and (2) for any two nodes  $f, g \in \delta_i$ ,  $f$  is neither a descendant nor an ancestor of  $g$  [12]. For example,  $\{Bad, a, l, s, n\}$  is a cut through the AVT for *odor* shown in Figure 1. Note that a cut through  $T_i$  corresponds to a partition of the values in  $V_i$ . Let  $\Delta = \{\delta_1, \delta_2, \dots, \delta_n\}$  be a set of cuts associated with AVTs in  $T = \{T_1, T_2, \dots, T_n\}$ .

The problem of learning AVTs from data can be stated as follows: given a data set  $D \subseteq V_1 \times V_2 \times \dots \times V_n \times C$  and a measure of dissimilarity (or equivalently similarity) between any pair of values of an attribute, output a set of AVTs  $T = \{T_1, T_2, \dots, T_n\}$  such that each  $T_i$  (AVT associated with the attribute  $A_i$ ) corresponds to a hierarchical grouping of values in  $V_i$  based on the specified similarity measure.

We use hierarchical agglomerative clustering (HAC) of the attribute values according to the distribution of classes that co-occur with them. Let  $DM(P(x)||P(y))$  denote a measure of pairwise divergence between two probability distributions  $P(x)$  and  $P(y)$  where the random variables  $x$  and  $y$  take values from the same domain. We use the pairwise divergence between the distributions of class labels associated with the corresponding attribute values as a measure of the dissimilarity between the attribute values. The lower the divergence between the class distributions associated with two attributes, the greater is their similarity. The choice of this measure of dissimilarity between attribute values is motivated by the intended use of the AVT, namely, the construction of accurate, compact, and robust classifiers. If two values of an attribute are indistinguishable from each other with respect to their class distributions, they provide statistically similar information for classification of instances.

The algorithm for learning AVT for a nominal attribute is shown in Figure 2. The basic idea behind AVT-Learner is to construct an AVT  $T_i$  for each attribute  $A_i$  by starting with the primitive values in  $V_i$  as the leaves of  $T_i$  and recursively add nodes to  $T_i$  one at a time by merging two existing nodes. To aid this process, the algorithm maintains a cut  $\delta_i$  through the AVT  $T_i$ , updating the cut  $\delta_i$  as new nodes are added to  $T_i$ . At each step, the two attribute values to be grouped together to obtain an abstract attribute value to be added to  $T_i$  are selected from  $\delta_i$  based on the divergence between the class distributions associated with the corresponding values. That is, a pair of attribute values in  $\delta_i$  are merged if they have more similar class distributions than any other pair of

### AVT-Learner:

#### begin

1. **Input** : data set D
  2. For each attribute  $A_i$ :
  3. For each attribute value  $v_i^j$ :
  4. For each class label  $c_k$ : estimate the probability  $p(c_k|v_i^j)$
  5. Let  $P(C|v_i^j) = \{p(c_1|v_i^j), \dots, p(c_k|v_i^j)\}$  be the class distribution associated with value .
  6. Set  $\delta_i \leftarrow V_i$ ; Initialize  $T_i$  with nodes in  $\delta_i$ .
  7. Iterate until  $|\delta_i| = 1$ :
  8. In  $\delta_i$ , find  $(x, y) = \underset{\text{argmin}}{\{DM(P(C|v_i^x) || P(C|v_i^y))\}}$
  9. Merge  $v_i^x$  and  $v_i^y$  ( $x \neq y$ ) to create a new value  $v_i^{xy}$ .
  10. Calculate probability distribution  $P(C|v_i^{xy})$ .
  11.  $\lambda_i \leftarrow \delta_i \cup \{v_i^{xy}\} \setminus \{v_i^x, v_i^y\}$ .
  12. Update  $T_i$  by adding nodes  $v_i^{xy}$  as a parent of  $v_i^x$  and  $v_i^y$ .
  13.  $\delta_i \leftarrow \lambda_i$ .
  14. **Output** :  $T = \{T_1, T_2, \dots, T_n\}$
- #### end.

Figure 2. Pseudo-code of AVT-Learner

attribute values in  $\delta_i$ . This process terminates when the cut  $\delta_i$  contains a single value which corresponds to the root of  $T_i$ . If  $|V_i| = m_i$ , the resulting  $T_i$  will have  $(2m_i - 1)$  nodes when the algorithm terminates.

In the case of continuous-valued attributes, we define intervals based on observed values for the attribute in the data set. We then generate a hierarchical grouping of adjacent intervals, selecting at each step two adjacent intervals to merge using the pairwise divergence measure. A cut through the resulting AVT corresponds to a discretization of the continuous-valued attribute. A similar approach can be used to generate AVT from ordinal attribute values.

## 2.2. Pairwise divergence measures

There are several ways to measure similarity between two probability distributions. We have tested thirteen divergence measures for probability distributions  $P$  and  $Q$ . In this paper, we limit the discussion to Jensen-Shannon divergence measure.

**Jensen-Shannon divergence** [21] is weighted information gain, also called Jensen difference divergence, information radius, Jensen difference divergence, and Sibson-Burbea-Rao Jensen Shannon divergence. It is given by:

$$I(P||Q) = \frac{1}{2} \left[ \sum p_i \log \left( \frac{2p_i}{p_i + q_i} \right) + \sum q_i \log \left( \frac{2q_i}{p_i + q_i} \right) \right]$$

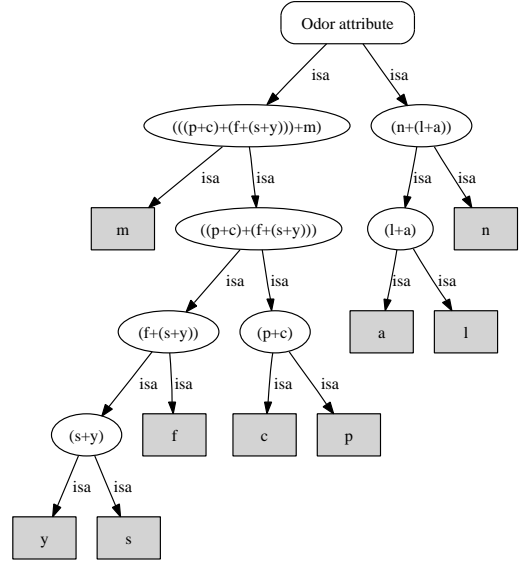


Figure 3. AVT of 'odor' attribute of UCI AGARICUS-LEPIOTA mushroom data set generated by AVT-Learner using Jensen-Shannon divergence (binary clustering)

Jensen-Shannon divergence is reflexive, symmetric and bounded. Figure 3 shows an AVT of 'odor' attribute generated by AVT-Learner (with binary clustering).

## 3. Evaluation of AVT-Learner

The intuition behind our approach to evaluating the AVT generated by AVT-Learner is the following: an AVT that captures relevant relationships among attribute values can result in the generation of simple and accurate classifiers from data, just as an appropriate choice of axioms in a mathematical domain can simplify proofs of theorems. Thus, the simplicity and predictive accuracy of the learned classifiers based on alternative choices of AVT can be used to evaluate the utility of the corresponding AVT in specific contexts.

### 3.1. AVT guided variants of standard learning algorithms

It is possible to extend standard learning algorithms in principled ways so as to exploit the information provided by AVT. AVT-DTL [26, 30, 28] and the AVT-NBL [29] which extend the decision tree learning algorithm [20] and the Naive Bayes learning algorithm [16] respectively are examples such algorithms.

The basic idea behind AVT-NBL is to start with the Naive Bayes Classifier that is based on the most abstract at-

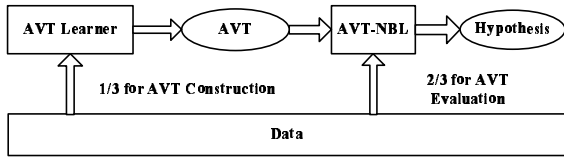


Figure 4. Evaluation of AVT using AVT-NBL

tribute values in AVTs and successively refine the classifier by a scoring function - a Conditional Minimum Description Length (CMDL) score suggested by Friedman et al. [8] to capture trade-off between the accuracy of classification and the complexity of the resulting Naive Bayes classifier.

The experiments reported by Zhang and Honavar [29] using several benchmark data sets show that AVT-NBL is able to learn, using human generated AVT, substantially more accurate classifiers than those produced by Naive Bayes Learner (NBL) applied directly to the data sets as well as NBL applied to data sets represented using a set of binary features that correspond to the nodes of the AVT (PROP-NBL). The classifiers generated by AVT-NBL are substantially more compact than those generated by NBL and PROP-NBL. These results hold across a wide range of missing attribute values in the data sets. Hence, the performance of Naive Bayes classifiers generated by AVT-NBL when supplied with AVT generated by the AVT-Learner provide useful measures of the effectiveness of AVT-Learner in discovering hierarchical groupings of attribute values that are useful in constructing compact and accurate classifiers from data.

## 4. Experiments

### 4.1. Experimental setup

Figure 4 shows the experimental setup. The AVT generated by the AVT-Learner are evaluated by comparing the performance of the Naive Bayes Classifiers produced by applying

- NBL to the original data set
- AVT-NBL to the original data set (See Figure 4).

For the benchmark data sets, we chose 37 data sets from UCI data repository [5].

Among the data sets we have chosen, AGARICUS-LEPIOTA data set and NURSERY data set have AVT supplied by human experts. AVT for AGARICUS-LEPIOTA data was prepared by a botanist, and AVT for NURSERY data was based on our understanding of the domain. We are not aware of any expert-generated AVTs for other data sets.

In each experiment, we randomly divided each data set into 3 equal parts and used 1/3 of the data for AVT construction using AVT-Learner. The remaining 2/3 of the data were used for generating and evaluating the classifier. Each set of AVTs generated by the AVT-Learner was evaluated in terms of the error rate and the size of the resulting classifiers (as measured by the number of entries in conditional probability tables). The error rate and size estimates were obtained using 10-fold cross-validation on the part of the data set (2/3) that was set aside for evaluating the classifier. The results reported correspond to averages of the 10-fold cross-validation estimates obtained from the three choices of the AVT-construction and AVT-evaluation. This process ensures that there is no information leakage between the data used for AVT construction, and the data used for classifier construction and evaluation.

10-fold cross-validation experiments were performed to evaluate human expert-supplied AVT on the AVT evaluation data sets used in the experiments described above for the AGARICUS-LEPIOTA data set and the NURSERY data set.

We also evaluated the robustness of the AVT generated by the AVT-Learner by using them to construct classifiers from data sets with varying percentages of missing attribute values. The data sets with different percentages of missing values were generated by uniformly sampling from instances and attributes to introduce the desired percentage of missing values.

### 4.2. Results

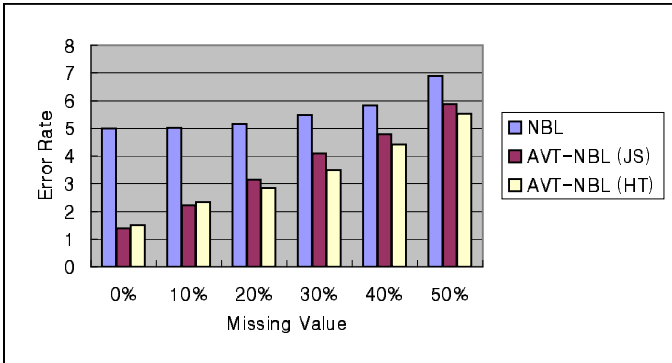
#### AVT generated by AVT-Learner are competitive with human-generated AVT when used by AVT-NBL.

The results of our experiments shown in Figure 5 indicate that AVT-Learner is effective in constructing AVTs that are competitive with human expert-supplied AVTs for use in classification tasks with respect to the error rates and the size of the resulting classifiers.

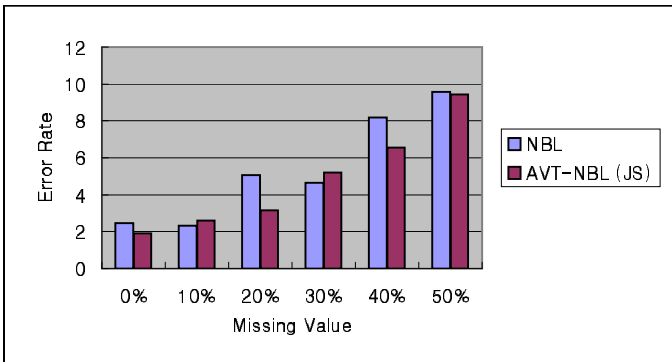
#### AVT-Learner can generate useful AVT when no human-generated AVT are available.

For most of the data sets, there are no human-supplied AVT's available. Figure 6 shows the error rate estimates for Naive Bayes classifiers generated by AVT-NBL using AVT generated by the AVT-Learner and the classifiers generated by NBL applied to the DERMATOLOGY data set. The results shown suggest that AVT-Learner, using Jensen-Shannon divergence, is able to generate AVTs that when used by AVT-NBL, result in classifiers that are more accurate than those generated by NBL.

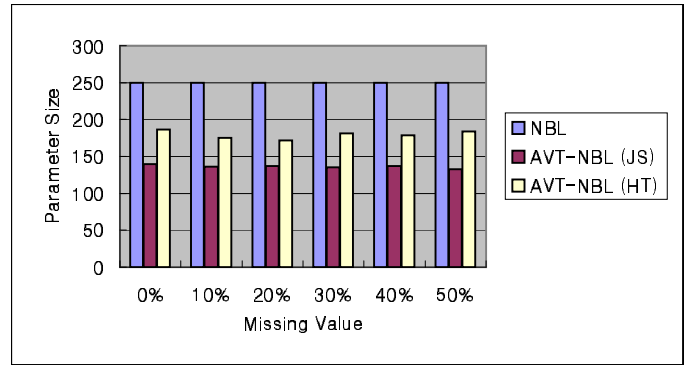
Additional experiments with other data sets produced similar results. Table 1 shows the classifier's accuracy on original UCI data sets for NBL and AVT-NBL that uses



**Figure 5.** The estimated error rates of classifiers generated by NBL and AVT-NBL on AGARICUS-LEPIOTA data with different percentages of missing values. HT stands for human-supplied AVT. JS denotes AVT constructed by AVT-Learner using Jensen-Shannon divergence.



**Figure 6.** The error rate estimates of the Standard Naive Bayes Learner (NBL) compared with that of AVT-NBL on DERMATOLOGY data. JS denotes AVT constructed by AVT-Learner using Jensen-Shannon divergence.



**Figure 7.** The size (as measured by the number of parameters) of the Standard Naive Bayes Learner (NBL) compared with that of AVT-NBL on AGARICUS-LEPIOTA data. HT stands for human-supplied AVT. JS denotes AVT constructed by AVT-Learner using Jensen-Shannon divergence.

AVTs generated by AVT-Learner. 10-fold cross-validation is used for evaluation, and Jensen-Shannon divergence is used for AVT generation. The user-specified number for discretization is 10.

Thus, AVT-Learner is able to generate AVTs that are useful for constructing compact and accurate classifiers from data.

### AVT generated by AVT-Learner, when used by AVT-NBL, yield substantially more compact Naive Bayes Classifiers than those produced by NBL

Naive Bayes classifiers constructed by AVT-NBL generally have smaller number of parameters than those from NBL (See Figures 7 for representative results). Table 2 shows the classifier size measured by the number of parameters on selected UCI data sets for NBL and AVT-NBL that uses AVTs generated by AVT-Learner.

These results suggest that AVT-Learner is able to group attribute values into AVT in such a way that the resulting AVT, when used by AVT-NBL, result in compact yet accurate classifiers.

## 5. Summary and discussion

### 5.1. Summary

In many applications of data mining, there is a strong preference for classifiers that are both accurate and compact [15, 18]. Previous work has shown that attribute value taxonomies can be exploited to generate such classifiers from data [28, 29]. However, human-generated AVTs are

**Table 1. Accuracy of NBL and AVT-NBL on UCI data sets**

Data	NBL	AVT-NBL
Anneal	86.3029	98.9978
Audiology	73.4513	76.9912
Autos	56.0976	86.8293
Balance-scale	90.4	91.36
Breast-cancer	71.6783	72.3776
Breast-w	95.9943	97.2818
Car	85.5324	86.169
Colic	77.9891	83.4239
Credit-a	77.6812	86.5217
Credit-g	75.4	75.4
Dermatology	97.8142	98.0874
Diabetes	76.3021	77.9948
Glass	48.5981	80.8411
Heart-c	83.4983	87.1287
Heart-h	83.6735	86.3946
Heart-statlog	83.7037	86.6667
Hepatitis	84.5161	92.9032
Hypothyroid	95.281	95.7847
Ionosphere	82.6211	94.5869
Iris	96	94.6667
Kr-vs-kp	87.8911	87.9224
Labor	89.4737	89.4737
Letter	64.115	70.535
Lymph	83.1081	84.4595
Mushroom	95.8272	99.5938
Nursery	90.3241	90.3241
Primary-tumor	50.1475	47.7876
Segment	80.2165	90
Sick	92.6829	97.8261
Sonar	67.7885	99.5192
Soybean	92.9722	94.5827
Splice	95.3605	95.768
Vehicle	44.7991	67.8487
Vote	90.1149	90.1149
Vowel	63.7374	42.4242
Waveform-5000	80	65.08
Zoo	93.0693	96.0396

**Table 2. Parameter size of NBL and AVT-NBL on selected UCI data sets**

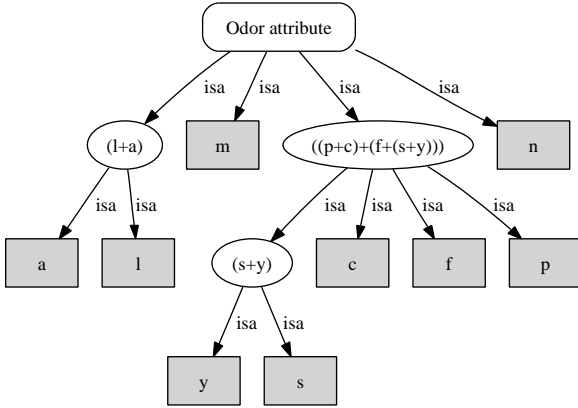
Data	NBL	AVT-NBL
Audiology	3720	3600
Breast-cancer	104	62
Car	88	80
Dermatology	906	540
Kr-vs-kp	150	146
Mushroom	252	124
Nursery	140	125
Primary-tumor	836	814
Soybean	1919	1653
Splice	864	723
Vote	66	66
Zoo	259	238

unavailable in many application domains. Manual construction of AVTs requires a great deal of domain expertise, and in case of large data sets with many attributes and many values for each attribute, manual generation of AVTs is extremely tedious and hence not feasible in practice. Against this background, we have described in this paper, AVT-Learner, a simple algorithm for automated construction of AVT from data. AVT-Learner recursively groups values of attributes based on a suitable measure of divergence between the class distributions associated with the attribute values to construct an AVT. AVT-Learner is able to generate hierarchical taxonomies of nominal, ordinal, and continuous valued attributes. The experiments reported in this paper show that:

- AVT-Learner is effective in generating AVTs that when used by AVT-NBL, a principled extension of the standard algorithm for learning Naive Bayes classifiers, result in classifiers that are substantially more compact (and often more accurate) than those obtained by the standard Naive Bayes Learner (that does not use AVTs).
- The AVTs generated by AVT-Learner are competitive with human supplied AVTs (in the case of benchmark data sets where human-generated AVTs were available) in terms of both the error rate and size of the resulting classifiers.

## 5.2. Discussion

The AVTs generated by AVT-Learner are binary trees. Hence, one might wonder if k-ary AVTs yield better results when used with AVT-NBL. Figure 8 shows an AVT of ‘odor’ attribute generated by AVT-Learner (with quaternary



**Figure 8. AVT of ‘odor’ attribute of UCI AGARICUS-LEPIOTA mushroom data set generated by AVT-Learner using Jensen-Shannon divergence (with quaternary clustering)**

**Table 3. Accuracy of NBL and AVT-NBL for k-ary AVT-Learner**

Data	2-ary	3-ary	4-ary
Nursery	90.3241	90.3241	90.3241
Audiology	76.9912	76.5487	76.9912
Car	86.169	86.169	86.169
Dermatology	98.0874	97.541	97.541
Mushroom	99.5938	99.7292	99.7538
Soybean	94.5827	94.4363	94.4363

clustering). Table 3 shows the accuracy of AVT-NBL when k-ary clustering is used by AVT-Learner. It can be seen that AVT-NBL generally works best when binary AVTs are used. It is because reducing internal nodes in AVT-Learner will eventually reduce the search space for possible cuts in AVT-NBL, which leads to generating a less compact classifier.

### 5.3. Related work

Gibson and Kleinberg [10] introduced STIRR, an iterative algorithm based on non-linear dynamic systems for clustering categorical attributes. Ganti et. al. [9] designed CACTUS, an algorithm that uses intra-attribute summaries to cluster attribute values. However, both of them did not make taxonomies and use the generated for improving classification tasks. Pereira et. al. [19] described distributional clustering for grouping words based on class distributions associated with the words in text classification. Yamazaki et al., [26] described an algorithm for extracting hierar-

chical groupings from rules learned by FOCL (an inductive learning algorithm) [17] and reported improved performance on learning translation rules from examples in a natural language processing task. Slonim and Tishby [21, 22] described a technique (called the agglomerative information bottleneck method) which extended the distributional clustering approach described by Pereira et al. [19], using Jensen-Shannon divergence for measuring distance between document class distributions associated with words and applied it to a text classification task. Baker and McCallum [3] reported improved performance on text classification using a technique similar to distributional clustering and a distance measure, which upon closer examination, can be shown to be equivalent to Jensen-Shannon divergence [21].

To the best of our knowledge, there has been little work on the evaluation of techniques for generating hierarchical groupings of attribute values (AVTs) on classification tasks using a broad range of benchmark data sets using algorithms such as AVT-DTL or AVT-NBL that are capable of exploiting AVTs in learning classifiers from data.

### 5.4. Future work

Some directions for future work include:

- Extending AVT-Learner described in this paper to learn AVTs that correspond to tangled hierarchies (which can be represented by directed acyclic graphs (DAG) instead of trees).
- Learning AVT from data for a broad range of real world applications such as census data analysis, text classification, intrusion detection from system log data [13], learning classifiers from relational data [2], and protein function classification [25] and identification of protein-protein interfaces [27].
- Developing algorithms for learning hierarchical ontologies based on part-whole and other relations as opposed to ISA relations captured by an AVT.
- Developing algorithms for learning hierarchical groupings of values associated with more than one attribute.

## 6. Acknowledgments

This research was supported in part by grants from the National Science Foundation (IIS 0219699) and the National Institutes of Health (GM 066387). The authors wish to thank members of the Iowa State University Artificial Intelligence Laboratory and anonymous referees for their helpful comments on earlier drafts of this paper.

## References

- [1] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25(1):25–29, 2000.
- [2] A. Atramentov, H. Leiva, and V. Honavar. A multi-relational decision tree learning algorithm - implementation and experiments. In T. Horváth and A. Yamamoto, editors, *Proceedings of the 13th International Conference on Inductive Logic Programming (ILP 2003)*. Vol. 2835 of *Lecture Notes in Artificial Intelligence* : Springer-Verlag, pages 38–56, 2003.
- [3] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM Press, 1998.
- [4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
- [5] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.
- [6] V. Dhar and A. Tuzhilin. Abstract-driven pattern discovery in databases. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):926–938, 1993.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
- [8] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29(2-3):131–163, 1997.
- [9] V. Ganti, J. Gehrke, and R. Ramakrishnan. Cactus - clustering categorical data using summaries. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 73–83. ACM Press, 1999.
- [10] D. Gibson, J. M. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. *VLDB Journal: Very Large Data Bases*, 8(3-4):222–236, 2000.
- [11] J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AIII Press/MIT Press, 1996.
- [12] D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant’s learning framework. *Artificial intelligence*, 36:177 – 221, 1988.
- [13] G. Helmer, J. S. K. Wong, V. G. Honavar, and L. Miller. Automated discovery of concise predictive rules for intrusion detection. *J. Syst. Softw.*, 60(3):165–175, 2002.
- [14] J. Hendler, K. Stoffel, and M. Taylor. Advances in high performance knowledge representation. Technical Report CS-TR-3672, University of Maryland Institute for Advanced Computer Studies Dept. of Computer Science, 1996.
- [15] R. Kohavi and F. Provost. Applications of data mining to electronic commerce. *Data Min. Knowl. Discov.*, 5(1-2):5–10, 2001.
- [16] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *National Conference on Artificial Intelligence*, pages 223–228, 1992.
- [17] M. Pazzani and D. Kibler. The role of prior knowledge in inductive learning. *Machine Learning*, 9:54–97, 1992.
- [18] M. J. Pazzani, S. Mani, and W. R. Shankle. Beyond concise and colorful: Learning intelligible rules. In *Knowledge Discovery and Data Mining*, pages 235–238, 1997.
- [19] F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *31st Annual Meeting of the ACL*, pages 183–190, 1993.
- [20] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.
- [21] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proceedings of the 13th Neural Information Processing Systems (NIPS 1999)* , 1999.
- [22] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–215. ACM Press, 2000.
- [23] M. Taylor, K. Stoffel, , and J. Hendler. Ontology based induction of high level classification rules. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1997.
- [24] J. L. Undercoffer, A. Joshi, T. Finin, and J. Pinkston. A Target Centric Ontology for Intrusion Detection: Using DAML+OIL to Classify Intrusive Behaviors. *Knowledge Engineering Review*, January 2004.
- [25] X. Wang, D. Schroeder, D. Dobbs, and V. G. Honavar. Automated data-driven discovery of motif-based protein function classifiers. *Inf. Sci.*, 155(1-2):1–18, 2003.
- [26] T. Yamazaki, M. J. Pazzani, and C. J. Merz. Learning hierarchies from ambiguous natural language data. In *International Conference on Machine Learning*, pages 575–583, 1995.
- [27] C. Yan, D. Dobbs, and V. Honavar. Identification of surface residues involved in protein-protein interaction – a support vector machine approach. In A. Abraham, K. Franke, and M. Koppen, editors, *Intelligent Systems Design and Applications (ISDA-03)*, pages 53–62, 2003.
- [28] J. Zhang and V. Honavar. Learning decision tree classifiers from attribute value taxonomies and partially specified data. In *the Twentieth International Conference on Machine Learning (ICML 2003)*, Washington, DC, 2003.
- [29] J. Zhang and V. Honavar. AVT-NBL: An algorithm for learning compact and accurate naive bayes classifiers from attribute value taxonomies and data. In *International Conference on Data Mining (ICDM 2004)*, 2004. To appear.
- [30] J. Zhang, A. Silvescu, and V. Honavar. Ontology-driven induction of decision trees at multiple levels of abstraction. In *Proceedings of Symposium on Abstraction, Reformulation, and Approximation 2002*. Vol. 2371 of *Lecture Notes in Artificial Intelligence* : Springer-Verlag, 2002.