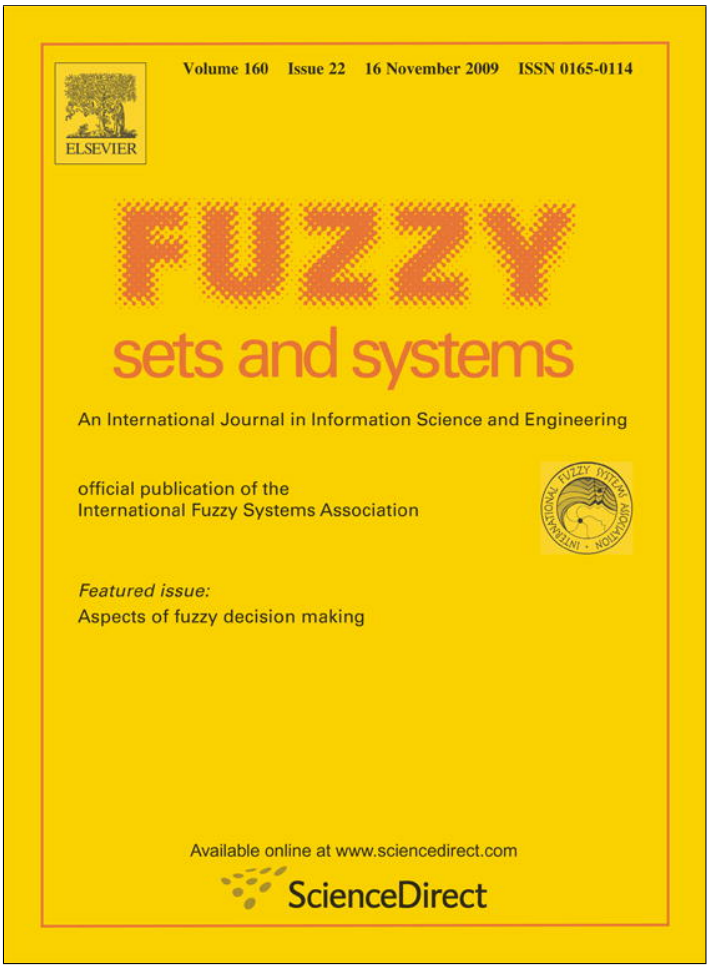
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 160 (2009) 3224–3238
FUZZY sets and systems
www.elsevier.com/locate/fss

Naïve possibilistic network classifiers

Bakhta Haouari[a,*], Nahla Ben Amor[a], Zied Elouedi[a], Khaled Mellouli[b]

[a]*LARODEC, Institut Supérieur de Gestion de Tunis, 41 Avenue de la liberté, 2000 Le Bardo, Tunisia*
[b]*Institut Hautes Etudes Commerciales de Tunis, Tunisia*

Available online 5 February 2009

Abstract

Naïve Bayesian network classifiers have proved their effectiveness to accomplish the classification task, even if they work under the strong assumption of independence of attributes in the context of the class node. However, as all of them are based on probability theory, they run into problems when they are faced with imperfection. This paper proposes a new approach of classification under the possibilistic framework with naïve classifiers. To output the naïve possibilistic network classifier, two procedures are studied namely the building phase, which deals with imperfect (imprecise/uncertain) dataset attributes and classes, and the classification phase, which is used to classify new instances that may be characterized by imperfect attributes. To improve the performance of our classifier, we propose two extensions namely selective naïve possibilistic classifier and semi-naïve possibilistic classifier. Experimental study has shown naïve Bayes style possibilistic classifier, and is efficient in the imperfect case.
© 2009 Elsevier B.V. All rights reserved.

Keywords: Possibility theory; Classification; Naïve Bayes classifier; Possibilistic classifier; Aggregation operators

1. Introduction

Classification presents an important task in machine learning and data mining applications. It consists in inducing a classifier from a set of historical examples known as training set, where each instance is characterized by an attribute set with known class values. Then, the induced classifier is used to predict the class value of new instances given known values of their attributes.

Various learning methods have been developed to ensure classification. The commonly used ones in supervised learning are decision trees [29], neural networks [4], K-nearest neighbors [7], Bayesian networks [28], etc.

In this paper, we are interested in a particular case of Bayesian networks which are naïve Bayesian networks. They are one of the most efficient classifiers, in the sense that their predictive performance is competitive with state-of-the-art classifiers as decision trees [8]. Nevertheless, *naïve Bayes classifier* (NBC) as most probabilistic approaches runs into problems when faced with imperfection that may pervade datasets.

Several non-classical theories of uncertainty have been proposed in order to deal with uncertain and imprecise data such as, evidence theory [31], fuzzy set theory [33] and possibility theory [9,10]. This fact has given us the idea to build a new classifier, called *naïve possibilistic network classifier*, based on NBC, to take benefit from its simplicity and high accuracy, and use possibility theory to handle imperfection.

* Corresponding author. Tel.: +216 96 671 317; fax: +216 71 588 487.
E-mail address: bakhtahaouari@yahoo.fr (B. Haouari).

0165-0114/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2009.01.009

Author's personal copy

This paper is organized as follows. Section 2 presents a study of some existing Bayesian classifiers. In Section 3, we briefly recall some basics of the possibility theory. In Section 4, we give an overview on some existing possibilistic network (PN) classifiers. Section 5 develops our new approach and details building and classification phases. Finally, Section 6 describes the experimental setup and results showing the efficiency of our classifier.

2. Bayesian classification

Bayes networks are powerful tools for decision and reasoning under uncertainty. A very simple form of Bayes networks is called Naïve Bayesian classifiers [13,18,22,23,28] dedicated to the classification task. Such networks are composed of DAGs with a unique structure having one root node (called parent), representing the class node, and several children, corresponding to attribute nodes. NBC work under the strong *naïve* assumption of independence among child nodes in the context of their parent. Hence, in presence of a training set we should only compute conditional probabilities in a frequentist manner since the structure is unique.

Once the NBC is constructed, it can be used to classify any new instance characterized by a set of attributes x_1, $x_2, \ldots, x_n$ by computing for each possible class c_i its posterior probability and then by taking the highest one. More formally the most probable class c^* is computed as follows:

$$c^* = \arg\max_{c_i} p(c_i) \prod_{j=1}^{n} p(x_j|c_i) \tag{1}$$

Although the independence assumption may be problematic since, non-realistic, naïve-Bayes has surprisingly outperformed many sophisticated classifiers over a large number of datasets, especially where the features are not strongly correlated [22].

Nevertheless, other Bayesian classifiers taking into account more or fewer dependencies between variables have appeared. In fact, it has been shown in [19,25] that the redundancy of variables affects the classification results, hence, a new approach called *selective naïve Bayes* (SBC) [20,23] was proposed. The principle of this approach is to search for a good subset of the attributes, using an estimation scheme, such as cross validation, to repeatedly evaluate the predictive accuracy of the NBC on various subsets and to remove redundant variables in order to obtain more efficient classifiers [22,23,27].

Another proposed approach is the *semi-naïve Bayes classifier* (SNBC) [21]. It can be considered as a more sophisticated type of the Bayesian classifier regarding the type of dependencies that it can take into account, as it allows groups of partially correlated variables to be considered as a single node in the Bayesian network. In fact, in the case of two fully correlated attributes, the removal of one can only improve the performance of the NBC. Problems arise if two attributes are only partially correlated. In this case, the removal of an attribute may lead to the loss of useful information, and the SBC may still retain both attributes. The aim of SNBC is to avoid the strict premises of the naïve Bayes paradigm by allowing to group some variables in a cluster Clu_i.

Other examples of Bayesian classifiers that can be found in the literature are the *tree augmented naïve Bayes* [16], *Bayesian multi-net* [17], *general Bayesian network* [16], *K-dependence Bayesian classifier* [30], *extended tree-augmented naïve Bayes* [24], and *clique-augmented naïve Bayes* [24].

3. Basics of possibility theory

In this work, we are interested in possibility theory as an approach to uncertainty induced by pieces of vague linguistic information. This theory was initiated by Zadeh [34] in the late seventies and developed by Dubois and Prade [9,10]. In the following, we only give a brief recalling on possibility theory, for more details see [9–12].

Let $V = \{A_1, A_2, \ldots, A_M\}$ be a set of variables. We denote by $D_A = \{a_1, \ldots, a_m\}$ the domain associated with the variable A. By a we denote any instance of A. $\Omega = \times_{A_i \in V} D_{A_i}$ denotes the universe of discourse, which is the Cartesian product of all variable domains in V.

Possibility theory is based on the notion of *possibility distribution* denoted by π and which corresponds to a mapping from Ω to the scale $[0, 1]$ encoding our knowledge on the real world, denoted by w. The possibility distribution represents a state of *knowledge* distinguishing what is plausible from what is less plausible. A possibility distribution

Author's personal copy

is often attached to a variable ω ranging on Ω and then denoted $\pi(\omega)$. Such a variable ω stands for the unknown value taken by one or several attributes used to describe Ω.

Given a possibility distribution π defined on Ω, we can define a mapping grading the possibility measure of an event $\varphi \subseteq \Omega$ by $\Pi(\varphi) = \max_{\omega\in\varphi} \pi(\omega)$. A possibility distribution π is said to be α-normalized, if $\max_\omega \pi(\omega) = 1$, then π is simply said to be normalized. The particularity of the possibilistic scale is that it can be interpreted in two manners: in an *ordinal* manner if the handled values reflect only an ordering between the different states of the world, or in a *numerical* manner if the handled values make sense in the ranking scale.

Conditioning is also a critical notion in possibility theory, and it consists in modifying our initial knowledge, encoded by the possibility distribution π by the arrival of a new fully certain piece of information e. Conditioning depends on the interpretation of the possibilistic scale, in fact:

- The *min-based conditioning* corresponds to the ordinal setting, when we assign to the best elements of φ, the maximal possibility degree and we obtain

$$\pi(\omega|_m\varphi) = \begin{cases} 1 & \text{if } \pi(\omega) = \Pi(\varphi) \text{ and } \omega \in \varphi \\ \pi(\omega) & \text{if } \pi(\omega) < \Pi(\varphi) \text{ and } \omega \in \varphi \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

- The *product-based conditioning* corresponds to the numerical setting. In this case we proportionally shift up all elements of φ as follows:

$$\pi(\omega|_p\varphi) = \begin{cases} \dfrac{\pi(\omega)}{\Pi(\varphi)} & \text{if } \omega \in \varphi \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

4. Possibilistic network classifiers

PN were defined as an alternative to Bayesian networks in the possibilistic framework with the same complexities. In fact, possibilistic propagation remains an NP-complete problem [2,15]. This statement makes such networks unsuitable to the classification task due to the amount of data we have to manipulate. Our idea is then to propose a possibilistic counterpart of naïve Bayesian networks in order to take benefit from the advantages of this technique (see Section 2).

An attempt in the development of possibilistic classifiers is *naïve Bayes style possibilistic classifier* (NBSPC) developed by Borgelt et al. [6]. The NBSPC is induced from imprecise training sets. Imprecision is localized in attributes of instances excepting the class attribute, whereas the testing set (Ts) is perfect. The basic idea is to replace each imprecise instance by n precise ones, where n is the product of domain cardinals of the imprecise attributes in this instance. The resulted dataset is named extended dataset, it only contains precise instances (without redundancy) thus a column containing the frequency of each instance is added. The possibility distribution of an attribute given the class is inferred from the compute of the maximum projection [5] over the set S of precise instances (S is included in the extended dataset) that contains both the target value of the attribute A with the class c_i. More formally

$$\pi(a_j|c_i) = \frac{1}{K} \max_S \mathit{Frequence}(a_j, c_i), \tag{4}$$

where K is the cardinal of the training set and a_j is an instance of A.

- The classification model is expressed by (it is defined with sub-normalized possibility distributions):

$$\pi(c_i|a_1, \ldots, a_j, \ldots, a_n) = \pi(a_1, \ldots, a_j, \ldots, a_n|c_i) = \min_{j=1,\ldots,n} \pi(a_j|c_i), \tag{5}$$

Then the class of a given instance $(a_1, \ldots, a_j, \ldots, a_n)$ is c^* computed as

$$c^* = \arg\max_{c_i} \min_{j=1,\ldots,n} \pi(a_j|c_i) \tag{6}$$

NBSPC is specified to handle a particular case of imperfection which is imprecise attributes present only in the training set. To overcome this limitation, we develop what we call a *naïve possibilistic network classifier* (NPC), a new

Author's personal copy

classification technique based on naïve Bayes networks within the framework of possibility theory. NPC is able to deal with imperfection that can appear either in the construction or in the classification phase.

5. Naïve possibilistic network classifiers

In this section, we first focus on the construction of the NPC from an imperfect training set then we present the classification process of both perfect and imperfect testing set.

5.1. Building procedure

Generally, training sets are assumed to be well known. This is not always the case. In fact, the structure of the training set may contain data where attributes or classes are ill-known. We consider here, the case where this imperfection is represented by means of possibility distributions. These latters express partial ignorance, due to a lack of a priori knowledge, as to the actual value of the class of a given instance in the training set. To deal with this situation, we propose to calculate a new value which better represents the plausibility of belonging of an instance to a given class. This new value is called possibilistic mean (π_{Pm}), it is based on the independence assumption of attributes in NBC and it is only defined in the product-based setting.

Processing of uncertain classes in the training set: Let Tr be a training set composed of P instances characterized by K symbolic attributes and a possibility distribution π reflecting the uncertainty over instance classes. Our problem is to find a combination of the different possibility distributions in order to obtain their general representative value.

Possibility theory presents many operators for information fusion. The fusion process aims at finding the most plausible values of the parameter w on the basis of the available pieces of information. Different combination modes are proposed, such as, the conjunctive and disjunctive modes [12]. However, all these modes are used to combine possibility distributions issued from distinct sources and which must pertain to the same variable that means, possibilistic combination operators are particularly suitable when heterogeneous sources provide pieces of information pertaining to the same object. However, in our case, sources of information are viewed as instances of a single random source and they supply point-value estimates of the possibility distribution related to different instance classes. These values reflect the possibility of getting the given class value as the actual one. Then, our problem is closer to being an estimation task than a fusion.

To perform the estimation task, some function F, called estimation function, should be used, such a function takes multiple observations of a parameter w and provides a single representative value of them. In what follows we introduce necessary properties that this function should satisfy to be a mean operator [12,32]:

(a) *Idempotency*: If all the x_i are the same then our estimated value should be this value.
(b) *Commutativity*: The indexing of the observations does not affect the final aggregated value.
(c) *Monotonicity*: It is generally required, $x_i \geqslant a_i$ then $F(x_1, \ldots, x_n) \geqslant F(a_1, \ldots, a_n)$.

Indeed these three properties are sufficient to ensure that $F(x_1, \ldots, x_n)$ lies between $\min(x_1, \ldots, x_n)$ and $\max(x_1, \ldots, x_n)$.

The most used functions for F are the geometric mean, the harmonic mean, and the arithmetic mean. The two first functions are not adequate for our case. In fact, the geometric mean is used for a set of variables whose values are dedicated to be multiplied together or are exponential in nature, and we have to use this mean when the considered variable evolves in time, which is not our case. Moreover, the harmonic mean is an average, useful for variables expressed by two units, which is, also, not our case. The third function, which is the arithmetic mean, is relevant to handle commensurable quantities, which corresponds to our case, this motivates us to use it in order to solve our problem (*in product-based setting*). In order to adapt this operator to the possibilistic framework, we propose to use the following formula:

$$\pi_{Pm}(a_j|c_i) = \underset{\mathrm{Tr}(a_j,c_i)}{mean}\ \pi(I_k|c_i) \tag{7}$$

where $\pi_{Pm}(a_j|c_i)$ is called the *possibilistic mean* of the attribute value a_j given the class c_i, computed over Tr and $\pi(I_k|c_i)$ is the possibility distribution of the attribute values (I_k such that a_j belongs to I_k) of the instance k given the class c_i.

Author's personal copy

Table 1
Imperfect golf training set.

No	Outlook	Temperature	Humidity	Wind	$\pi(I_k\|c_i)$	
					No	Yes
1	{*sunny*, *rain*}(0.9)	*hot*	*high*	*weak*	1	0.4
2	*sunny*	*hot*	*high*	*strong*	1	0.6
3	*overcast*	{*hot*, *cool*, *mild*}	*high*	*weak*	0.1	1
4	*rain*	*mild*	{*high*, *normal*}	*weak*	0.3	1
5	{*rain*, *sunny*}	*cool*	*normal*	*weak*	0.5	1
6	*rain*	{*cool*, *hot*}(0.2)	*normal*	*strong*	1	0
7	*overcast*	*cool*	*normal*	*strong*	1	0.8
8	{*sunny*}(0.6)	*mild*	*high*	*weak*	0.9	1
9	*sunny*	{*cool*, *mild*}(0.8)	*normal*	*weak*	0.6	1
10	*rain*	*mild*	*normal*	*weak*	0	1
11	*sunny*	*mild*	*normal*	*strong*	1	0.8
12	*overcast*	*mild*	*high*	*strong*	0.4	1
13	*overcast*	*hot*	*normal*	*weak*	0.3	1
14	*rain*	*mild*	*high*	{*strong*, *weak*}	1	0

Processing of imperfect attributes in the training set: In general, a training set can be characterized by some imperfect attributes. Thus, instead of an exact attribute value, we have a possibility distribution α over a possible set S of candidate values that belong to the attribute domain. In fact, α is a scaled [0, 1] uncertainty factor that reflects the degree of belief of the expert that the set S contains the actual attribute value. Elements of the set S are supposed equipossible to be the real attribute value thus, following the Principle of Insufficient Reason, to each one of them we assign a possibility distribution $\beta = 1/n$, where β expresses the imprecision relative to the attribute and n denotes the cardinal of S. Then, we have to weight the possibilistic mean to consider the β and α factors. Given β and α factors, the possibilistic mean π_{Pm} of an attribute value a_j given the class c_i is

$$\pi_{Pm}(a_j|c_i) = \underset{\mathrm{Tr}(a_j,c_i)}{mean}\ \beta_j\alpha_j\pi(I_k|c_i) \tag{8}$$

where $\pi_{Pm}(a_j|c_i)$ is the possibilistic mean of the attribute value a_j given the class c_i, computed over Tr and $\pi(I_k|c_i)$ is the possibility distribution of the attribute values (I_k such that a_j belongs to I_k) of the instance k given the class c_i.

Example 1. Let Tr be the imperfect golf training set represented by Table 1 and composed of 14 instances characterized by four symbolic attributes (outlook, temperature, humidity and wind) and belonging to two exclusive classes C (no, yes), where 0.6 in the second line of Table 1 means that we have 60% that the actual class of the instance (sunny, hot, high, strong) is Yes.

To quantify the network we use Eq. (8). For instance

$$\pi_{Pm}(sunny|Yes) = (0.4 * \mathbf{0.9} * 0.5 + 0.6 + 1 * 0.5 + 1 * \mathbf{0.6} + 1 + 0.8)/6 = 0.61,$$

(where bold numbers represent the uncertainty factors and italic numbers correspond to imprecision factors). The initial possibility distribution values, computed from Eq. (8), are represented in Table 2. Values between parentheses correspond to normalized values of these distributions. For the sake of simplicity we use, in Table 2 and in the rest of the paper $\pi(a_j|c_i)$ instead of $\pi_{Pm}(a_j|c_i)$.

5.2. Classification procedure

In this section, we first present a classification method of perfect testing set. Then, an extension of our method handling imperfect instances is developed.

Classification of perfect instances: In order to classify new instances, we should define a model that expresses the relation between the class attribute and the instance attributes. This model reflects the influence of the evidence localized in the attributes and the class of an instance. Since NPC is a simplified model of PN and since classification task is

Author's personal copy

Table 2
Initial and normalized distributions.

$\pi(C)$	π(outlook $\|C$)	π(Temperature $\|C$)	π(Humidity $\|C$)	π(Wind $\|C$)
$\pi(No)$: 0.65(0.86)	$\pi(sunny\|No)$: 0.64(1) $\pi(overcast\|No)$: 0.45(0.7) $\pi(rain\|No)$: 0.5(0.78)	$\pi(hot\|No)$: 0.48(1) $\pi(mild\|No)$: 0.48(1) $\pi(cool\|No)$: 0.37(0.74)	$\pi(high\|No)$: 0.65(1) $\pi(normal\|No)$: 0.56(0.86)	$\pi(weak\|No)$: 0.46(0.56) $\pi(strong\|No)$: 0.81(1)
$\pi(Yes)$: 0.75(1)	$\pi(sunny\|Yes$: 0.61(0.64) $\pi(overcast\|Yes)$: 0.95(1) $\pi(rain\|Yes)$: 0.45(0.47)	$\pi(hot\|Yes)$: 0.45(0.6) $\pi(mild\|Yes)$: 0.74(1) $\pi(cool\|Yes)$: 0.5(0.67)	$\pi(high\|Yes)$: 0.64(0.84) $\pi(normal\|Yes)$: 0.76(1)	$\pi(weak\|Yes)$: 0.82(1) $\pi(strong\|Yes)$: 0.53(0.65)

a special case of propagation process, naïve possibilistic model can be deduced from general propagation in PN (the proof is given in the Appendix).

PN have two settings depending on the definition of the possibilistic conditioning: the product-based one and the min-based one then, to each setting a classification model is distinguished. In the product-based setting a given instance is assigned to the most plausible class c^* defined as

$$c^* = \arg\max_{c_i} \prod_{j=1}^{m} \Pi(a_j|c_i) \tag{9}$$

Example 2. Let us continue with the same example of golf-base treated in the building step. Following Eq. (9) the instance (*sunny*, *mild*, *normal*, *weak*) will be assigned to

$$\begin{aligned} c^* &= \arg\max_{\{No,Yes\}} \Pi(sunny|PG)\cdot\Pi(mild|PG)\cdot\Pi(normal|PG)\cdot\Pi(weak|PG) \\ &= \arg\max([\Pi(sunny|No)\cdot\Pi(mild|No)\cdot\Pi(normal|No)\cdot\Pi(weak|No)],\ [\Pi(sunny|Yes, \\ &\Pi(mild|Yes\cdot\Pi(normal|Yes\cdot\Pi(weak|Yes]) \\ &= \arg\max([1*1*0.86*0.56],\ [0.64*1*1*1]) \\ &= \arg\max(0.47, \mathbf{0.64}). \end{aligned}$$

Hence the instance is assigned to the class Yes.

As in product-based setting the classification model, in the min-based setting, can be deduced from the min-based propagation in PN thus a given instance is assigned to the class c^* defined as

$$c^* = \arg\max \min_{j=1}^{m} \pi(a_j \wedge u) \tag{10}$$

Example 3. Let us reconsider values in Table 1 and suppose that the possibility distributions of Table 2 are valid in min-based setting (since the possibilistic mean has been only validated in product-based setting). Following Eq. (10) the instance (*sunny*, *mild*, *normal*, *weak*) is assigned to

$$\begin{aligned} c^* &= \arg\max_{\{No,Yes\}} \min_{j=1}^{m} \pi(sunny\wedge PG),\ \pi(mild\wedge PG),\ \pi(normal\wedge PG),\ \pi(weak\wedge PG) \\ &= \arg\max_{\{No,Yes\}} min([\pi(sunny\wedge No),\ \pi(mild\wedge No),\ \pi(normal\wedge No),\ \pi(weak\wedge No)], \\ &[\pi(sunny\wedge Yes),\ \pi(mild\wedge Yes),\ \pi(normal\wedge Yes),\ \pi(weak\wedge Yes)]) \\ &= \arg\max\min([1;\ 1;\ 0.86;\ 0.56];\ [\mathbf{0.64};\ 1;\ 1;\ 1]) = Yes. \end{aligned}$$

Classification of imperfect instances: When they are considered as a classifier PN deal with perfect testing set whereas, in the NPC, attributes of instances to classify may be described by a set S of values. Instead of having a perfect value for an attribute, we have an uncertainty degree α over the set S. The α degree expresses the possibility of the assigned set S to contain the true value of the attribute. If S contains more than one element then the expert is

Author's personal copy

Table 3
Imperfect instance $\overrightarrow{i_1}$.

π_{outlook}	$\pi_{\text{Temperature}}$	π_{Humidity}	π_{Wind}
sunny: 1	*hot*: 0	*high*: 1	*strong*: 0
overcast: 0.8	*mild*: 0	*normal*: 0	*weak*: 1
rain: 0	*cool*: 1		

also imprecise and since we have no indication that prefers one or some values over the others, elements of S have the same possibility β to be the true value (since we have no indication that prefers one or some values over the others). Thus they are treated in an equipossible manner and following the Principle of Insufficient Reason, β is equal to $1/n$. This effect is considered in the computation of the possibility distribution of the attribute given the class throughout the whole dataset by weighting this distribution by the α and β degrees. Thus, whenever an instance contains an imperfect attribute A, the idea consists in discounting the initial possibility distribution computed via the possibilistic mean in the building step in accordance with the α and β degrees. The possibility distribution of the class c_i given the set of attribute values of an instance k is

$$\pi(c_i|I_k) = \arg\max_{c_i} \prod_{j=1}^{m} \beta_j * \alpha_j * \Pi(a_j|c_i) \tag{11}$$

Example 4. Let us continue with the same example of golf-dataset and consider a new instance $\overrightarrow{i_1}$ (see Table 3) to classify, we first start by computing the different possibility distributions π' relative to each value that the attribute outlook and the class C can take

$$outlook = overcast : \pi' = \pi(overcast|C) * \alpha * 1/n * \pi(cool|C) * \pi(high|C) * \pi(weak|C),$$

$$\begin{aligned} If\ C = No : \pi' &= \pi(overcast|No) * \alpha * 1/n * \pi(cool|No) * \pi(high|No) * \pi(weak|No) \\ &= 0.7 * 0.8 * 0.74 * 1 * 0.56 = 0.23, \end{aligned}$$

$$\begin{aligned} &If\ C = Yes: \pi' = \pi(overcast|Yes) * \alpha * 1/n * \pi(cool|Yes) * \pi(high|Yes) * \pi(weak|Yes) \\ &\quad = 1 * 0.8 * 0.67 * 0.84 * 1 = \mathbf{0.45} \end{aligned}$$

$$outlook = sunny : \pi' = \pi(sunny|C) * \alpha * 1/n * \pi(cool|C) * \pi(high|C) * \pi(weak|C),$$

$$\begin{aligned} &If\ C = No : \pi' = \pi(sunny|No) * \alpha * 1/n * \pi(cool|No) * \pi(high|No) * \pi(weak|No) \\ &\quad = 1 * 1 * 0.74 * 1 * 0.56 = 0.41 \end{aligned}$$

$$\begin{aligned} If\ C = Yes: \pi' &= \pi(sunny|Yes) * \alpha * 1/n * \pi(cool|Yes) * \pi(high|Yes) * \pi(weak|Yes) \\ &= 0.64 * 1 * 0.67 * 0.84 * 1 = 0.36 \end{aligned}$$

The instance is assigned to the class having the highest possibility distribution i.e. Yes.

5.3. Improvement methods of naïve possibilistic networks

In this section, we propose two new classifiers namely, the *selective naïve possibilistic network* (SPC) and *the semi-naïve possibilistic network* (SNPC), as a possibilistic counterpart of SBC and SNBC.

The idea in SPC consists in eliminating attributes one at a time, if the classifier accuracy is improved, the attribute is definitively removed from the model otherwise it is inserted again. Selecting the best set of attributes is an intractable problem, since it is equivalent to learning the best PN. Thus, even if we can improve the performance of the NPC in this way, the required computational effort may not be worthwhile unless, the feature selection is done in an acceptable time.

In SNPC, we allow variables to form local groups of intra-dependent attributes, and then these groups are naïvely considered inter-dependent. In other terms, instead of computing $P(X_1, X_2|C)$ (as in SNBC), we will compute $\pi(X_1, X_2|C)$ following the possibilistic mean method.

Author's personal copy

Table 4
Description of databases.

Database	# Data	# Tr	# Ts	# Attributes	# Classes
Wisconsin Breast Cancer	699	629	70	8	2
Congressional Voting	497	448	49	16	2
Balance Scale Weight	625	562	63	4	3
Solar Flare	925	832	93	10	2
Nursery	750	675	75	8	5

6. Implementation and simulation

In this section we present principal results of the realized experiments.

6.1. Generation of imperfect dataset

Available databases are perfect, thus we have to artificially introduce imperfection. Standard instances are modified, to each imperfect attribute we assign a normalized random set of possibility distributions, in such a manner that the actual value receives the highest possibility distribution which is equal to 1. The number of imperfect attributes is determined by the degrees of imperfection given as input to the classification model (the same modification is applied to the class attribute if we are dealing with a Tr).

We have applied our approach to five real databases obtained from the University of California at Irvine (U.C.I) repository of Machine Learning databases [26], namely, the Wisconsin Breast Cancer database, the Congressional Voting database, the Balance Scale Weight database, the Solar Flare database and the Nursery database. A brief description of these nominal-valued databases is given in Table 4 where #Data, #Tr, #Ts, #attributes, and #classes denote, respectively, the total number of instances in the database, the number of training instances, the number of testing instances, the number of attributes and the number of classes.

6.2. Evaluation measures

As a first evaluation measure we use the *percent of correct classification* (*PCC*) defined as follows:

$$PCC = \frac{\textit{number of well classified instances}}{\textit{total number of classified instances}} * 100 \tag{12}$$

For each instance to classify, we choose the class having the highest degree of possibility. If more than one class is obtained, then one of them is randomly chosen.

The limit of the *PCC* as described above is that it chooses randomly one of the more plausible classes which may misclassify some instances. Moreover, it ignores the possibility degrees assigned to the classes other than the selected one since it only considers the most plausible class.

To overcome these problems, we will use a second evaluation measure i.e. the *qualitative-percent of correct classification* (*Q_PCC*) already proposed in [3]. The *Q_PCC* is based on a qualitative distance measure (D), which allows the computation of the error degree between the real class and the one supplied by the *NPC*:

$$Q_PCC = \sum_{i \in \textit{classified objects}} D(i)/\textit{number of classified objects} \tag{13}$$

where $D(i) = 1 - d(i)/n$, $d(i) = \sum_{j=1}^{n} (\pi'_j(c_i) - \pi_r(c_i))^2$, $\pi'_j(c_i)$ is the possibility distribution computed and assigned to the class c_i supposed to be the actual class value, n is the number of classes and $\pi_r(c_i)$ is equal to 1 for the actual class and to 0 otherwise.

In the probabilistic setting d is computed as follows (for more details see [14]): $d(i) = (1 - p_{c_r})^2 + \sum_{j=2}^{n} p_j(c_i)^2$, where p_{Cr} denotes the probability distribution of the real class and $p_j(c_i)$ denotes the probability distribution of the class c_i (other than the real class). D and *Q_PCC* are computed as in the possibilistic case.

Author's personal copy

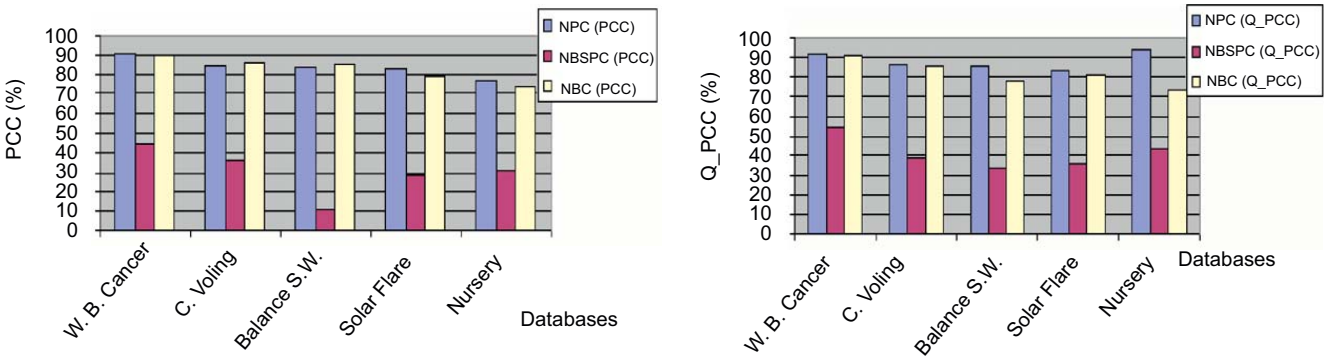


Fig. 1. Experimental *PCC* and *Q_PCC* in the perfect case.

In order to obtain an unbiased estimation of the *PCC* and *Q_PCC* in the simulations step, we have used a 10-fold cross validation. Three cases are distinguished: the first one is the process of perfect training and testing sets. In this case, a comparison with NBC and NBSPC is made up. Case two deals with imprecise training sets and a comparison with NBSPC is made up. Finally the third case handles imperfection in training and testing sets.

6.3. Results of the perfect case

Table 5 shows the performances (*PCC* and *Q_PCC*) of NBC, NBSPC and NPC applied on the five mentioned datasets using the 10-fold-cross validation.

- *PCC analyzes*: By comparing the performance of the three classifiers we can see that, when the dataset has a large number of attributes or classes with a variable attribute domains such as the Solar Flare dataset or the Nursery dataset, the results of our classifier are largely better than the two others. In the other cases, such as for Balance Scale Weight dataset, our classifier performs slightly worse than the NBC but usually better than NBSPC.
- *Q_PCC analyzes*: For each dataset, besides to the *PCC*, the three classifiers compute the *Q_PCC*, results are illustrated in Table 5 and schematically represented by Fig. 1. From Table 5, we can see that the *Q_PCC* of NPC is always higher than the *Q_PCC* of NBC even for Balance or Cancer datasets where the *PCC* of NBC is greater than the *PCC* of NPC. These results can be explained by two reasons: first, for misclassified instances NPC is closer to predict the real class than the NBC. Second, in the perfect case ($\beta = 1$ and $\alpha = 1$), following NPC (resp. NBC) and Eq. (8) (resp. Eq. (1)), $\pi(a_j|c_i)$ (resp. $p(a_j|c_i)$) is computed as

$$\pi(a_j|c_i) = \frac{\sum_{k=1\cdots|\mathrm{Tr}|, A=a_j, c=c_i} \pi(I_k|c_i)}{N_{a_j}}$$

where $\pi(I_k|c_i) = 1$, if the class is c_i and 0 otherwise and N_{a_j} is the number of instances where the attribute value is a_j.

$$p(a_j|c_i) = \frac{N_{a_j c_i}}{N_{c_i}},$$

where $N_{a_j c_i}$ is the number of instances where the value of the attribute is a_j and the class is c_i and N_{c_i} is the number of instances where the class is c_i.

Thus, if we have a great number of classes the number of instances where the class is c_i will be a reduced number, then $p(a_j|c_i)$ will be a high number and so $d(i)$ and consequently both $D(i)$ and *Q_PCC* will have a low value. This fact is well shown by the low value of *Q_PCC* (see Table 5) of NBC for Balance and Nursery datasets where the number of classes is relatively high (respectively, 3 and 5).

Concerning NBSPC, in the precise case the possibility distributions of any attribute value given any class is the same and is equal to $(1/|\mathrm{Tr}|)$, this fact justifies the low value of both the *PCC* and the *Q_PCC* of this classifier.

Author's personal copy

Table 5
Experimental *PCC* and *Q_PCC*.

Database	NPC		NBSPC		NBC	
	PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
Wisconsin Breast Cancer	91	91.19	44.71	53.9	89.8	90.38
Congressional Voting	84.03	85.9	35.6	38.7	86.04	84.97
Balance Scale Weight	83.3	85.5	11.11	33.33	85.16	77.69
Solar Flare	82.79	82.84	29.03	35.96	78.74	80.25
Nursery	76.6	94.01	31.63	43.51	73.59	73.06

Table 6
Experimental *PCC/Q_PCC* (10%).

		Wisconsin Breast Cancer		Balance Scale Weight		Congressional Voting		Solar Flare		Nursery	
		PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
10%	NPC	85.83	85.71	83.11	86.13	90.9	91.20	82.7	82.81	76.01	78.9
	NBSPC	64.28	65.29	25.39	43.65	39.2	40.40	39.78	41.22	45.66	50.20
30%	NPC	82.18	84.09	82.22	84.15	89.29	90.80	81.72	82.30	76.03	76.81
	NBSPC	65.71	66.83	26.98	44.70	40.40	45.60	44.28	48.19	49.9	52.40
50%	NPC	80.61	82.85	80.35	81.60	89.292	89.42	80.09	81.10	74.08	76.20
	NBSPC	73.19	79.45	59.60	60.74	49.9	51.2	49.46	50.56	51.2	53.4

6.4. Results of the imprecise case

Experimentations in the imprecise case are carried in three steps. In each step, we fix the degree of imprecision (10%, 30%, 50%) relative to the training set and we vary the degree of imprecision (10%, 30%, 50%) relative to attributes of these instances. Tables 6–8 show the results of *PCC* and *Q_PCC* of both NBSPC and NPC, we can see that NPC presents good results and outperforms NBSPC, while the performance of NPC decreases with the increase of the percentage of imprecision. However, what is attracting is the variation of *PCC* and *Q_PCC* of NBSPC, both measures increase with the increase of imprecision. This can be explained by the fact that the Borgelt et al. method is unsuitable in the perfect case (low *PCC* and *Q_PCC* for imprecision degrees equal to 10%). However, from Table 9, which shows the results of a total imprecise training set (the two imprecision degrees are equal to 100%), we remark that the performance of NBSPC is stopped when the imprecision degrees are important and both the *PCC* and *Q_PCC* decrease. Thus NBSPC is relatively suited when imprecision has a moderate value.

6.5. Results of the imperfect case

As we have mentioned, to each dataset, two degrees of imperfection are assigned. Our experiments are performed in three steps, for each step, we fix the degree of imperfection relative to the number of imperfect instances in the dataset and we vary the degree of imperfection relative to the number of imperfect attributes in each instance (10%, 30%, 50%).

Tables 10–12 present different results carried out by testing the NPC on imperfect training and testing sets relative to the five previously mentioned databases.

As with the perfect case, the results show that NPC deals with imperfect instances as good as with perfect ones. However, the *PCC* decreases with the increase of the imperfection, this ratio remains high in average. For instance, if we analyze the results relative to Congressional Voting database (where the two degrees of imperfection are equal to 50%), we remark that the *PCC* remains higher than 78%.

Author's personal copy

Table 7
Experimental *PCC/Q_PCC* (30%).

		Wisconsin Breast Cancer		Balance Scale Weight		Congressional Voting		Solar Flare		Nursery	
		PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
10%	NPC	84.28	84.73	83.02	84.66	90.20	90.80	82.30	82.39	75.66	75.96
	NBSPC	75.33	78.57	26.98	35.79	42.80	45.30	42.78	44.66	47.77	53.20
30%	NPC	81.12	83.92	81.17	84.09	88.24	89.62	80.64	81.22	75.01	75.08
	NBSPC	78.57	77.76	63.49	65.81	59.20	63.50	45.66	49.91	56.81	57.02
50%	NPC	79.66	80.02	79.39	81.52	87.19	87.91	80.38	81.19	73.33	74.01
	NBSPC	78.85	79.04	65.07	66.64	60.20	64.90	60.21	60.9	59.2	62.52

Table 8
Experimental *PCC/Q_PCC* (50%).

		Wisconsin Breast Cancer		Balance Scale Weight		Congressional Voting		Solar Flare		Nursery	
		PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
10%	NPC	83.92	83.96	80.95	83.81	89.30	90.20	81.02	81.09	74.16	76.15
	NBSPC	63.28	64.29	49.2	61.52	46.30	48.90	43.02	46.15	50.23	54.02
30%	NPC	80.01	82.60	79.19	82.02	87.02	87.03	77.27	79.40	68.40	70.8
	NBSPC	65.71	66.83	66.49	70.31	60.20	61.30	59.6	63.18	60.02	66.04
50%	NPC	78.14	79.66	77.87	80.35	86.12	88.15	75.17	75.82	65.67	69.90
	NBSPC	60.01	64.64	73.99	74.60	66.20	68.30	61.21	70.76	66.66	69.48

Table 9
Experimental *PCC* and *Q_PCC* (in the worst case).

Database	NPC		NBSPC	
	PCC (%)	*Q_PCC* (%)	*PCC*(%)	*Q_PCC* (%)
Wisconsin Breast Cancer	75.71	80.52	55.71	63.90
Congressional Voting	64.15	66.76	60.20	63.29
Balance Scale Weight	72.71	79.06	60.31	65.95
Solar Flare	73.64	75.22	43.01	50.98
Nursery	64.66	70.64	33.33	43.72

Table 10
Experimental *PCC/Q_PCC* (10%).

	Wisconsin Breast Cancer		Balance Scale Weight		Congressional Voting		Solar Flare		Nursery	
	PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
10%	83.3	83.6	83	83.21	90.88	90.91	82.55	82.6	86	82.66
30%	82.53	83.67	82.55	82.56	89	89.87	82.48	82.5	85	82.46
50%	81.2	82.53	80.71	80.75	87	87.24	80.64	80.65	84.02	80.16

Author's personal copy

Table 11
Experimental *PCC/Q_PCC* (30%).

	Wisconsin Breast Cancer		Balance Scale Weight		Congressional Voting		Solar Flare		Nursery	
	PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
10%	82.53	85.31	81.52	81.53	88	88.35	81.87	81.87	80.8	83.05
30%	81.3	82.73	81.4	81.42	87.9	88.37	80.01	80.01	79.72	83.05
50%	80.34	80.95	76.83	76.59	85	85.13	78.44	78.04	74.21	78.67

Table 12
Experimental *PCC/Q_PCC* (50%).

	Wisconsin Breast Cancer		Balance Scale Weight		Congressional Voting		Solar Flare		Nursery	
	PCC (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)	*PCC* (%)	*Q_PCC* (%)
10%	80.95	83.81	79.88	79.89	79.89	86	80.79	82.66	74	83.51
30%	79.47	82.36	78.95	78.96	79.73	85	79.7	81.41	72.66	78.53
50%	77.15	77.77	74.6	74.6	78.96	84.02	74.19	74.6	68.66	74.86

Table 13
Experimental *PCC* and *Q_PCC* (in the worst case).

Dataset	*PCC* (%)	*Q_PCC* (%)
Wisconsin Breast Cancer	62.4	72.78
Congressional Voting	73.71	74.28
Balance Scale Weight	68.09	68.15
Solar Flare	71.56	72.64
Nursery	61.39	70.43

As we have studied the perfect case, Table 13 shows the results of the worst case where the two imperfection degrees are equal to 100%. We remark that while the *PCC* has considerably decreased by comparison to the *PCC* of the perfect case, it is still greater than 60%, which can be considered good regarding the imperfection rate.

In the same manner, the *Q_PCC* follows the track of the *PCC*, but it is higher than this latter, 68%. This is can be justified as in perfect case, by the fact that misclassified instances have a great chance, with NPC to be well classified.

Results in the three mentioned cases (perfect, imprecise, imperfect) clearly demonstrate that NPC is a good classifier and well adapted to deal with any type of datasets.

7. Conclusion and future work

In this paper, we have addressed the problem of constructing a classifier from imperfect data. The proposed approach named, *naïve possibilistic network classifier* is a possibilistic counterpart of the naïve Bayes one, a very simple kind of Bayesian networks, competitive with the state-of-the-art classifiers such as C4.5 [16]. Our classifier is built from a training set where instances can be characterized by imperfect attributes and classes. Furthermore, the proposed classifier is able to classify new instances that can have imperfect attributes. Imperfection is expressed by a random normalized set of possibility distributions, respectively, defined on possible attribute and class values.

Experiments show that our classifier maintains the robustness and computational complexity of naïve Bayes, and at the same time often, displays better accuracy. In the imperfect case, both the *PCC* and the *Q_PCC* decrease but their values stay high regarding the assigned imperfection degrees.

To improve the accuracy of our classifier we have proposed two direct extensions of it, which maintain its basic structure, namely *selective possibilistic network classifier* and *semi-naïve possibilistic classifier*.

In the future, we plan to make the proposed classifier able to handle even quantitative or qualitative data then, it will be very interesting to extend our method to build a classifier from ordinal training sets, that is to say attributes or

Author's personal copy

classes are expressed through an order between their, respectively, possible values. In this case, properties of possibility theory can be very well used since this theory is well suited to process this kind of data. Also, throughout this paper we have made the assumption that all attributes have finite number of values. Then, it will be very interesting to extend our method to handle continuous variables.

Appendix A

Before giving the formal proof, of the induction of the classification model of NPC, let us give some basic notions on propagation in possibilistic networks. In PN [1], nodes communicate via two kinds of messages, called λ messages and μ messages circulating, respectively, from children to parents and vice versa.

Each node A in the PN is characterized by its conditional possibility measure based on the total evidence, denoted by $\mathrm{Bel}(A)$. This value depends on the λ-value and μ-value relative to the node A (denoted, respectively, by $\lambda(A)$ and $\mu(A)$).

The values of $\lambda(A)$ and $\mu(A)$ are computed from the λ-messages received from children of A and μ-messages received from its parents (denoted, respectively, by $\lambda_{Y_j}(A)$ and $\mu_A(U_i)$ where $\lambda_{Y_j}(A)$ is the message that A receives from its child Y_j and $\mu_A(U_i)$ is the message that A receives from its parent U_i).

PN have two settings depending on the definition of the possibilistic conditioning: the product-based one and the min-based one then, to each setting a classification model is distinguished.

In what follows we give the proof relative to the deduction of the classification model in the product-based setting. In the min-based one the classification model can be inferred in analogical manner from the propagation process in this setting.

$\forall a \in D_A$, the current conditional possibility measure of A based on the total evidence e (over nodes of the network) is defined by

$$Bel(A) = \Pi(a|e) = \alpha \cdot \lambda(a) \cdot \mu(a) \text{ where } \alpha \text{ is a normalization factor.}$$

The λ value $\forall a \in D_A$ is defined by $\lambda(a) = \Pi(a|e_A^-) = \lambda_A(a) \cdot \prod_{j=1}^{m} \lambda_{Y_j}(a)$, where e_A^- denotes the evidence localized in nodes situated below A in the network, m is the number of children of A and λ_A is the local evidence related to A such that

$$\lambda_A(a) = \begin{cases} 0 & \text{if } A \text{ is not instantiated to } a, \\ 1 & \text{otherwise } (A \text{ instantiated to } a \text{ or } A \text{ is not instantiated}). \end{cases}$$

The μ value $\forall a \in D_A$ is defined by $\mu(a) = \Pi(a|e_A^+) = \max_u \Pi(a|u) \cdot \prod_{i=1}^{n} \mu_A(u_i)$ where e_A^+ denotes the evidence localized in nodes situated above A in the network and n is the number of parents of A.

The λ message from A to its parent U_i ($i \in \{1, \ldots, n\}$) when $U_i = u_i$ is defined by

$$\lambda_A(u_i) = \Pi(e_{U_i A}^- | u_i) = \max_a \lambda(a) \left[\max_{u_k : k \neq i} \Pi(a|u) \cdot \prod_{k \neq i} \mu_A(u_k) \right].$$

If each node has a unique parent this message is simplified to

$$\lambda_A(u_i) = \max_a (\lambda(a) \cdot \Pi(a|u)).$$

The μ message from A to its child Y_j when $A = a$ is defined by

$$\mu_{Y_j}(a) = \lambda_A(a) \prod_{i=1,\ldots,m, i \neq j} \lambda_{Y_i}(a) \cdot \mu(a).$$

Thus, when considering the particular case of NPC in the product-based setting, a given instance is assigned to the most plausible class c^* defined as

$$c^* = \arg\max_{c_i} \prod_{j=1}^{m} \Pi(a_j | c_i).$$

Author's personal copy

In fact, in NPC defined in product-based setting, the λ value can be simplified as follows: $\lambda(u) = \pi(u|Y)$, where y denotes the child node, u denotes the root node of the network and Y denotes the set of children of u. As NPC is a rooted graph, λ can be written as $\lambda(u) = \lambda_U(u) * \prod_{j=1}^{m} \lambda_{Y_j}(u)$, where $\lambda_U(u)$ denotes the local evidence of u. Since U is not instantiated, then $\lambda_U(u) = 1 => \lambda(u) = \prod_{j=1}^{m} \lambda_{Y_j}(u)$,

$$\lambda_{Y_j}(u) = \max_{y_j} \lambda(y_j)\Pi(y_j|u),$$

$\lambda(y_j) = \lambda_{Y_j}(y_j) \prod_{j=1}^{m} \lambda_{Y_j}(y_j)$, since y_j has no children, then $\prod_{j=1}^{m} \lambda_{Y_j}(y_j)$ can be ignored

$$=> \lambda(y_j) = \lambda_{Y_j}(y_j).$$

If Y_j is instantiated to y_j or Y_j is not instantiated, then, $\lambda(y_j) = 1$

$$=> \lambda(y_j) * \Pi(y_j|u) = \Pi(y_j|u)$$

else $\lambda(y_j) = 0 => \lambda(y_j) * \Pi(y_j|u) = 0$

$$=> \max_{y_j} \lambda(y_j) * \Pi(y_j|u) = \Pi(y_j|u) => \pi(u|Y) = \prod_{j=1}^{m} \Pi(y_j|u)$$

The class c^*, of a new instance (I_k) is the one corresponding to the maximum of $\pi(c_i|I_k)$ (where c_i and I_k correspond, respectively, to u and Y) and we have

$$c^* = \arg\max_{c_i} \prod_{j=1}^{m} \Pi(a_j|c_i).$$

References

[1] N. Ben Amor, S. Benferhat, D. Dubois, K. Mellouli, H. Prade, A theoretical framework for possibilistic independence in a weakly ordered setting, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (2002) 117–155.

[2] N. Ben Amor, S. Benferhat, D. Dubois, K. Mellouli, H. Prade, Anytime propagation algorithm for min-based possibilistic graphs, Soft Computing, A fusion of Foundations, Methodologies and Applications, Vol. 8, Springer, Berlin, 2003, pp. 150–161.

[3] N. Ben Amor, S. Benferhat, Z. Elouedi, Qualitative classification and evaluation, in: Possibilistic Decision Trees, FUZZ-IEEE 2, 2004, pp. 653–657.

[4] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1996.

[5] C. Borgelt, R. Kruse, Efficient maximum projection of database-induced multivariate possibility distributions, in: Proc. 7th IEEE Internat. Conf. on Fuzzy Systems, USA, 1998, pp. 663–668.

[6] C. Borgelt, J. Gebhardt, A naïve Bayes style possibilistic classifier, in: Proc. 7th European Congress on Intelligent Techniques and Soft Computing, Germany, 1999, pp. 556–565.

[7] T.M. Cover, P.E. Hart, Nearest neighbour pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.

[8] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero–one loss, Machine Learning 29 (1997) 103–130.

[9] D. Dubois, H. Prade, Possibility Theory, Plenum Press, New York, 1988.

[10] D. Dubois, H. Prade, Possibility Theory: An Approach to computerized Processing of Uncertainty, Plenum Press, New York, 1994.

[11] D. Dubois, H. Prade, Possibility theory: qualitative and quantitative aspects, in: D. Gabbay, P. Smets (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol. 1, Kluwer Academic Publishers, Dordrecht, 1998, pp. 169–226.

[12] D. Dubois, H. Prade, Possibility theory in information fusion, in: Proc. 3rd Internat. Conf. on Information Fusion, Paris, 2000, pp. 21–34.

[13] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[14] Z. Elouedi, P. Vannorenberghe, Inférence statistique et modèle de croyances transférables, in: LFA Toulouse, France, 2006, pp. 61–68.

[15] P. Fonck, A comparative study of possibilistic conditional independence and lack of interaction, International Journal of Approximate Reasoning 16 (1997) 149–171.

[16] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine Learning 29 (1997) 131–161.

[17] D. Geiger, D. Heckerman, Knowledge representation and inference in similarity networks and Bayesian multinets, Artificial Intelligence 82 (1996) 45–74.

[18] D. Grossman, P. Dominigos, Learning Bayesian maximizing conditional likelihood, in: Proc. on Machine Learning, Canada, 2004, pp. 46–57.

[19] I. Inza, P. Larrañaga, R. Etxeberria, B. Sierra, Feature subset selection by Bayesian network-based optimization, Artificial Intelligence 123 (2000) 157–184.

[20] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence Journal, Special Issue on Relevance 97 (1997) 273–324.

[21] I. Kononenko, Semi-naïve Bayesian classifier, in: Proc. 6th European Working Session on Learning, 1991, pp. 206–219.

Author's personal copy

[22] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: Proc. AAAI-92, 1992, pp. 223–228.
[23] P. Langley, S. Sage, Induction of selective Bayesian classifiers, in: Proc. 10th Conf. on Uncertainty in Artificial Intelligence UAI-94, 1994, pp. 399–406.
[24] X. Li, Augmented naive Bayesian classifiers for mixed-mode data, in: IEEE Workshop on Automatic Speech Recognition and Understanding, Virgin Islands, 2003.
[25] H. Liu, H. Motoda, Feature Selection, Knowledge Discovery and Data Mining, Kluwer, Boston, 1998.
[26] P.M. Murphy, D.W. Aha, Uci repository of machine learning databases ⟨http://www.ics.uci.edu/mlearn⟩, 1996.
[27] M.J. Pazzani, Searching for dependencies in Bayesian classifiers, in: Artificial Intelligence and Statistics IV, Lecture Notes in Statistics, Springer, New York, 1997, pp. 239–248.
[28] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmman, San Francisco, CA, 1988, pp. 221–226.
[29] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[30] M. Sahami, Learning limited dependence Bayesian classifiers, in: Proc. 2nd Int. Conf. On Knowledge Discovery and Data Mining KDD'96, Portland, OR, USA, 1996, pp. 335–338.
[31] G. Shafer, A mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.
[32] R.R. Yager, A general approach to the fusion of imprecise information, International Journal of Intelligent Systems 12 (1997) 1–29.
[33] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[34] L.A. Zadeh, Fuzzy sets as a basic for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.