Control and Cybernetics

vol. **40** (2011) No. 2

Handling the description noise using an attribute value ontology*

by

Tomasz Łukaszewski, Joanna Józefowska, Agnieszka Ławrynowicz and Łukasz Józefowski

Institute of Computing Science, Poznan University of Technology
ul. Piotrowo 2, 60-965 Poznan, Poland

Abstract: The quality of any classifier depends on a number of factors, including the quality of training data. In real-world scenarios, data are often noisy. One reason for noisy data (erroneous values) is in the representation language, insufficient to model different levels of knowledge granularity. In this paper, to address the problem of such description noise, we propose a novel extension of the naïve Bayesian classifier by an attribute value ontology (AVO). In the proposed approach, every attribute is a hierarchy of concepts from the domain knowledge base. In this way an example is described either very precisely (using a concept from the low-level of the hierarchy) or, when it is not possible, in a more general way (using a concept from higher levels of the hierarchy). Our general strategy is to classify a new example using training examples described in the same way or more precisely at lower levels of knowledge granularity. Hence, the hierarchy introduces a bias which in effect can contribute to improvement of a classification.

Keywords: imprecise descriptions, attribute noise, ontology, naïve Bayesian classifier.

1. Introduction

The ability to *learn* is a very important feature of intelligent behavior, so any attempt to understand intelligence has to include the understanding of learning. A dictionary definition of learning includes phrases such as "to obtain knowledge of facts, or of how to do things, or an understanding of ideas". Obviously, the ability to learn is considered as a feature of any intelligent system and as such belongs to the most challenging research areas in artificial intelligence. Learning that can be realised with machines to improve their performance is usually referred to as 'machine learning'. In this paper we concentrate on supervised

*Submitted: October 2010; Accepted: June 2011.

learning which consists in inferring a model from supervised training data. This model should predict the correct output value for any valid input data. This requires the learning algorithm to generalize from the training data (training examples) to previously unobserved situations (testing examples). Training and testing examples are described in a representation language. The *description* often has the form of a simple attribute-value formalism. Moreover, the class label is assigned to each training example.

One of the tasks in supervised machine learning is *classification*. Classification is the process of predicting categorical (discrete, unordered) classes (class labels) and the derived model may be represented, e.g. in the form of rules, decision trees, mathematical formulae or neural networks. Moreover, the classification methods that can be used include: Bayesian classifier, support vector machines or k-nearest neighbours algorithms (Han and Kamber, 2006). A classical example of the classification task is to derive a medical diagnosis from symptoms, where diseases stand for class labels.

The *predictive accuracy* of classifiers (the accuracy of classifying objects other than those in the training set) is determined by the following factors (Zhu and Wu, 2004): inductive bias of the learning algorithm and quality of training data. Given a specific learning algorithm it is obvious that its predictive accuracy depends on the quality of training data. The quality of training data is determined by two factors: internal (the selection of attributes and the class) and external (errors introduced into a dataset) (Zhu and Wu, 2004). Two types of errors occurring in a dataset are distinguished (Quinlan, 1986): erroneous attribute values and misclassified objects. Non-systematic errors of these kinds are usually called *noise* (Quinlan, 1986). Consequently, noise can decrease the classifier performance in terms of the predictive accuracy of a classifier, time of building a classifier and the size of a classifier (Zhu and Wu, 2004).

In order to increase the predictive accuracy of a classifier many possible solutions for data preprocessing or noise handling have been proposed. For example, predictive accuracy can be enhanced by the following procedures: elimination of noisy examples, predicting unknown or missing attribute values or correcting attribute or class values. Although all these approaches seem very different, they try to somehow 'clean' this noisy training data.

In this paper, we propose an approach that 'prevents' from introducing noise. Firstly, we show how to handle (prevent from introducing) some type of noise (we call it *description noise*) in order to reduce the amount of erroneous or missing attribute values. To that end, we use levels of knowledge granularity modeled by an attribute value ontology (AVO). This AVO is used to handle descriptions of training and testing examples. Secondly, we demonstrate how to use AVO in the naïve Bayesian classifier. The general strategy is to *classify a new example using training examples described in the same way or more precisely at lower levels of knowledge granularity*. Our strategy is the generalization of a classical approach, where a new example is classified using training examples described by the same specific attribute value.

The main contributions of the paper are the following:

- introduction of the concept of description noise,
- a method of description noise handling by AVO,
- an extension of the naïve Bayesian classifier by AVO.

The rest of the paper is organized as follows. In Section 2 we shortly present some known approaches to the generalization of attribute values in data mining. In Section 3 the problem of noise in machine learning is discussed. In Section 4 we introduce the concept of description noise. In Section 5 we define the attribute value ontology and propose a method of description noise handling using this ontology. In Section 6 we propose a novel extension of the naïve Bayesian classifier by AVO. In Section 7 we conclude and show the possibilities of further work.

2. Generalization of attribute values

Data mining with background knowledge has been extensively studied in the past. One of the aspects of background knowledge are relations between the attribute values. *Generalization of attribute values* is the simplest relation considered in this context. It allows for obtaining *abstract* concepts as generalizations of the *primitive* ones. Background knowledge used in this approach has the 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 along 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. In particular, 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 use concepts at multiple levels of generalization. The number of possible cuts (split tests) grows exponentially in the number of leaves of the hierarchy. However, 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), called LASA (Learning Algorithm with Structured Attributes). This approach defines the *unique and complete cover node set* which corresponds to the *cut* through the 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 the so called strong indicators for class membership.

Handling imprecise descriptions A more recent approach is to use abstract attribute values in order to represent real objects that cannot be precisely described by the available primitive values. The use of taxonomies (*Attribute Value Taxonomies*) in the decision tree learning (AVT-DTL) and the naïve Bayesian classifier (AVT-NBL) is presented, respectively in Zhang et al. (2002, 2006). AVT-DTL and AVT-NBL, to the best of our knowledge, are the only existing approaches for learning classifiers from imprecisely described instances and classifying imprecisely described instances.

AVT-DTL and AVT-NBL use a 'cut' through a hierarchy of concepts. When training instances have abstract values 'below' the cut through a taxonomy, their class counts are aggregated upwards and stored in abstract values of the cut. When training examples are 'above' the cut through a taxonomy, their class counts have to be propagated to their descendants in the cut, proportionally.

Our approach (AVO) is an improvement over the AVT-NBL in three directions. Firstly, AVT-NBL uses a 'cut' through a taxonomy. The use of such a 'cut' was required in AVT-DTL in order to define split tests. However, this 'cut' is not required in the naïve Bayesian classifier. The use of this 'cut' in AVT-NBL is a tradeoff between the complexity and accuracy of the classifier Zhang et al. (2006). Secondly, the use of taxonomies may not allow to represent all the necessary abstract concepts. This problem is discussed in the paper. We show that AVO enables using any abstract concept. Thirdly, the semantics of AVO allows to classify a new example not only by 'positive' observations, but also using 'negative' observations. All these improvements can contribute to improvement of a classification.

3. Noise in machine learning

According to Hickey (1996) there are three major sources of noise:

- insufficiency of the description of the training examples,

- corruption of attribute values in the training examples,
- erroneous classification of the training examples.

Consequently, the definition of noise proposed in Hickey (1996) is the following: in learning from examples, *noise* is "anything which obscures the relationship between description and class". However, for real-world data it is difficult to quantitatively characterize the sufficiency of the description of examples. In consequence, only the latter two sources of noise are usually considered and so the following two types of noise are distinguished: *attribute noise* and *class noise* (Zhu and Wu, 2004).

Noise in databases occurs in various forms, for example (Wu, 1995):

- Erroneous attribute values. Some data in the training set are distorted for some reasons.
- Missing attribute values or *Don't Know* values. A *Don't Know* value may take any value in its attribute domain.
- Incomplete attributes. When the discriminant attributes are not available, a learning algorithm must use other features that may not be sufficient.
- *Don't Care* values. *Don't Care* values should not be viewed as noise. However, if an example with such a value is converted into a number of equivalent examples that have no *Don't Care* value, and the expanded examples contradict other examples in the training set, we say that *Don't Care* values generate noise.
- Misclassifications. An example is labeled with a wrong class label.
- Contradictory examples. The same example appears more than once in the training data and is labeled with different classifications at different times.
- Uneven distribution of training examples in the example space.
- Redundant data. In addition to increasing the computational complexity, redundant data may become contradictory examples if multiple copies of the same example are assigned to different classes.

There are several possible solutions for dealing with the existence of noise. The noise handling process can be carried out at different stages of inductive learning and can be classified as follows (Wu, 1995):

- preprocessing of the training examples,
- handling the noise during the induction process,
- postprocessing of the results.

Detailed description of the noise handling approaches is out of the scope of the paper. An extensive survey of such methods can be found, for example, in Zhu and Wu (2004).

4. Description noise

The first source of noise, i.e. insufficiency of the description of the training examples, although not easily quantifiable, is important. We also consider insufficiency of the description of the testing examples. In this section we introduce the concept of the description noise. Further, we propose an approach to handle this type of noise.

4.1. The concept of description noise

Let us call the noise following from insufficient description of the training and testing examples *description noise*. Observe that it may affect the attribute values as well as the class labels. In the paper we consider the problem of attribute values only.

Following Clark and Niblett (1987) the main source for description noise may be the language used to represent the attribute values, which is not expressive enough to model different *levels of knowledge granularity*. In such a case, erroneous attribute values and 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 too general to precisely describe the observation by the appropriate value of an attribute. Even if the person is an expert in the domain, erroneous or missing attribute values can be observed as a consequence of lack of time or other resources necessary to make detailed observations (i.e. a more specific description).

Observe that if the language enabled modeling different levels of knowledge granularity (*precise descriptions* and *imprecise descriptions*), we would be able to reduce the number of erroneous or missing attribute values.

4.2. Handling the description noise

In our work we propose to handle the description noise of a given attribute by introducing the levels of knowledge granularity. The levels of knowledge granularity should reflect the domain knowledge and can not be constructed arbitrarily. Let us notice that in some domains, hierarchical relationships between *concepts* may be observed and this knowledge could be explored. Such knowledge is often available in the form of *ontologies*. Thus, the precise and imprecise descriptions for a given attribute A, are represented by a hierarchical structure, called the *Attribute Value Ontology* (AVO). We assume that precise descriptions (specific values) are represented by *primitive* concepts while imprecise descriptions are represented by *abstract* concepts.

Before we formally define the attribute value ontology, we show a demonstrative example of a hierarchy of concepts.

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 spe-*

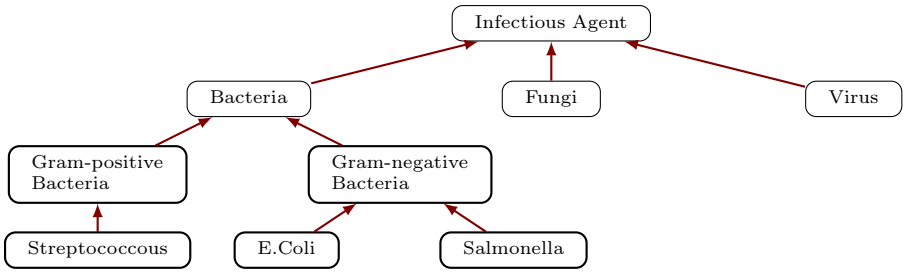

Figure 1. Example of an attribute value ontology

cified. 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. Description noise is handled by a hierarchy of primitive and abstract concepts. Primitive concepts are the following:* Streptococcus, E.Coli, Salmonella, Fungi, Virus*. Abstract concepts are the following:* Infectious Agent, Bacteria, Gram-positive Bacteria, Gram-negative Bacteria*.*

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 other Gram-positive Bacteria. Therefore, the concept of Gram-positive Bacteria should be correctly interpreted as: Streptococcus or other of Gram-positive Bacteria. Similarly, the concept of Gram-negative Bacteria should be interpreted as: E.Coli or Salmonella or other Gram-negative Bacteria. It is easy to observe that we cannot represent any Gram-positive Bacteria other than Streptococcus using the specific values only. This example illustrates the fact that a language containing the specific values only suffers from the description noise.

5. Attribute value ontology in description noise handling

To handle the description noise we use ontology. In this section we formally define the attribute value ontology (AVO) and the partitioning attribute value ontology (PAVO). Further, we show how precise and imprecise descriptions are associated to concepts of PAVO.

5.1. Attribute value ontology

Given is an attribute A and set $V = \{v_1, v_2, ..., v_n\}$, $n > 1$, of specific values of this attribute. Let us assume also that given is an ontology, which represents domain knowledge. In particular, it expresses a multilevel subsumption (”is-a”) hierarchy of concepts representing the precise and imprecise descriptions. We define an attribute value ontology (AVO) $\mathcal{A}$ as follows:

DEFINITION 1 *An Attribute Value Ontology (AVO) $\mathcal{A}$ is a directed acyclic graph $\langle C, R\rangle$, 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 $\mathcal{A}$.*

Further in this paper we use an AVO with the following properties: each concept $c_i \in C$ represents a *non-empty subset* of V, and a hierarchy of concepts represents a *hierarchical partitioning* of set V. We call an AVO with the above properties a *partitioning attribute value ontology* (PAVO).

DEFINITION 2 *A Partitioning Attribute Value Ontology (PAVO) $\mathcal{P}$ is an attribute value ontology $\mathcal{A}$, such that:*

- *$\mathcal{P}$ is a tree.*
- *In the set of concepts we distinguish a root, a set C^P of primitive concepts and a set $C^A = C \setminus (C^P \cup root)$ of abstract concepts.*
- *The root represents set V.*
- *Each primitive concept $c_i \in C^P$ represents a value $v_i \in V$.*
- *Each abstract concept $c_i \in C^A$ represents a proper, non-empty subset V_i of set V.*
- *For each concept $c_k \in (\{root\} \cup C^A)$ all its children are pairwise disjoint and c_k is a union of its children (hierarchical partitioning of set V).*
- *For each pair of concepts $(c_i, c_j) \in R$ we have $V_i \neq V_j$.*

Let us notice that the definition of PAVO allows to form different hierarchies that are a hierarchical partitioning of a given set V. For example, for the set $V=\{v_1, v_2, v_3, v_4, v_5\}$ two (but not all) examples of PAVO are presented in Figs. 2 and 3. Each concept is labeled with the set $V_i \subseteq V$ represented by this concept. PAVO without abstract concepts is called a *flat PAVO*. A PAVO with abstract concepts is called a *complex PAVO*.

5.2. Association of descriptions to concepts of PAVO

We assume that each training and testing example is described by a non-empty set Z_l such that $Z_l \subseteq V$ and V is the set of specific values of an attribute A. For $|Z_l| = 1$ a description is the *precise description* and for $|Z_l| > 1$ a description is the *imprecise description*. Moreover, we assume that the association of a description to a concept c_i changes the semantics of the original description into a set $V_i \subseteq V$ represented by c_i.

DEFINITION 3 *A* corresponding *concept c_i for a given description Z_l is a concept $c_i \in C$ such that $Z_l = c_i$.*

We can observe that all the precise descriptions have corresponding concepts in each PAVO. However, not the all imprecise descriptions have corresponding concepts in a given PAVO. Let us consider the PAVO presented in

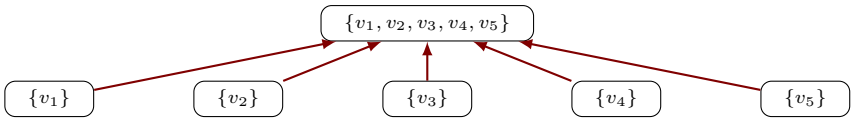

Figure 2. Example of a flat PAVO

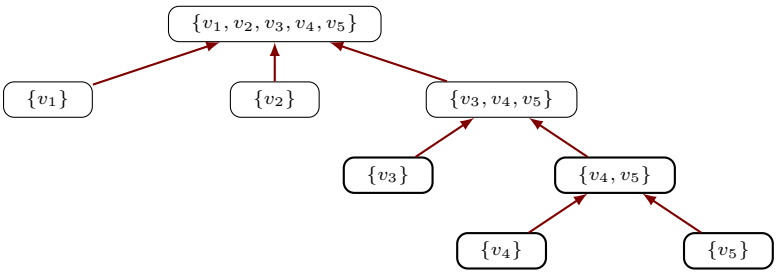

Figure 3. Example of a complex PAVO

Fig. 3. The following imprecise descriptions have the corresponding concepts only: $\{v_1, v_2, v_3, v_4, v_5\}$, $\{v_3, v_4, v_5\}$ and $\{v_4, v_5\}$.

Descriptions with the corresponding concepts in a given PAVO are associated to these concepts. However, the association of descriptions without corresponding concepts in a PAVO is not a trivial task and is analyzed later in this section.

For a given concept c_i in a PAVO, all concepts that are more specific than the concept c_i are the descendants of this concept c_i. The association of descriptions to corresponding concepts inherits this property.

REMARK 1 *For a given new example, whose description is associated to a corresponding concept c_i, all the training examples described by this concept c_i or described more precisely have descriptions, which are associated to this concept c_i or its descendants.*

EXAMPLE 2 *Given is an attribute A such that there $V = \{v_1, v_2, v_3, v_4, v_5\}$. Let us assume, that there are the following descriptions $Z_1 = \{v_3, v_4, v_5\}$ and $Z_2 = \{v_4, v_5\}$, where Z_1 is a description of a testing example and Z_2 is a description of a training example. We create a PAVO with two abstract concepts representing the sets Z_1 and Z_2, respectively. Therefore, sets Z_1 and Z_2 have corresponding concepts and are associated to these concepts. The resulting PAVO is presented in Fig. 4. As we can see, for a testing example Z_1 associated to a corresponding concept, all the training examples described by this concept or described more precisely are associated to this concept or its descendants.*

However, the association of descriptions to *not corresponding* concepts corrupts these descriptions. Let us associate a description Z_l to a concept c_i such

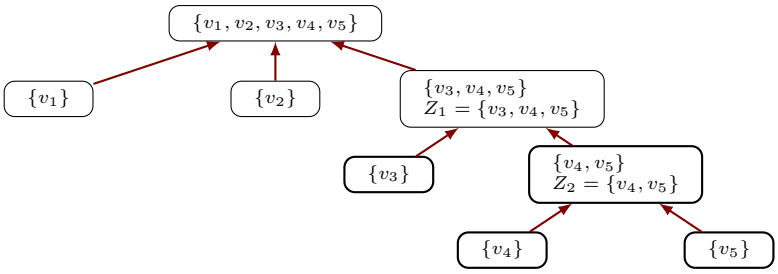

Figure 4. Association of descriptions to corresponding concepts

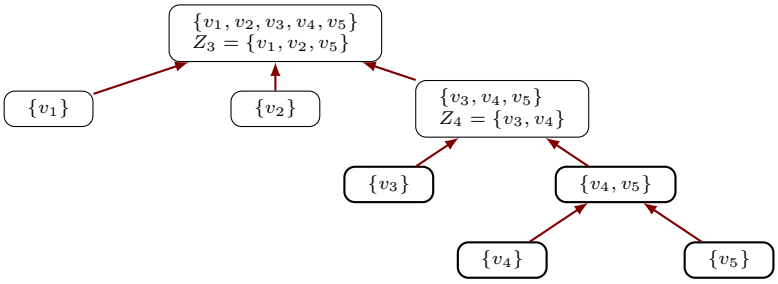

Figure 5. Association of descriptions to not corresponding concepts

that $Z_l \subset c_i$. In such a case, we increase (generalize) the level of knowledge granularity of this description and we may corrupt the classification results.

EXAMPLE 3 *We use the PAVO from the previous example. Let us assume that descriptions, that have not corresponding concepts, are represented by sets* $Z_3 = \{v_1, v_2, v_5\}$ *and* $Z_4 = \{v_3, v_4\}$. *Let us associate the description* Z_3 *to the root (*$Z_3 \subset$ *root) and the description* Z_4 *to the concept* $\{v_3, v_4, v_5\}$ *(*$Z_4 \subset \{v_3, v_4, v_5\}$*). The resulting PAVO is presented in Fig. 5. We can observe that we increase the level of knowledge granularity of both descriptions. Let us notice that the associated description* Z_4 *is a descendant of the associated description* Z_3. *However, the original description* Z_4 *is not more precisely described than the original description* Z_3. *The use of such a PAVO may corrupt the classification results: classification of* Z_3 *using* Z_4 *would be corrupted.*

Concluding, we are allowed to associate descriptions to corresponding nodes only. Descriptions without corresponding nodes in PAVO should not be used in the classification process, these descriptions may corrupt the classification results. The problem of descriptions without corresponding nodes can be solved by using an AVO with all possible primitive and abstracts concepts. An example of such AVO is presented in Fig. 6. We may notice that such an AVO may be always reduced to an AVO that uses all the necessary primitive and abstract concepts only.

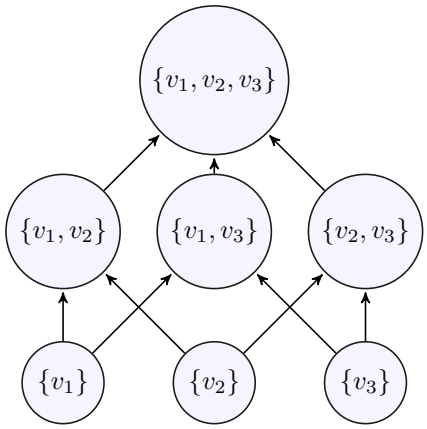

Figure 6. AVO representing all concepts ($V = \{v_1, v_2, v_3\}$)

6. Extending naïve Bayesian classifier by PAVO

Using PAVO 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 (using PAVO) according to our general strategy: to classify a new example using training examples described in the same way or more precisely at lower levels of knowledge granularity.

6.1. 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 relies upon, 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 $A_1, A_2, ..., A_n$. A (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:

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

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

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

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, direct application of these rules is difficult due to 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:

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

$P(v_i|C_j)$ 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.

6.2. Inference with ontological attributes

In the classification without abstract values, the naïve Bayesian classifier needs to estimate the value $P(v_i|C_j)$. In the proposed approach with abstract values, 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 remind that for a given concept c_i in a PAVO, all concepts that are more specific than the concept c_i are the descendants of this concept c_i. The association of descriptions to corresponding concepts in a PAVO inherits this property (Remark 1). In order to estimate this probability, e.g. by relative frequency, we use this property:

$$P(c_i|C_j) = \frac{\sum_{c_k \in \{c_i\} \cup desc(c_i)} n_{kj}}{n_j} \ , \tag{6}$$

where n_j is the number of training examples with 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, 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.

6.3. 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. However, an ontological attribute describing the domain of infectious agents presented in Fig. 1 is not a PAVO. The concept of Gram-positive Bacteria is more general than the concept of Streptococcus, yet both of these would be described by the same subset of specific values: Streptococcus. Therefore, we introduce a new specific value *Other* to make a clear difference between the concepts of Gram-positive Bacteria and Streptococcus. The resulting PAVO is presented in Fig. 7

The training data is given in Table 1. For the simplicity of presentation we consider only one ontological attribute. Therefore, all the cases with the same description of the infectious agent attribute are aggregated and the number of examples for each description is also given in Table 1.

All the descriptions have corresponding concepts, therefore all the training examples are used in the classification process. These associations are also presented in Fig. 7. For each concept we present the number of instances associated to this concept for each class. For example, the concept of *Bacteria* represents 6 instances with the class label C_1 and 7 instances with class label C_2. 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.

Infectious Agent = Bacteria Let us consider the following scenario: there is a patient and the diagnosis is *Bacteria*. We estimate the posterior probability $P(C_j|Bacteria)$. Therefore, we concentrate on these instances that are associated to the node *Bacteria* or its descendants. This fragment of our PAVO is presented in Fig. 8. From Equation 6 we have:

$$P(Bacteria|C_1) = \frac{6+3+1}{10} = 1 \qquad P(Bacteria|C_2) = \frac{7+2+1}{10} = 1.$$

From Equation 4 we have:

$$P(C_1|Bacteria) \propto 0.5 * 1 = 0.5 \qquad P(C_2|Bacteria) \propto 0.5 * 1 = 0.5.$$

As we can see, both class labels are equally probable. Therefore, we need to conduct a medical diagnosis test to know what kind of Bacteria is the infectious agent: Gram-positive or Gram-negative.

Table 1. A medical diagnosis training data

Number of 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

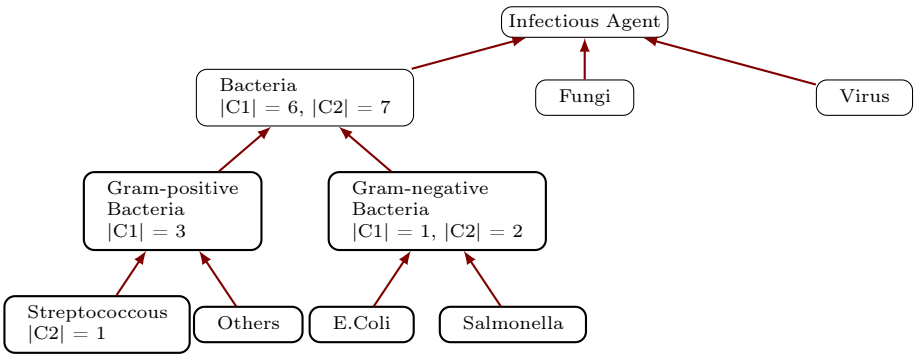

Figure 7. Example of a PAVO for the medical diagnosis problem

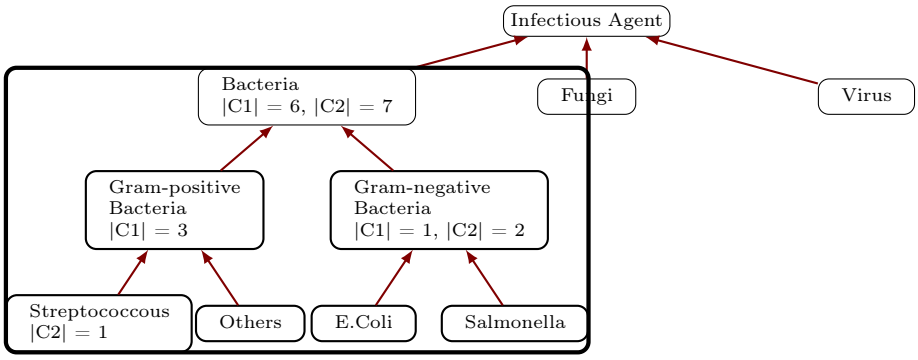

Figure 8. Infectious Agent = Bacteria

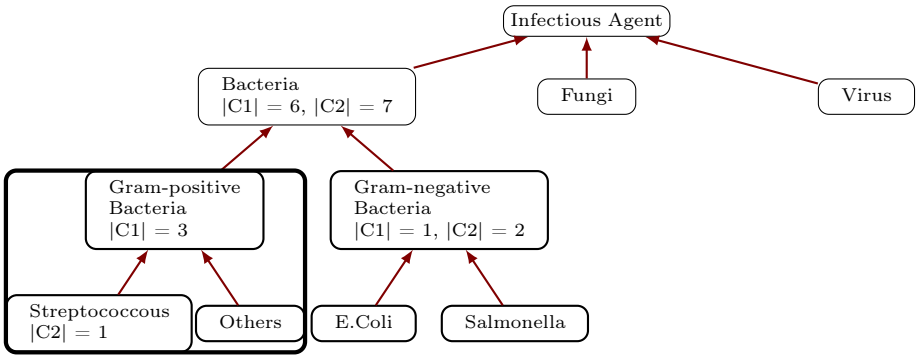

Figure 9. Infectious Agent = Gram-positive Bacteria

Infectious Agent = Gram-positive Bacteria The result of the test indicated that a Gram-positive Bacteria (shortly GP.B.) is the infectious agent. The analyzed fragment of PAVO is presented in Fig. 9. Taking into account this information we estimate the posterior probabilities. From Equation 6 we have:

$$P(GP.B.|C_1) = \frac{3}{10} = 0.3 \qquad P(GP.B.|C_2) = \frac{1}{10} = 0.1.$$

From Equation 4 we have:

$$P(C_1|GP.B.) \propto 0.5 * 0.3 = 0.15 \qquad P(C_2|GP.B.) \propto 0.5 * 0.1 = 0.05.$$

As we can see, for Gram-positive Bacteria class C_1 is three times more probable than class C_2 and the treatment represented by class C_1 is recommended.

Infectious Agent = Gram-positive Bacteria and not Streptococcus Finally, let us assume that this Gram-positive Bacteria is not Streptococcus. However, we have no training example described by the primitive concept *Others*. Therefore, we are not able to make a decision for a testing instance described by this value. However, we may take into account the semantics of the PAVO: Each 'Gram-positive Bacteria other than Streptococcus' (Others) is a 'Gram-positive Bacteria'. The analyzed fragment of PAVO is presented in Fig. 10. The concept of 'Gram-positive Bacteria' has 3 instances that support class C_1 only and class C_1 is recommended.

This example shows that, we can use both 'positive' observations (Bacteria is a Gram-positive Bacteria) and 'negative' observations (Gram-positive Bacteria is not a Streptococcus) in order to get more precise descriptions of new examples.

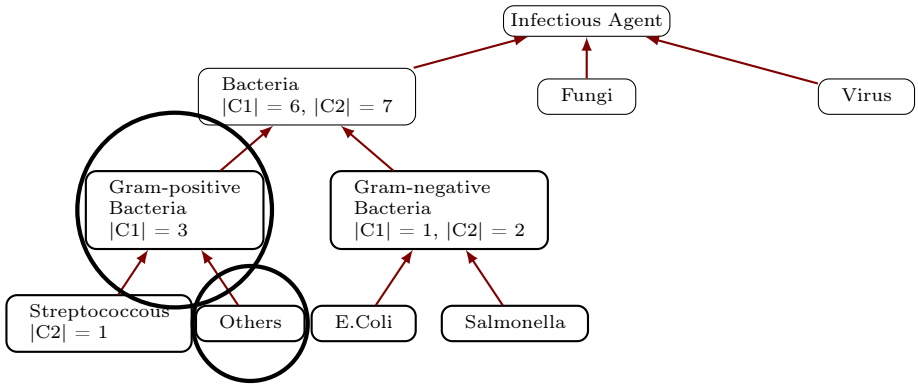

Figure 10. Infectious Agent = Gram-positive Bacteria and not Streptococcus

7. Summary

In this paper we proposed a novel extension of the naïve Bayesian classifier by an attribute value ontology (AVO). In the proposed approach, every attribute is a hierarchy of concepts from the domain knowledge base. In this way an example is described either very precisely (using a concept from the low-level of the hierarchy) or, when it is not possible, in a more general way (using a concept from higher levels of the hierarchy). Our general strategy is to classify a new example using training examples described in the same way or more precisely at lower levels of knowledge granularity.

The proposed approach is a generalization of the classical approach. 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, 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 training examples described by this concept and *also* all training examples described by descendants of c_i.

Our proposal is motivated by the problem of description noise which may occur when the language used to model attribute values in the training as well as testing dataset is insufficient. In such a case, erroneous attribute values and missing attribute values may be introduced by users that are required to provide very specific values. By introducing an attribute value ontology (AVO) we are able to use the examples described at different levels of granularity in classification and 'prevent' from introducing erroneous or missing attribute values.

The semantics of the AVO allows for performing two different diagnostic tests in order to get more precise descriptions of new examples. We have shown, that the more precise observation can be reached not only by 'positive' observations

but also by the 'negation' of subconcepts of a given concept. Such a rejection of a hypothesis is a very common approach, e.g. in a medical diagnosis.

For simplicity of presentation, all considerations were performed for a PAVO. We should be aware that the association of examples to not corresponding concepts in a PAVO corrupts these descriptions and may corrupt the classification results. This problem can be solved by using an AVO with the necessary primitive and abstract concepts only.

In the future we plan to use an AVO to model possible diagnostic tests and their characteristics: costs, time and interactions with the objects.

Acknowledgments

This research was supported by the Ministry of Science and Higher Eduction Research 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: *AAAI/IAAI, Vol. 1.* AAAI press, 703–708.

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

Clark, P. and Niblett, T. (1987) Induction in Noisy Domains. In: *Progress in Machine Learning (Proceedings of the 2nd European Working Session on Learning)*, Sigma Press, Bled, Yugoslavia, 11–30.

Han, J., Cai, Y. and Cercone, N. (1992) Knowledge Discovery in Databases: An Attribute-Oriented Approach. In: L.Y. Yuan, ed., *VLDB*, Morgan Kaufmann, 547–559.

Han, J. and Kamber, M. (2006) *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann.

Haussler, D. (1988) Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. *Artif. Intell.*, **36**(2), 177–221.

Hickey, R.J. (1996) Noise Modelling and Evaluating Learning from Examples. *Artif. Intell.*, **82** (1-2), 157–179.

Kudoh, Y., Haraguchi, M. and Okubo, Y. (2003) Data abstractions for decision tree induction. *Theor. Comput. Sci.*, **292**(2), 387–416.

Núñez. M. (1991) The Use of Background Knowledge in Decision Tree Induction. *Machine Learning*, **6**(3), 231–250.

Quinlan, J.R. (1986) Induction of Decision Trees. *Machine Learning*, **1**(1), 81–106.

Tanaka, H. (1996) Decision Tree Learning Algorithm with Structured Attributes: Application to Verbal Case Frame Acquisition. In: *COLING.* Center for Sprogteknologi, Copenhagen, 943–948.

Taylor, M.G., Stoffel, K. and Hendler, J.A. (1997) Ontology-based Induction of High Level Classification Rules. In: *Proceedings of the SIGMOD Dataming and Knowledge Discovery Workshop*. ACM Press.

Walker, A. (1980) On Retrieval from a Small Version of a Large Data Base. In: *VLDB*, IEEE Computer Society, 47–54.

Wu, X. (1995) *Knowledge Acquisition from Databases*. Ablex Publishing Corp., Norwood.

Zhang, J., Kang, D., Silvescu, A. and Honavar, V. (2006) Learning accurate and concise naïve Bayes classifiers from attribute value taxonomies and data. *Knowl. Inf. Syst.*, **9**(2):157–179.

Zhang, J., Silvescu, A. and Honavar, V. (2002) Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction. In: S. Koenig and R.C. Holte, eds., *SARA*, LNCS 2371, Springer, 316–323.

Zhu, X. and Wu, X. (2004) Class Noise vs. Attribute Noise: A Quantitative Study. *Artif. Intell. Rev.*, **22**(3), 177–210.