

Generation of Attribute Value Taxonomies from Data and Their Use in Data-Driven Construction of Accurate and Compact Naive Bayes 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, jzhang, honavar}@cs.iastate.edu

Abstract. Attribute Value Taxonomies (AVT) have been shown to be useful in constructing compact and robust classifiers. However, in many application domains, human-designed AVTs are unavailable. For this problem, 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 of AVT-Learner on several benchmark 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 as well as a data set generated by augmenting the original data set with a set of additional attributes corresponding to the nodes in the AVTs. 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 that 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 ‘*Odor*’ attribute of the UC Irvine AGARICUS-LEPIOTA mushroom data set [1].

Hierarchical groupings of attribute values (AVT) are quite common in several application domains [2] and represent one of the most common types of ontologies. There are several reasons for exploiting AVT in learning classifiers from data:

- Preference for simple yet accurate and robust classifiers [3] that are expressed in terms of *abstract* attribute values.

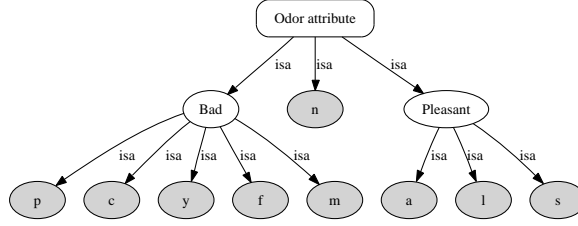


Fig. 1. Human-made AVT from ‘odor’ attribute of UCI AGARICUS-LEPIOTA mushroom data set.

- Exploiting information provided by an AVT can be an effective approach to performing regularization to minimize over-fitting [4].

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 [4–8]. 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^{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 let $Leaves(T_i)$ denote a set of all leaf nodes in T_i . We define a cut δ_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 descendent of a node $n \in \delta_i$; and (2) for any two nodes $f, g \in \delta_i$, f is neither a descendent nor an ancestor of g [9].

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) [10] 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 pair-wise 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 pair-wise divergence between the distributions of class labels associated with the corresponding attribute values as a measure of the dissimilarity between the attribute values. Thus, two values of an attribute are considered to be more similar to each other than any other pair of values if their class distributions are more similar to each other than the class distributions associated with any other pair of values for the same attribute. 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.

Our algorithm for learning AVT 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 δ_i through the AVT T_i , updating the cut δ_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 δ_i based on the divergence between the class distributions associated with the corresponding values. That is, a pair of attribute values in δ_i are merged if they have more similar class distributions than any other pair of attribute values in δ_i . This process terminates when the cut δ_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. The algorithm can be easily generalized to more than two branching factors.

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 [11] is weighted information gain, also called Jensen difference divergence, information radius, Jensen difference divergence, and Sibson-Burbea-Rao Jensen Shannon divergence. Jensen-Shannon divergence is reflexive, sym-

```

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
the values.
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  $(x, y) = \operatorname{argmin} \{DM(P(C|v_i^x) || P(C|v_i^y))\}$ 
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.

```

Fig. 2. Pseudo-code of AVT-Learner

metric and bounded. It is given by:

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

3 Evaluation of AVT

The intuition behind our approach to evaluating the AVT generated by AVT-Learner is the following: An AVT that captures the 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. There are at least two ways to exploit AVT in learning classifiers from data.

AVT-based propositionalization methods In propositionalization method, the data set is represented using a set of Boolean attributes obtained from the AVT T_i of the 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 L Boolean attributes where

$$L = \left(\sum_{i=1}^n |Nodes(T_i)| \right)$$

An advantage of the AVT-based propositionalization methods is that they require no modification to the learning algorithm. The statistical dependence among the Boolean attributes in the propositionalized representation of the original data set can degrade the performance of classifiers e.g., Naive Bayes that rely on independence of attributes given class. Hence, algorithms that extend standard learning algorithms so as to exploit the information supplied by an AVT without violating the underlying assumptions of the algorithm are of interest [2].

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 [12, 8, 4] and the AVT-NBL [2] which extend the decision tree learning algorithm [13] and the Naive Bayes learning algorithm [14] respectively are examples of such algorithms.

The standard algorithm (NBL) for learning a Naive 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 $|V_i||C|$ entries, where C is a set of class labels. The probabilities are typically estimated using a Bayesian approach [15].

AVT-NBL is a natural extension of the standard algorithm for learning a Naive Bayes classifier from data [2]. AVT-NBL starts with the Naive Bayes Classifier that is based on the most abstract value of each attribute and successively refines the classifier using a criterion that is designed to tradeoff between the accuracy of classification and the complexity of the resulting Naive Bayes classifier.

The experiments reported by Zhang and Honavar [2] using several benchmark data sets show that AVT-NBL is able to learn using human generated AVT, substantially more accurate Naive Bayes 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 augmented features obtained by propositionalization using 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 without human intervention for building AVT.

4 Experiments

4.1 Experimental setup

Figure 3 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

- NBL to a propositionalized version of the data set (where propositionalization introduces additional features based on the AVT) (See Figure 3(a))
- AVT-NBL to the original data set (See Figure 3(b)).

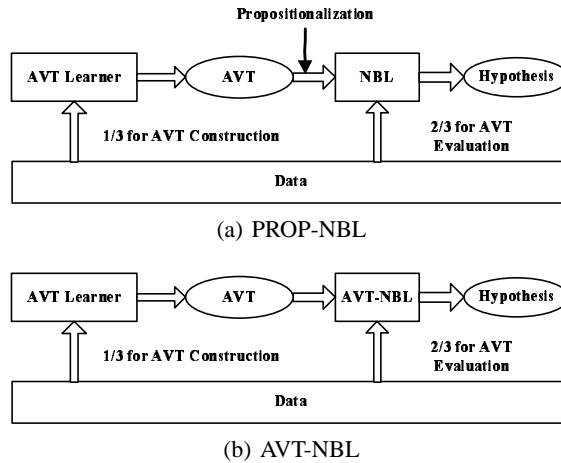


Fig. 3. Evaluation of AVT using PROP-NBL and AVT-NBL

For the benchmark data sets, we chose seven data sets from UCI data repository [1].

The first data set is AGARICUS-LEPIOTA mushroom having 8,124 instances with 22 attributes and two class labels. The second data set is DERMATOLOGY that has 366 instances with 34 attributes and 6 class labels. And the rest data sets are NURSERY, AUDIOLOGY, CAR EVALUATION, SOYBEAN, and ZOO (results not shown in this paper) that showed similar results.

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 the 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

Figure 4 shows one example of AVT generated by AVT-Learner.

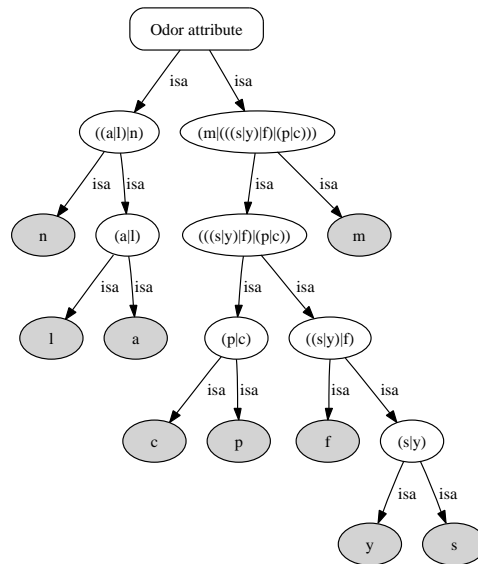


Fig. 4. AVT for the 'odor' attribute of UCI AGARICUS-LEPIOTA mushroom data set generated by AVT-Learner using Jensen-Shannon divergence.

In terms of accuracy and compactness, 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. Human-generated AVT usually is easier to comprehend, however in many domains, human-supplied AVT is unavailable because generation of AVT by human is not a simple task.

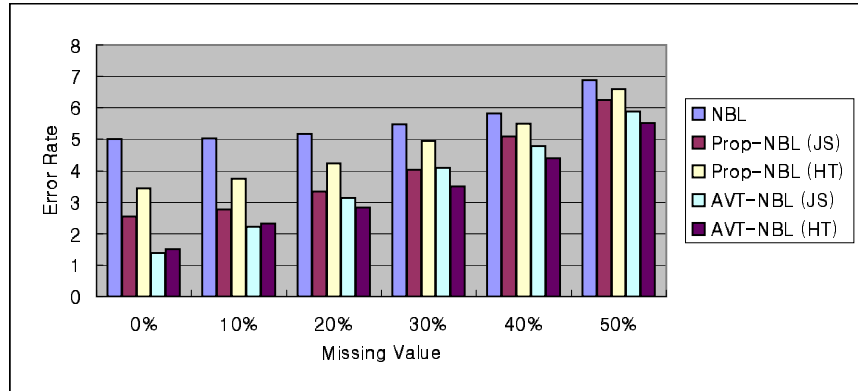


Fig. 5. The estimated error rates of classifiers generated by NBL, PROP-NBL and AVT-NBL on AGARICUS-LEPIOTA data with different percentages of missing values. HT stands for human-supplied AVT. JS denotes AVT constructed by AVT-Learner using Jensen-Shannon divergence.

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

In the case of DERMATOLOGY data sets, there are no human-supplied AVT's available.

Figures 6 show 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 original and propositionalized versions of 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 substantially more accurate than those generated by NBL (with and without propositionalization). 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

For all seven data sets, Naive Bayes classifiers constructed by AVT-NBL generally have smaller number of parameters than those from PROP-NBL or NBL (See Figures 7 for representative results). This suggests 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.

It is worth noting that NBL applied to the data set in which the original set of features are augmented by a set of additional features obtained through propositionalization based on an AVT performs poorly both with respect to accuracy and size (as measured by the number of parameters needed to describe the classifier) relative to NBL applied to the original data set. This may be explained by the fact that the binary attributes introduced by propositionalization using an AVT are *not* independent of the binary attributes that correspond to the original set of attribute values (given class) thereby violating one of the key assumptions of the Naive Bayes Classifier.

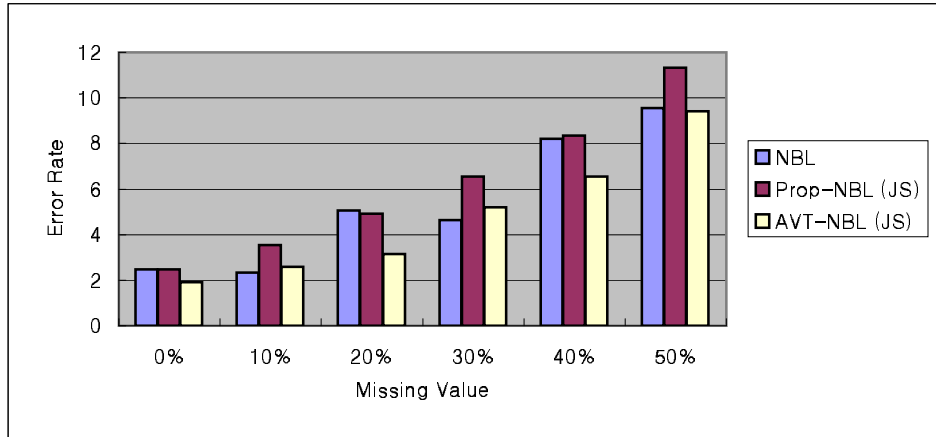


Fig. 6. The error rate estimates of the Standard Naive Bayes Learner (NBL) compared with that of AVT-NBL and Prop-NBL on DERMATOLOGY data. JS denotes AVT constructed by AVT-Learner using Jensen-Shannon divergence.

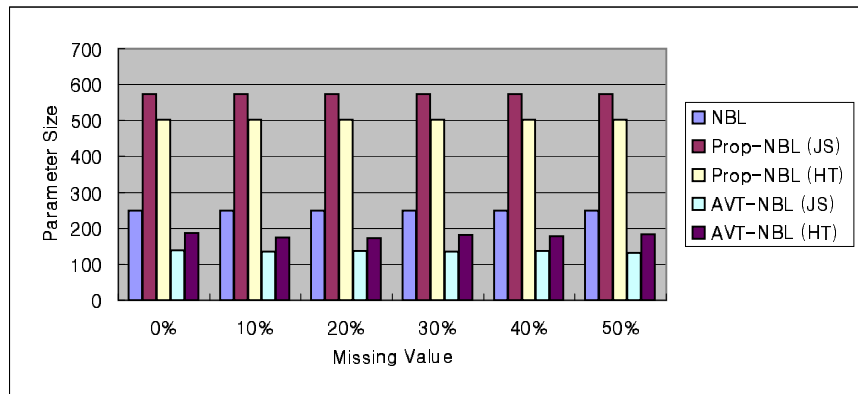


Fig. 7. The size (as measured by the number of parameters) of the Standard Naive Bayes Learner (NBL) compared with that of AVT-NBL and Prop-NBL on AGARICUS-LEPIOTA data. JS denotes AVT constructed by AVT-Learner using Jensen-Shannon divergence.

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. Previous work has shown that attribute value taxonomies can be exploited to generate such classifiers from data [4, 2]. However, human-generated AVTs are unavailable in many application domains. Manual construction of AVTs requires a great deal of domain expertise, and in the 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.

The experiments reported in this paper show that:

- AVT-Learner is effective in generating AVT 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 AVT).
- The AVT generated by AVT-Learner are competitive with human supplied AVT (in the case of benchmark data sets where human-generated AVT were available) in terms of both the error rate and size of the resulting classifiers.

5.2 Related work

Cimiano et. al [16, 17] used agglomerative clustering for learning taxonomies from text. Gibson and Kleinberg [18] introduced STIRR, an iterative algorithm based on non-linear dynamic systems for clustering categorical attributes. Ganti et. al. [19] designed CACTUS, an algorithm that uses intra-attribute summaries to cluster attribute values. However, both of them didn't make taxonomies and use the generated for improving classification tasks. Pereira et. al. [10] describe distributional clustering for grouping words based on class distributions associated with the words in text classification. Yamazaki et al., [12] describe an algorithm for extracting hierarchical groupings from rules learned by FOCL (an inductive learning algorithm) [20] and report improved performance on learning translation rules from examples in a natural language processing task. Slonim and Tishby [11, 21] describe a technique (called the agglomerative information bottleneck method) which extends the distributional clustering approach described by Pereira et al. [10], 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 [22] report 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 [11].

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.

5.3 Future work

Some directions for future work include:

- Extending AVT-Learner described in this paper to learn AVTs that correspond to hierarchies of intervals to handle numerical attribute values, ordered generalization hierarchies where there is an ordering relation among nodes at a given level in a hierarchy e.g., hierarchies defined over education levels, temporally ordered events such as those captured by system logs in the intrusion detection application, 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 [23], learning classifiers from relational data [24], and protein function classification [25] and identification of protein-protein interfaces [26].
- Development of algorithms that learn hierarchical groupings of values associated with more than one attribute.

6 Acknowledgements

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. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
2. Zhang, J., Honavar, V.: Learning naive bayes classifiers from attribute value taxonomies and partially specified data. In: International Conference on Intelligent System Design and Applications (ISDA 2004). (2004) To appear.
3. Pazzani, M.J., Mani, S., Shankle, W.R.: Beyond concise and colorful: Learning intelligible rules. In: Knowledge Discovery and Data Mining. (1997) 235–238
4. Zhang, J., Honavar, V.: 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)
5. Han, J., Fu, Y.: Exploration of the power of attribute-oriented induction in data mining. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds.: Advances in Knowledge Discovery and Data Mining. AIII Press/MIT Press (1996)
6. Hendler, J., Stoffel, K., Taylor, M.: Advances in high performance knowledge representation. Technical Report CS-TR-3672, University of Maryland Institute for Advanced Computer Studies Dept. of Computer Science (1996)

7. Taylor, M., Stoffel, K., Hendler, J.: Ontology based induction of high level classification rules. In: SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. (1997)
8. Zhang, J., Silvescu, A., Honavar, V.: 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)
9. Haussler, D.: Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial intelligence* **36** (1988) 177 – 221
10. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. In: 31st Annual Meeting of the ACL. (1993) 183–190
11. Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: NIPS-12. (1999)
12. Yamazaki, T., Pazzani, M.J., Merz, C.J.: Learning hierarchies from ambiguous natural language data. In: International Conference on Machine Learning. (1995) 575–583
13. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc. (1993)
14. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: National Conference on Artificial Intelligence. (1992) 223–228
15. Mitchell, T.M.: *Machine Learning*. McGraw-Hill, New York (1997)
16. Cimiano, P., Staab, S., Tane, J.: Automatic acquisition of taxonomies from text: Fca meets nlp. In: Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining, Cavtat–Dubrovnik, Croatia. (2003) 10–17
17. Cimiano, P., Hotho, A., Staab, S.: Comparing conceptual, partitional and agglomerative clustering for learning taxonomies from text. In: Proceedings of the European Conference on Artificial Intelligence (ECAI'04). (2004)
18. Gibson, D., Kleinberg, J.M., Raghavan, P.: Clustering categorical data: An approach based on dynamical systems. *VLDB Journal: Very Large Data Bases* **8** (2000) 222–236
19. Ganti, V., Gehrke, J., Ramakrishnan, R.: Cactus - clustering categorical data using summaries. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press (1999) 73–83
20. Pazzani, M., Kibler, D.: The role of prior knowledge in inductive learning. *Machine Learning* **9** (1992) 54–97
21. Slonim, N., Tishby, N.: 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, ACM Press (2000) 208–215
22. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (1998) 96–103
23. Helmer, G., Wong, J.S.K., Honavar, V.G., Miller, L.: Automated discovery of concise predictive rules for intrusion detection. *J. Syst. Softw.* **60** (2002) 165–175
24. Atramentov, A., Leiva, H., Honavar, V.: A multi-relational decision tree learning algorithm - implementation and experiments. In Horváth, T., Yamamoto, A., eds.: ILP03. Volume 2835 of LNAI., Springer-Verlag (2003) 38–56
25. Wang, X., Schroeder, D., Dobbs, D., Honavar, V.G.: Automated data-driven discovery of motif-based protein function classifiers. *Inf. Sci.* **155** (2003) 1–18
26. Yan, C., Dobbs, D., Honavar, V.: Identification of surface residues involved in protein-protein interaction – a support vector machine approach. In Abraham, A., Franke, K., Koppen, M., eds.: *Intelligent Systems Design and Applications (ISDA-03)*, Springer-Verlag (2003) 53–62