Attribute Value Ontology
Using Semantics in Data Mining

Tomasz Łukaszewski, Joanna Józefowska and Agnieszka Ławrynowicz
Poznan University of Technology, Poznan, Poland

Keywords: Ontologies, Imperfect Data, Naïve Bayesian Classifier.

Abstract: We propose a new concept to represent attribute values as an ontology that allows modeling different levels of abstraction. In this way more or less precise values may be used instead of missing or erroneous data. The goal is to use this representation in order to improve analysis of imperfect data. The proposed attribute value ontology (AVO) allows to upgrade the precision of information not only from positive observations but also from negative ones. We show how to classify a new example using a set of training examples described in the same or more precise way. Another advantage of the proposed approach is providing an efficient way to avoid the effect of overfitting.

1 INTRODUCTION

The concept of using ontologies in order to enhance data with semantics resulted in a new vision of computing - semantic computing. On the one hand, semantic computing allows to integrate heterogeneous data sources. On the other hand, semantic computing allows to improve the data analysis e.g. the analysis of imperfect data.

Imperfect data (e.g. erroneous or missing attribute values) are very common in the field of Data Mining and they have a negative effect on the mining results. Let us notice, that some erroneous or missing attribute values may be introduced by users that are required to provide very specific values, but the level of their knowledge of the domain is only very general and they are unable to precisely describe the observation by an appropriate value of an attribute. Even if a person is an expert in the domain, erroneous or missing attribute values can be introduced as a consequence of lack of time or other resources to precisely describe the observation by an appropriate value.

In this paper we present a semantic approach to *Data Mining* (Witten et al., 2011) aiming at improvement of the analysis of imperfect data using some background knowledge. Most common approach to exploit background knowledge in data mining is generalization of attribute values (Han et al., 1992). It allows to define *abstract* concepts as generalizations of the *primitive* ones. Background knowledge used in this approach has a form of taxonomies, categories or more general relationships between attributes.

We introduce an *attribute value ontology* (AVO) in order to improve the expressiveness of the knowledge representation language. Firstly, AVO allows to model different *levels of abstraction - precise* and *imprecise* values. As a result, users are allowed to use (less or more) imprecise values instead of erroneous or missing values. Secondly, AVO allows to *precise* values not only by indicating the subconcept of a current concept (*a positive observation*) but also rejecting some subconcepts of the current concept (*a negative observation*). This is a common technique of describing what something is by explaining what it is not. Moreover, AVO allows to avoid *overfitting* by analysing results not only for a current value but also for less precise values. This approach is an analogy to the analysis of results for special cases (exceptions) and for general ones. Finally, we present a naïve Bayesian classifier extended to AVO.

2 ATTRIBUTE VALUE ONTOLOGY

Let us assume that given is an ontology, which represents the domain knowledge. In particular, it expresses a multilevel subsumption hierarchy of concepts (ISA hierarchy) representing different levels of abstraction - precise and imprecise values. We define an *attribute value ontology* (AVO) as follows:

Łukaszewski T., Józefowska J. and Ławrynowicz A..
Attribute Value Ontology - Using Semantics in Data Mining.
DOI: 10.5220/0004157003290334
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (SCOE-2012), pages 329-334
ISBN: 978-989-8565-11-2

Definition 1. An *attribute value ontology* (AVO) is a pair $\mathcal{A} = \langle C, R \rangle$, where: C is a set of concepts (*primitive* and *abstract* ones), R is a subsumption relation over C, subset $C^P \subseteq C$ of concepts without predecessors is a finite set of primitive concepts of $\mathcal{A}$.

For simplicity of presentation we consider the hierarchy of concepts such that each concept has at most one predecessor (tree structure). In general concepts may have multiple predecessors (DAG structure).

2.1 Levels of Abstraction

Given is an attribute A and a set $V = \{v_1, v_2, ..., v_n\}$, $n > 1$, of *specific* values of this attribute. These specific values can be interpreted as a single level of abstraction. The user that is unable to precisely describe the observation by a specific value of A has to manage with a missing or erroneous value.

Introducing AVO we add new levels of abstraction. We assume that *primitive* concepts of AVO represent these specific values of A (the original level of abstraction). *Abstract* concepts of AVO are used when the users are unable to precisely describe the observation by a specific value of A (the new levels of abstraction). Therefore, we call primitive and abstract concepts: *precise* and *imprecise* values of AVO, respectively.

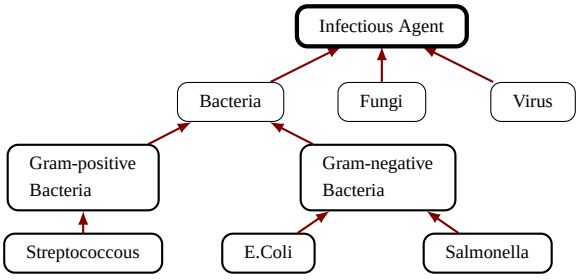

Figure 1: Example of an attribute value ontology.

Example 1. Let us consider the following medical problem. In order to determine the correct treatment, an agent that caused the infection needs to be specified. Although, all viral infections determine the same treatment (similarly infections cased by fungi), identification of the bacteria type is important in order to decide about the appropriate treatment. Thus, specific values of this attribute are the following: *Streptococcus, E.Coli, Salmonella, Fungi, Virus.* An AVO describing the domain of infectious agents is presented in Fig. 1. Primitive concepts of AVO represent these specific values. Abstract concepts are the following: *Infectious Agent, Bacteria, Gram-positive Bacteria, Gram-negative Bacteria*.

A user that is unable to precisely describe the infectious agent can use the most abstract concept of this AVO (*Infectious Agent*) or one of its abstract subconcepts (*Bacteria, Gram-positive Bacteria, Gram-negative Bacteria*). In the next subsection we show that AVO allows to precise values not only by indicating the subconcept of a current concept but also rejecting some subconcepts of the current concept.

Let us observe that *Streptococcus* is not the only *Gram-positive Bacteria* in the real world and our hierarchy, for some reasons, does not contain concepts of the other *Gram-positive Bacteria*. In order to represent this we make the *open world assumption* (OWA). Therefore, the concept *Gram-positive Bacteria* should be correctly interpreted as: *Streptococcus* or other *Gram-positive Bacteria*. Similarly, the concept *Gram-negative Bacteria* should be interpreted as: *E.Coli* or *Salmonella* or other *Gram-negative Bacteria*. In the next subsection we show the consequences of assuming the open world.

2.2 Increasing the Precision

In the previous section we have shown that the user that is unable to precisely describe the observation may use abstract concepts (imprecise values) of AVO. A person that knows *nothing* can use the most abstract concept of AVO. Considering our medical problem, the user shall use the abstract concept *Infectious Agent* - Fig. 1.

AVO allows to *precise* descriptions *explicitly*, indicating the subconcept of a current concept. Considering our medical problem, let us assume, that the user has made a *positive observation* that this infectious agent is *Bacteria* - Fig. 2.

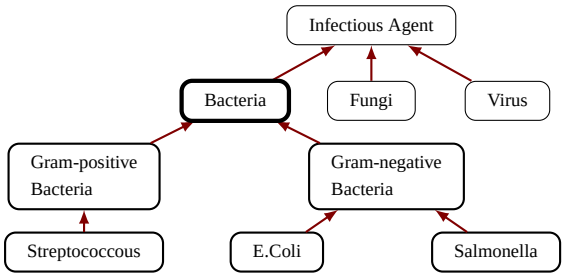

Figure 2: Positive observation: infectious agent is bacteria.

AVO allows to *precise* descriptions also *implicitly*, rejecting some subconcepts of a current concept. This is a common technique of describing what something is by explaining what it is not. Continuing or medical example, let us assume, that the user has made a *negative observation* that this *Bacteria* is not *Gram-negative Bacteria* - Fig. 3.

Let us notice, that making this negative observation, all but one subconcepts of *Bacteria* have been

rejected. Making the closed world assumption (CWA) we would be allowed to say automatically, that *Bacteria* is *Gram-positive Bacteria*. However, we have made the open world assumption (OWA) and we are not allowed to precise the description in this way.

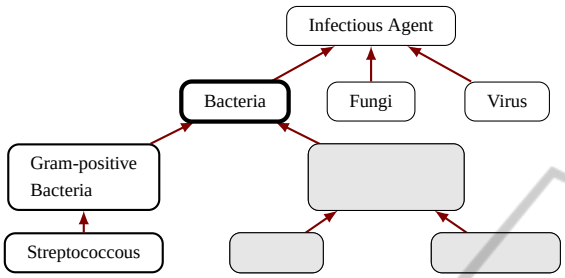

Figure 3: Negative observation: bacteria is not gram-negative bacteria.

2.3 Avoidance of Overfitting

Many Data Mining algorithms construct rules or trees that are *overfitted* to the training data. The best way to avoid overfitting is to simplify the learned model (Witten et al., 2011). This simplification can be reached by generating sensible rules or pruning trees. Both approaches are aimed at rejecting rules that cover too few training examples.

AVO allows to avoid *overfitting* by analysing results not only for a current level of precision but also for less precise values (superconcepts of the current concept). This approach is an analogy to the analysis of results for a special case (an exception) and for the general case.

3 EXTENDING NAÏVE BAYESIAN CLASSIFIER BY AVO

Using AVO we are able to represent training and testing examples with precise and imprecise descriptions. In this section we show how to extend the naïve Bayesian classifier by AVO in order to learn this classifier from precisely and/or imprecisely described examples and classify precisely and/or imprecisely described examples.

3.1 Naïve Bayesian Classifier

The most straightforward and widely tested method for probabilistic induction is known as the naïve Bayesian classifier. Despite its simplicity and the strong conditional independence assumptions it makes, the naïve Bayesian classifier often performs remarkably well, competitively with other well-known induction techniques such as decision trees and neural networks. The naïve Bayesian classifier is often used for classification problems, in which a learner attempts to construct a classifier from a given set T of training examples with class labels.

Assume that given is a set of n attributes $A_1, A_2, ..., A_n$. An (training or testing) example is represented by a vector $(v_1, v_2, ..., v_n)$, where v_i is the specific value of A_i. Let C represent the class variable and C_j represent the value it takes (a class label).

The Bayesian classifier (and also the naïve Bayesian classifier) is a classification method, which classifies a new observation E by selecting the class C_j with the largest posterior probability $P(C_j|E)$, as indicated below:

$$P(C_j|E) = \frac{P(C_j)P(E|C_j)}{P(E)} \quad . \tag{1}$$

$P(E)$ is ignored, since it is the same for all classes, and does not affect the relative values of their probabilities:

$$P(C_j|E) \propto P(C_j)P(E|C_j) \quad . \tag{2}$$

Since E is a composition of n discrete values, one can expand this expression:

$$P(C_j|v_1, v_2, ..., v_n) \propto P(C_j)P(v_1, v_2, ..., v_n|C_j) \quad . \tag{3}$$

where $P(v_1, v_2, ..., v_n|C_j)$ is the conditional probability of the example E given the class C_j; $P(C_j)$ is the prior probability that one will observe class C_j. All these parameters are estimated from the training set. However, a direct application of these rules is difficult due to the lack of sufficient data in the training set to reliably obtain all the probabilities needed by the model. The naïve Bayesian classifier assumes that the attributes are *conditionally independent* given the class variable, which gives us:

$$P(C_j|v_1, v_2, ..., v_n) \propto P(C_j) \prod_i P(v_i|C_j) \quad . \tag{4}$$

$P(v_i|C_j)$ is the probability of an instance of class C_j having the observed attribute A_i value v_i. The probabilities in the above formula must be estimated from training examples, e.g. using relative frequency:

$$P(C_j) = \frac{n_j}{n} \quad P(v_i|C_j) = \frac{n_{ij}}{n_j} \tag{5}$$

where n is the number of training examples, n_j is the number of training examples with class label C_j, n_{ij} is the number of training examples with the value of the attribute $A_i = v_i$ and class label C_j.

3.2 Inference with AVO

In the proposed approach with abstract concepts (with AVO), the naïve Bayesian classifier needs to be generalized to estimate $P(c_i|C_j)$, where c_i is a primitive or an abstract concept of $\mathcal{A}_i$. Let us recall, that for a given concept c_i in $\mathcal{A}_i$, all the concepts that are more specific than the concept c_i are the descendants of this concept c_i. In order to estimate $P(c_i|C_j)$, e.g. by relative frequency, we use the following property:

$$P(c_i|C_j) = \frac{n_{ij} + \sum_{c_k \in desc(c_i)} n_{kj}}{n_j} \quad (6)$$

where n_j is the number of training examples with class label C_j, n_{ij} is the number of training examples with the value of the attribute $\mathcal{A}_i = c_i$ and class label C_j, n_{kj} is the number of training examples with the value of the attribute $\mathcal{A}_i = c_k$ and class label C_j, $desc(c_i)$ is the set of concepts that are descendants of the concept c_i.

The proposed approach is a generalization of the classical approach (without abstract concepts). In the classical approach, specific attribute values can be interpreted as a single level of knowledge granularity, and a new example is classified using training examples described by the same specific attribute value only. In the proposed approach (with abstract concepts) each descendant of a given concept 'is' this concept. Therefore, in the classification of a new example described by a concept c_i we use *also* all training examples described by descendants of c_i.

The algorithm has been implemented and tested on a series of examples available in the literature. The results are promising, although more extensive experiments are necessary in order to present statistically valid conclusions.

3.3 Illustrative Example

Let us consider the medical problem presented in Example 1. In order to determine the correct treatment, an agent that caused the infection needs to be specified. For the simplicity of the presentation we consider only one attribute. The training data is given in Table 1. The first column of the table presents the number of training examples described by a given value (precise or imprecise) of the infectious agent for each class. For example, the imprecise value *Bacteria* is used in order to describe 6 instances with the class label C_1 and 7 instances with the class label C_2. In the considered example each class is described exactly by the same number of instances, therefore the prior probability that one will observe class C_j is equal to 0.5 for C_1 and C_2.

Table 1: A medical diagnosis training data.

Instances	Infectious Agent	Class
6	Bacteria	C1
3	Gram-positive Bacteria	C1
1	Gram-negative Bacteria	C1
7	Bacteria	C2
1	Streptococcus	C2
2	Gram-negative Bacteria	C2

Infectious Agent is not Known. Let us consider the following scenario: there is a patient and the diagnosis is not known. Knowing nothing we are allowed to use the most abstract concept of AVO - *Infectious Agent*. We estimate the posterior probability $P(C_j|InfectiousAgent)$. Therefore, we concentrate on these examples, that are described by the value *Infectious Agent* or its descendants. From (6) we have: $P(InfectiousAgent|C_1) = \frac{0+(6+3+1)}{10} = 1$, $P(InfectiousAgent|C_2) = \frac{0+(7+1+2)}{10} = 1$. From (4) we have: $P(C_1|InfectiousAgent) \propto 0.5 * 1 = 0.5$, and $P(C_2|InfectiousAgent) \propto 0.5 * 1 = 0.5$. As we can see, both classes are equally probable.

Infectious Agent is Bacteria. Let us assume, that the user has made a positive observation that this infectious agent is *Bacteria* - Fig. 2. We estimate the posterior probability $P(C_j|Bacteria)$. Therefore, we concentrate on these examples, that are described by the value *Bacteria* or its descendants. From (6) we have: $P(Bacteria|C_1) = \frac{6+(3+1)}{10} = 1$, $P(Bacteria|C_2) = \frac{7+(1+2)}{10} = 1$. From (4) we have: $P(C_1|Bacteria) \propto 0.5 * 1 = 0.5$, and $P(C_2|Bacteria) \propto 0.5 * 1 = 0.5$. As we can see, both classes are still equally probable.

Bacteria is not Gram-negative Bacteria. Let us assume, that the user has made a negative observation that *Bacteria* is not *Gram-negative Bacteria* (shortly: Bacteria&notGnB) - Fig. 3. We estimate the posterior probability $P(C_j|Bacteria\&notGnB)$. Therefore we again concentrate on these examples, that are described by the value *Bacteria* or its descendants. However, this time we do not take into the consideration these training examples, that are described by the concept *Gram-negative Bacteria* and its subconcepts. From (6) we have: $P(Bacteria\&notGnB|C_1) = \frac{6+(3)}{10} = 0.9$, $P(Bacteria\&notGnB|C_2) = \frac{7+(1)}{10} = 0.8$. From (4) we have: $P(C_1|Bacteria\&notGnB) \propto 0.5 * 0.9 = 0.45$, and $P(C_2|Bacteria\&notGnB) \propto 0.5 * 0.8 = 0.4$. As we can see, there is a small difference between the probabilities of these classes. Therefore, additional observations are recommended.

Avoidance of Overfitting. Let us assume, that the user has made a positive observation that *Bac-*

teria is *Streptococcus*. We estimate the posterior probability $P(C_j|Streptococcus)$. Therefore we concentrate on these examples, that are described by the value *Streptococcus* or its descendants. From (6) we have: $P(Streptococcus|C_1) = \frac{0+(0)}{10} = 0$, $P(Streptococcus|C_2) = \frac{1+(0)}{10} = 0.1$. From (4) we have: $P(C_1|Streptococcus) \propto 0.5 * 0 = 0$, and $P(C_2|Streptococcus) \propto 0.5 * 0.1 = 0.05$.

The result indicates that only class $C2$ is probable. However, this result is based on a very small set of training examples. Therefore, this result should be treated rather as the exception from a general rule. In our example the general rule indicates that both classes are equally probable. However, awareness of this exception may be very valuable in some cases.

4 RELATED WORK

Data Mining with background knowledge has been extensively studied in the past. One of the aspects of the background knowledge are relations between the attribute values. Generalization of attribute values is the simplest relation considered in this context. It allows to get *abstract* concepts as generalizations of the *primitive* ones. Background knowledge used in this approach has a form of taxonomies, categories or more general relationships between concepts. Abstract concepts are used in the data mining tasks in various ways.

Compactness and Generality of Results. In the early approaches generalization was carried out in order to get more compact and more general data mining results. Two groups of methods may be distinguished in this line. The first group consists of methods where abstract concepts replace the data values in the original database before applying the core data mining algorithm. This approach is used, for example, in: (Walker, 1980), (Han et al., 1992) and (Kudoh et al., 2003). In the methods of the second group generalization is integrated with the data mining algorithm. Among others, this approach was applied in: (Núñez, 1991), (Almuallim et al., 1996), (Tanaka, 1996), (Taylor et al., 1997). In (Núñez, 1991) an algorithm EG2 (Economic Generalizer 2) was proposed to build a decision tree. The background knowledge contains ISA hierarchies of attribute values. At each node of the decision tree, this algorithm builds a union of abstract values and primitive values. In (Almuallim et al., 1996) an algorithm was proposed to find a multiple-split test on hierarchical attributes (ISA hierarchies) in decision tree learning. The proposed multiple-split test is a *cut* through a hierarchy, which maximizes the gain-ratio measure. The idea of *cut* was proposed in (Haussler, 1988). The cut through a hierarchy allows to reduce multiple levels of abstraction to a single level of abstraction and apply classical algorithms. However, the number of possible cuts (split tests) grows exponentially in the function of leaves of the hierarchy. It turns out that this task is very similar to the task of decision tree pruning and this allows to employ a decision tree pruning technique introduced in (Breiman et al., 1984). In (Tanaka, 1996) a very similar approach was proposed to build decision trees using structured attributes (ISA hierarchies) and called LASA (LearniAVT-DTng Algorithm with Structured Attributes). This approach defines the *unique and complete cover node set* which corresponds to the *cut* through a hierarchy. A measure of *generalization goodness* was proposed, which takes into account two mutually conflicting factors: a generalization level and a penalty for the induced errors. An algorithm to find optimum generalization, that transforms the original problem to the shortest path problem, was also proposed. A simple experiment showed, that the classification results of the proposed approach are better than a standard approach in terms of classification accuracy. (Taylor et al., 1997) applied a tool ParkaDB to integrate databases and ontologies in order to generate classification rules based on generalized concepts from an ontology. The level of the generalization is determined by gathering frequency counts and evaluating so called strong indicators for class membership.

Handling Imprecise Descriptions. A more recent approach is to use abstract concepts (imprecise values) in order to represent real objects that can not be precisely described by the available specific values of an attribute. The use of *attribute value taxonomies* (AVT) in the decision tree learning (AVT-DTL) and the naïve Bayesian classifier (AVT-NBL) is presented respectively in (Zhang et al., 2002) and (Zhang et al., 2006). AVT-DTL and AVT-NBL, to the best of our knowledge, are the only one existing approaches for learning classifiers from imprecisely specified instances and classifying imprecisely specified instances. AVT-DTL and AVT-NBL use a *cut* through a hierarchy of concepts. When instances are described by abstract values *below* the cut through a taxonomy, they are aggregated upwards and stored in abstract values of the cut. When they are described by abstract values *above* the cut, they are propagated to their descendants in the cut, proportionally.

Our approach (naïve Bayesian classifier extended to AVO) is an improvement of the AVT-DTL and AVT-NBL in three directions. Firstly, AVT-DTL and AVT-NBL use a *cut* through a taxonomy. This *cut* reduces the considered taxonomy to a single level of abstrac-

tion only. This reduction was necessary in order to apply a classical algorithm of decision tree learning that was designed for a single level of abstraction. The use of this *cut* in AVT-NBL is just a tradeoff between the complexity and accuracy of the classifier (Zhang et al., 2006). We show that naïve Bayesian classifier can be extended to AVO without any reduction of a given ontology. Secondly, the semantics of AVO allows to precise descriptions explicitly (*positive observations*) and implicitly (*negative observations*). Thirdly, the semantics of AVO allows to avoid overfitting in a very effective way by analyzing results for less precise values. The overfitting avoidance is a very important issue in Data Mining and is very important from the practical point of view.

5 CONCLUSIONS

In this paper we proposed an extension of the naïve Bayesian classifier by an attribute value ontology (AVO) aiming at the improvement of the analysis of imperfect data. In the proposed approach, every attribute is a hierarchy of concepts from the domain knowledge base (ISA hierarchy). This semantic approach to Data Mining allows to describe examples either very precisely or, when it is not possible, in a more general way (using a concept from higher levels of the hierarchy). As a result, users that are unable to precisely describe the observation by a specific value of an attribute, are allowed to use (less or more) imprecise values.

Let us notice, that each imprecise value of AVO, except the most abstract concept, is *more* precise than the *missing value*, represented by this most abstract concept. Therefore, introducing these abstract concepts we improve the analysis of imperfect data. This improvement is increased by each upgrade of the precision of information. We showed that even negative observations improve this precision: ”knowing what we do not know” is already information.

We could ask a question: how far should we precise the description? There is no single answer for this question. Each *cut* through a hierarchy seems to be a *tradeoff* between the complexity and accuracy. Therefore, maintaning all the levels of abstraction is an alternative approach to this problem. It allows to compare results for *many* levels of abstraction. That can be an efficient way to avoid the effect of overfitting. Moreover, this comparison can be utilized by a cost sensitive computing. High precision carries a high cost. The challenge is to exploit the tolerance for imprecision. Further research aims at experimental evaluation of the proposed approach.

ACKNOWLEDGEMENTS

Research supported by the Polish Ministry of Science and Higher Education grant No. N N516 186437.

REFERENCES

Almuallim, H., Akiba, Y., and Kaneda, S. (1996). An efficient algorithm for finding optimal gain-ratio multiple-split tests on hierarchical attributes in decision tree learning. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. AAAI Press.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, California, 3rd edition.

Han, J., Cai, Y., and Cercone, N. (1992). Knowledge discovery in databases: An attribute-oriented approach. In *Proceedings of the 18th International Conference on Very Large Data Bases*. Morgan Kaufmann.

Haussler, D. (1988). Quantifying inductive bias: Ai learning algorithms and valiant’s learning framework. In *Artif. Intell., Vol. 36(2)*. Elsevier.

Kudoh, Y., Haraguchi, M., and Okubo, Y. (2003). Data abstractions for decision tree induction. In *Theoretical Computer Science, Vol. 292(1)*. Elsevier.

Núñez, M. (1991). The use of background knowledge in decision tree induction. In *Machine Learning, Vol. 6(3)*. Springer.

Tanaka, H. (1996). Decision tree learning algorithm with structured attributes: Application to verbal case frame acquisition. In *Proceedings of the 16th International Conference on Computational Linguistics*. Center for Sprogteknologi, Danmark.

Taylor, M. G., Stoffel, K., and Hendler, J. A. (1997). Ontology-based induction of high level classification rules. In *Proceedings of the Workshop on Research Issues on Data Mining and Knowledge Discovery*. ACM.

Walker, A. (1980). On retrieval from a small version of a large data base. In *Proceedings of the Sixth International Conference on Very Large Data Bases*. IEEE Computer Society.

Witten, I., Frank, E., and Hall, M. (2011). *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Burlington, 3rd edition.

Zhang, J., Kang, D., Silvescu, A., and Honavar, V. (2006). Learning accurate and concise naive bayes classifiers from attribute value taxonomies and data. In *Knowl. Inf. Syst., Vol. 9(2)*. Springer.

Zhang, J., Silvescu, A., and Honavar, V. (2002). Ontology-driven induction of decision trees at multiple levels of abstraction. In *LNCS Vol. 2371*. Springer.