Naive credal classifier 2: an extension of naive Bayes for delivering robust classifications

Giorgio Corani and Marco Zaffalon

***Abstract*—Naive credal classifier 2 (NCC2) extends naive Bayes in order to deliver more robust classifications. NCC2 is based on a set of prior densities rather than on a single prior; as a consequence, when faced with instances whose classification is prior-dependent (and therefore might not be reliable), it returns a set of classes (we call this an *indeterminate classification*) instead of a single class. Moreover, NCC2 introduces a very general and flexible treatment of missing data, which, under certain circumstances, can also lead to indeterminate classifications. In this case, indeterminacy can be regarded as a way to preserve reliability despite the information hidden by missing values. We call *hard*-to-classify the instances classified indeterminately by NCC2. Extensive empirical evaluations show that naive Bayes' accuracy drops considerably on the hard-to-classify instances identified by NCC2, and that on the other hand, NCC2 has high set-accuracy (the proportion of times that the actual class is contained in the set of returned classes) when it is indeterminate.**

I. Introduction

The *naive Bayes classifier* (NBC) is a well-known classifier, based on the (naive) assumption that the attributes are independent given the class. Despite its simple design, NBC has been recognized as surprisingly effective in real-world applications, even if the naive assumption is violated [1]. *Naive credal classifier 2* (NCC2) extends NBC to *imprecise probabilities* [2] in order to robustly deal with (a) the specification of the prior density and (b) the treatment of missing data.

In fact, on small data sets, the classifications issued by NBC may be quite sensitive to the specification of the prior, which represents the investigator beliefs before analyzing the data, about the distribution of the classes and the features. This is not a problem if the prior can be carefully elicited; but it can well be a problem when the prior is not modeling the investigator's beliefs properly because in this case the classifier risks drawing arbitrary conclusions. Unfortunately, it is just in the field of data mining that one of those cases is frequently originated: this happens when one assumes (almost) no domain knowledge and wants to learn (almost) entirely from data. In this case the prior should model the investigator's state of prior ignorance. In the Bayesian framework this is done by so-called *non-informative* priors; here the problem is that it can be argued that non-informative priors model *indifference* rather than ignorance (e.g., see [2, Section 5.5.1]) and therefore can lead to arbitrariness in the issued classifications.

Following Walley, NCC2 drops the idea of specifying a unique, precise, prior to model a state of ignorance and does this instead by *a set* of precise priors;[1] such a set is referred to as prior *credal set*. The set of prior densities is then turned into a set of posterior densities by element-wise application of Bayes rule. The classification is eventually issued by returning all the classes that are *non-dominated*[2] within the set of posterior densities. Whence, NCC2 returns a set of classes when faced with instances whose classification is prior-dependent; it issues hence a weaker and yet arguably more robust classification than NBC. We can regard the weaker answers of NCC2 as a logical consequence of weakening the traditional assumption that regards non-informative priors as models of prior ignorance. We call *determinate* the classifications made of a single class, and the others *indeterminate* or *set-valued*.

The second assumption that NCC2 weakens is related to the treatment of missing data. We recall the common definition of *missingness process* (MP) as a process that takes in input the complete data and outputs the incomplete ones (i.e., a data set for which some values can be missing), which are what we observe. A non-selective MP, which generates *missing at random* (MAR) data [3], is referred to as a MAR MP. An important point is that it is usually not possible to conclude whether the MP is MAR by analyzing the available, incomplete data; and as we said, in data mining one often assumes the data are the only source of information. As a consequence, the assumption that missing data are generated by a MAR MP is strong and not always justified.

NBC deals with missing data by ignoring them both at learning and test stage. Technically, however, such an approach is justified only if missing data are generated by a MAR MP. NCC2 is instead able to deal also with missing data generated by an unknown MP; in this case, it adopts a conservative approach, considering all the completions of missing values as possible. As a consequence, missing data, when treated as generated by an unknown MP, lead also sometimes to indeterminate classifications; this preserves reliability, despite the information hidden by missing values. NCC2 ignores instead, like NBC, missing data generated by a MAR MP. NCC2 deals flexibly with MAR and non-MAR missing data: it can handle mixed situations, i.e., data sets in which part of the features are subject to a MAR MP and part to an unknown MP. Moreover, the list of features subject

Giorgio Corani and Marco Zaffalon are with IDSIA, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, CH-6928 Manno (Lugano), Switzerland (e-mails: {giorgio,zaffalon}@idsia.ch).

[1] More precisely, we focus on what Walley calls *near*-ignorance.

[2] Class c_i is said to dominate c_j if *for all* the posteriors densities it holds that the probability of c_i is larger than that of c_j.

to the MAR and to the unknown MP can change between training and test stage.

The idea of extending NBC to a set of prior densities to model prior ignorance has been pioneered by *naive credal classifier* of [4], which however did not offer a method of general validity to deal with missing data. NCC2 extends the former naive credal classifier by a very general and flexible treatment of missing data, while keeping the original benefits of naive credal classifier on the front of prior ignorance. NCC2 is derived in Sect. II and III.

In Sect. IV we extensively compare NCC2 and NBC on 18 data sets; a key point of our empirical comparison is that we evaluate separately the accuracy achieved by NBC on the instances classified determinately and indeterminately by NCC2. In fact, we regard as *hard* the instances classified indeterminately by NCC2, and we expect NBC to decrease its accuracy on these instances. The experiments show that the accuracy of NBC drops consistently from the instances classified determinately NCC2 to those classified indeterminately. Moreover, we see that, especially when missing data are generated by a non-MAR MP, NBC can classify confidently some instances (i.e., computing a high probability for the returned class), over which NCC2 returns indeterminate classifications, and over which the accuracy achieved by NBC is bad indeed. On the other hand, set-valued classification do preserve the reliability of NCC2: they are accurate, as they contain the actual class with high frequency and deliver a non-negligible informative content, as they lead to drop up to 2/3 of the original set of the classes of the problem. Finally, when NCC2 is determinate, it is as accurate as NBC.

II. Setup

A. Notation and assumptions

In this paper, complete yet not observable variables are referred to as *latent*, while incomplete (but observable) variables are referred to as *manifest*. A given manifest value is identical to the corresponding latent one, unless the latent value has been turned into missing by the MP; in this case, the manifest value is actually the symbol of missing value.

In the following, i indexes a given unit (i.e., a certain row) of the data set: the *learning set* (or *training set*) is made up of the units for which $1 \leq i \leq N$, while the unit to classify (not belonging to the learning set) is indexed by $M := N+1$. A set of units to classify is referred to as *test set*.

In a classification problem there are typically *class variables* and *attribute variables*. We denote: (i) the latent *class variable* as C_i, and we assume that it is always observed; (ii) the latent attribute variables affected by an unknown MP (i.e., to be conservatively modelled as non-MAR) as $A_{i1},\dots,A_{ik}$; (iii) the latent attributes affected by a MAR MP as $\hat{A}_{i1}$, …, $\hat{A}_{ir}$. The two MPs are assumed to be independent of each other and their coarsening behavior is allowed to vary with different units, i.e., they are *not* assumed to be identically distributed.

For all i, C_i takes generic value c_i in the finite set $\mathcal{C}$, called *set of latent classes*, while A_{ij} ($\hat{A}_{il}$) take generic values a_j ($\hat{a}_l$) in the finite sets $\mathcal{A}_j$ ($\hat{\mathcal{A}}_l$), called *sets of latent attributes*.

We define the following groups of latent variables: $X_i := (A_{i1},\dots,A_{ik})$, $D_i := (C_i, X_i)$, and $\hat{X}_i := (\hat{A}_{i1},\dots,\hat{A}_{ir})$. We then extend such grouped variables to span the whole training set, instead than just the i-th unit, defining the vector $\boldsymbol{C} := (C_1,\dots,C_N)$ and the matrices $\boldsymbol{X} := (X_1,\dots,X_N)$, $\hat{\boldsymbol{X}} := (\hat{X}_1,\dots,\hat{X}_N)$, $\boldsymbol{D} := (D_1,\dots,D_N)$. The same grouped variables but with a "+" superscript, include also data of the M-th unit (i.e., the instance to be classified): $\boldsymbol{C}^{+} := (C_1,\dots,C_M)$, $\boldsymbol{X}^{+} := (X_1,\dots,X_M)$, $\hat{\boldsymbol{X}}^{+} := (\hat{X}_1,\dots,\hat{X}_M)$,$\boldsymbol{D}^{+} := (D_1,\dots,D_M)$. Let also define $\boldsymbol{D}^{-} := (\boldsymbol{C}, \boldsymbol{X}^{+})$.

Observe that realizations of the random matrix $(\boldsymbol{D}^{+}, \hat{\boldsymbol{X}}^{+})$ represent the possible complete data sets (i.e., the data sets before the missingness process), and that realizations of $(\boldsymbol{D}, \hat{\boldsymbol{X}})$ represent the possible complete learning sets, while those of $(X_M, \hat{X}_M)$ represent the possible complete units to classify.

To complete the notation regarding latent variables, we assume that the generic latent unit $(d, \hat{x}) \in \mathcal{D} \times \hat{\mathcal{X}}$ is generated in *independently and identically distributed* way according to the *aleatory probability* (or *chance*) $\vartheta_{(d,\hat{x})}$. The vector of such chances is denoted by ϑ, which belongs to Θ, i.e., a (non-empty) subset of the unitary $|\mathcal{D} \times \hat{\mathcal{X}}|$-dimensional simplex. Let θ denote the random variable of which ϑ is a generic value.

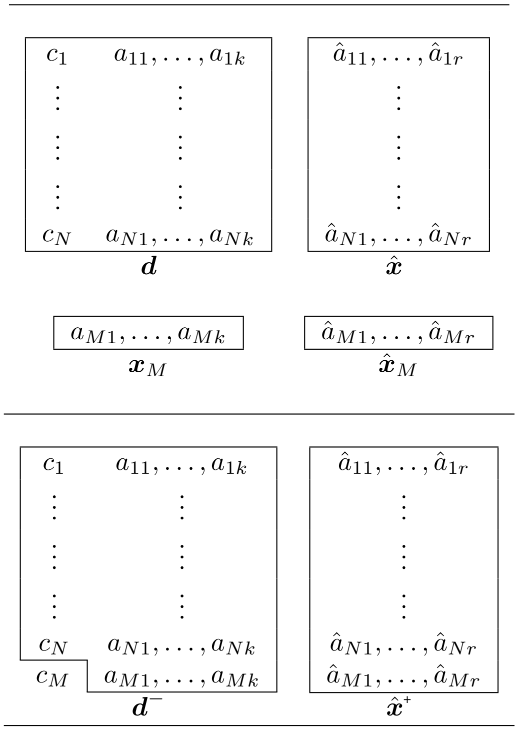


Fig. 1. Graphical representation of some vectors of latent variables. Rows $1,\dots,N$ constitute the training set, while the M-th unit is a new instance to be classified.

As for the manifest variables, we assume that we either observe a value precisely, or we do not observe it at all.

Manifest variables are denoted by the letter O followed by the latent variable they refer to, written as a subscript. We define hence the following manifest variables: $\boldsymbol{O} := O_{\boldsymbol{D}} = (O_1, \ldots, O_N)$, $\boldsymbol{O}^{+} := O_{\boldsymbol{D}^{+}} = (O_1, \ldots, O_M)$, $\boldsymbol{O}^{-} := O_{\boldsymbol{D}^{-}} = (O_1, \ldots, O_N, X_M)$, $\hat{\boldsymbol{O}} := O_{\hat{\boldsymbol{X}}} = (\hat{O}_1, \ldots, \hat{O}_N)$, $\hat{\boldsymbol{O}}^{+} := O_{\hat{\boldsymbol{X}}^{+}} = (\hat{O}_1, \ldots, \hat{O}_M)$.

B. *Classification with imprecise probabilities and conservative inference rule*

The goal of classification is to predict the class of the M-th unit, given the previous units $(1, \ldots, N)$ and the values of the M-th attribute variables.

To this extent, a traditional probabilistic classifier outputs what it deems to be the optimal prediction: i.e., the class with the highest probability (in the case of 0-1 loss function) on the basis of a uniquely computed posterior density. In the imprecise setting, however, the optimality criterion has to be extended to manage a set of posterior densities (derived from a set of priors and a set of likelihoods), instead of a single posterior; in particular, according to [2, Section 3.9.2], the optimality criterion in the imprecise setting prescribes to return the *non-dominated* classes. The definition of dominance is as follows: class c_i dominates c_j if for all the posteriors densities, the probability of c_i is greater than that of c_j; c_j is non-dominated if no class dominates c_j. The non-dominated classes are then detected via pairwise comparisons. Observe that, as a result of the uncertainty arising from both prior specification and non-MAR missing values, there can be several non-dominated classes; in this case, the classifier returns an indeterminate (or set-valued) classification.

A key point is that non-dominated classes are *incomparable*;[3] this means that there is no information in the model that allows us to rank them. In other words, credal classifiers are models that allow us to drop the dominated classes, as sub-optimal, and to express our indecision about the optimal class by yielding the remaining set of non-dominated classes.

In the setup of this paper, the test of dominance can be re-written as follows: c'' is dominated by c' if and only if it holds that

$$1 < \min_{\boldsymbol{x}_M \in \boldsymbol{o}_M} \min_{\boldsymbol{d} \in \boldsymbol{o}} \inf_{p(\theta) \in \mathcal{P}(\theta)} \frac{p(c'_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+})}{p(c''_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+})}. \qquad (1)$$

Actually, Equation (1) is the general form of the test of dominance for any classifier based on *conservative inference rule* (CIR), as presented in [5]; CIR is a conditioning rule (i.e., a rule for computing conditional expected values) that generalizes the traditional conditioning; it assumes that prior beliefs are dealt with via a credal set $\mathcal{P}(\theta)$ and it accounts for data sets in which the missingness process is MAR for some variables (the term $\hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+}$ refers indeed to the missing data of MAR features in the training set), and unknown for some others. Moreover, CIR is able to manage variables whose MP is MAR in learning and unknown in testing, or vice versa.

CIR can be regarded as unifying two rules [5]: a *conservative learning rule*, which prescribes how to learn the classifier from an incomplete training set, and a *conservative updating rule*, which prescribes how to classify a novel instance that contains missing values. Such a distinction is made clear by two distinct optimization loops of Equation (1); the middle optimization loop ($\min_{\boldsymbol{d} \in \boldsymbol{o}}$) realizes the conservative learning rule, by prescribing to loop on the completions of the non-MAR part of the learning set, i.e., $\boldsymbol{d} \in \boldsymbol{o}$, while the outer minimum implements the conservative updating rule, prescribing to loop on the replacements for the non-MAR missing values of the unit to classify. The inner loop, which minimizes over the prior credal set, is common to both learning and updating rules.

III. Introducing NCC2

In this section, we describe how NCC2 specializes the test of Equation (1) to the case of naive classification. We assume initially that only the observation of variables affected by the MAR MP contain missing data, while the observations of the variables affected by the unknown MP are complete; such a simplification will be removed in the next sections. Note that variables affected by the MAR MP and by the unknown MP will be treated separately in the equations.

The probability for class c_M, given the learning set and the values of the attribute variables in the current instance to be classified, can be re-written as follows:

$$p(c_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+}) \propto$$
$$\int_{\Theta} p(\boldsymbol{d}, \hat{\boldsymbol{x}} \in \hat{\boldsymbol{o}} | \vartheta) p(\vartheta) p(c_M, x_M, \hat{x}_M \in \hat{o}_M | \vartheta) d\vartheta, \qquad (2)$$

where:

- $p(\boldsymbol{d}, \hat{\boldsymbol{x}} \in \hat{\boldsymbol{o}} | \vartheta)$ is the *likelihood function*;
- $p(\theta)$ denotes an imprecise prior density for θ; this means that in our setting $p(\theta)$ belongs to a non-empty set of precise prior densities for θ, referred to as *prior credal set*.

Assuming the naive hypothesis (i.e, the mutual independence of the latent attribute variables conditional on the class variable), the likelihood can be expressed as follows:

Lemma 1.

$$p(\boldsymbol{d}, \hat{\boldsymbol{x}} \in \hat{\boldsymbol{o}} | \vartheta) =$$
$$\prod_{c \in \mathcal{C}} \{\vartheta_c^{n(c)} [\prod_{j=1}^{k} \prod_{a_j \in \mathcal{A}_j} \vartheta_{a_j|c}^{n(a_j,c)}][\prod_{l=1}^{r} \prod_{\hat{a}_l \in \hat{\mathcal{A}}_l} \vartheta_{\hat{a}_l|c}^{n(\hat{a}_l,c)}]\}. \qquad (3)$$

Here ϑ_c denotes the chance of $(C_i = c_i)$; $\vartheta_{a_{ij}|c}$ and $\vartheta_{\hat{a}_{il}|c}$ denote the chances of $(A_{ij} = a_j | C_i = c_i)$ and $(\hat{A}_{il} = \hat{a}_l | C_i = c_i)$, respectively. Moreover, $n(c)$ resp. $n(a_j, c)$ denote the number of occurrences of c resp. of joint occurrences of (a_j, c) in $\boldsymbol{d}$, and $n(\hat{a}_l, c)$ denotes the number of joint occurrences of $(\hat{a}_l, c)$ in the learning set after dropping the units with missing values of $\hat{A}_l$. Hence, missing data generated by the MAR MP are ignored in the computation of the likelihood. Technically, the likelihood

[3] If we exclude the classes that are non-dominated due to indifference rather than incomparability, and that constitute a very special case in the imprecise setting.

function of Lemma 1 has the same functional form as a product of Dirichlet densities.

With similar arguments to those used with Lemma 1, and assuming that the MAR attribute variables have been reordered so as to index the non-missing ones in the instance to classify from 1 to $r' \leq r$, we obtain:

Lemma 2.

$$p(c_M, x_M, \hat{x}_M \in \hat{o}_M | \vartheta) = \vartheta_{c_M} \prod_{j=1}^{k} \vartheta_{a_{Mj}|c_M} \prod_{l=1}^{r'} \vartheta_{\hat{a}_{Ml}|c_M}. \quad (4)$$

Note that restricting the second product between $l = 1$ and $l = r'$ prevents the inclusion in the expression of the attributes that are missing in the unit to classify.

A. Imprecise prior (prior credal set)

The remaining term in (2) is $p(\vartheta)$, i.e., the prior. We define it as a product of Dirichlet densities, so that it is *conjugate* to the likelihood, as follows:

$$\begin{aligned} p(\vartheta|s,t)d\vartheta \propto \prod_{c \in \mathcal{C}} \{ \vartheta_c^{st(c)-1} d\vartheta_C \cdot \\ \cdot [\prod_{j=1}^{k} \prod_{a_j \in \mathcal{A}_j} \vartheta_{a_j|c}^{st(a_j,c)-1} d\vartheta_{A_j|c}] \cdot \\ \cdot [\prod_{l=1}^{r} \prod_{\hat{a}_l \in \hat{\mathcal{A}}_l} \vartheta_{\hat{a}_l|c}^{st(\hat{a}_l,c)-1} d\vartheta_{\hat{A}_l|c}] \}. \end{aligned} \quad (5)$$

The real hyperparameter s can be regarded as the size of the *hypothetical sample*, in the common interpretation of conjugate Bayesian priors as additional sample units; the real hyperparameter $t(\cdot)$ can instead be regarded as the proportion of units of the given type (e.g., $t(c)$ is the proportion of units with class c) in the hypothetical sample. We adopt $s := 1$ for the experiments of Section IV.

Now, remember that we want the prior to be imprecise, i.e., to be a set of priors. We define the set by imposing a system of constraints on the t-hyperparameters that resemble the structural constraints of the observed frequencies $n(\cdot)$: in particular, $\sum_{c \in \mathcal{C}} t(c) = 1$, $\sum_{a_j \in \mathcal{A}_j} t(a_j, c) = t(c)$, $\sum_{\hat{a}_l \in \hat{\mathcal{A}}_l} t(\hat{a}_l, c) = t(c)$. We moreover impose the conditions $t(a_j, c) > 0$, $t(\hat{a}_l, c) > 0$. The credal set $\mathcal{P}(\theta)$ is defined as the set of all the precise priors that satisfy these constraints. The construction of $\mathcal{P}(\theta)$ is similar to the approach implemented by Walley's' *imprecise Dirichlet model* [6].

B. Probability of the Next Class

The tools introduced so far lead us to the following result.

Theorem 3. *Consider Expression* (2). *It holds that:*

$$\begin{aligned} p(c_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+}, s, t) = \frac{n(c_M) + st(c_M)}{(N+s)} \cdot \\ \cdot \prod_{j=1}^{k} \frac{n(a_{Mj}, c_M) + st(a_{Mj}, c_M)}{n(c_M) + st(c_M)} \cdot \\ \cdot \prod_{l=1}^{r'} \frac{n(\hat{a}_{Ml}, c_M) + st(\hat{a}_{Ml}, c_M)}{n_l(c_M) + st(c_M)}, \end{aligned} \quad (6)$$

where

- $\frac{n(c_M)+st(c_M)}{N+s} = p(c_M | \boldsymbol{d}, \hat{\boldsymbol{x}} \in \hat{\boldsymbol{o}}, s, t)$;
- $\frac{n(a_{Mj},c_M)+st(a_{Mj},c_M)}{n(c_M)+st(c_M)} = p(a_{Mj} | c_M, \boldsymbol{d}, \hat{\boldsymbol{x}} \in \hat{\boldsymbol{o}}, s, t)$;
- $\frac{n(\hat{a}_{Ml},c_M)+st(\hat{a}_{Ml},c_M)}{n_l(c_M)+st(c_M)} = p(\hat{a}_{Ml} | c_M, \boldsymbol{d}, \hat{\boldsymbol{x}} \in \hat{\boldsymbol{o}}, s, t)$;
- $n_l(c_M) := \sum_{\hat{a}_l \in \hat{\mathcal{A}}_l} n(\hat{a}_l, c_M)$.

Note that MAR attributes that are missing in the unit to be classified do *not* affect the posterior probabilities of the class.

C. Dominance tests with an imprecise prior

Let us now address the computation of the inner optimization problem in (1), under the choice of the imprecise prior made in Section III-A.

Lemma 4. *Consider the problem* $\inf_{p(\theta) \in \mathcal{P}(\theta)} p(c'_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+}) / p(c''_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+})$, *with the set of prior densities described in Section III-A, and the probabilities in the function to optimize (also called* objective function *in the following) defined as in* (6). *Such a problem is equivalent to the following:*

$$\begin{aligned} \inf_{0 < t(c_{M''}) < 1} \{ [\frac{n(c''_M) + st(c''_M)}{n(c'_M) + s - t(c''_M)}]^{k-1} \cdot \\ \prod_{j=1}^{k} \frac{n(a_{Mj}, c'_M)}{n(a_{Mj}, c''_M) + st(c''_M)} \cdot \prod_{l=1}^{r'} [\frac{n_l(c''_M) + st(c''_M)}{n_l(c'_M) + s - t(c''_M)} \cdot \\ \cdot \frac{n(\hat{a}_{Ml}, c'_M)}{n(\hat{a}_{Ml}, c''_M) + st(c''_M)}] \} =: \inf_{0 < t(c_{M''}) < 1} h(t(c_{M''})). \end{aligned} \quad (7)$$

See [7, pp. 613-614] for an efficient procedure (not reported here for lack of space) to solve Problem (7) exactly.

D. Incomplete, non-MAR, learning set

In case the learning data produced by the unknown MP are incomplete, the problem to be solved is

$$\min_{\boldsymbol{d} \in \boldsymbol{o}} \inf_{p(\theta) \in \mathcal{P}(\theta)} p(c'_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+}) / p(c''_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+}). \quad (8)$$

To solve Problem (8) we have to select the realization of the non-MAR part of the learning set, among the possible realizations $\boldsymbol{d} \in \boldsymbol{o}$ consistent with our observations, which minimizes the probability ratio.

Let us denote $\underline{n}(a_{Mj}, c'_M) := \min_{\boldsymbol{d} \in \boldsymbol{o}} n(a_{Mj}, c'_M)$ and $\overline{n}(a_{Mj}, c''_M) := \max_{\boldsymbol{d} \in \boldsymbol{o}} n(a_{Mj}, c''_M)$.

Lemma 5. *Problem* (8) *is solved as follows:*

- *in Problem (7), rename* $n(a_{Mj}, c'_M) := \underline{n}(a_{Mj}, c'_M)$ *and* $n(a_{Mj}, c''_M) := \overline{n}(a_{Mj}, c''_M)$*;*
- *solve the obtained instance of Problem (7).*

E. Incomplete, non-MAR, unit to classify

Finally, we consider the case when the unit to classify is missing some of the attribute variables subject to the unknown MP. We need to address the following problem:

$$\min_{\boldsymbol{x}_M \in \boldsymbol{o}_M} \min_{\boldsymbol{d} \in \boldsymbol{o}} \inf_{p(\theta) \in \mathcal{P}(\theta)} \frac{p(c'_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+})}{p(c''_M | \boldsymbol{d}^{-}, \hat{\boldsymbol{x}}^{+} \in \hat{\boldsymbol{o}}^{+})}. \quad (9)$$

Problem (9) can be trivially solved as follows: (i) considering all the possible realizations $\boldsymbol{x}_M$ of the non-MAR part of the

Algorithm 1 Learning

```
list the attributes affected by a MAR MP or by an unknown MP on the learning set;
compute on the learning set the counts $n(\hat{a}_l, c)$, $n_l(c)$ for MAR attributes and the counts $\underline{n}(a_j, c)$, $\overline{n}(a_j, c)$, $n(c)$ for non-MAR attributes.
```

Algorithm 2 Dominance test between two classes (c', c'')

```
list the attributes affected by a MAR MP or by an unknown MP in the instance;
for $\boldsymbol{x}_M \in \boldsymbol{o}_M$ (i.e., for each possible realizations of the non-MAR part of the unit to classify): do
    assume $\boldsymbol{x}_M$ to be the realization of the non-MAR part of the instance;
    solve Problem (8);
    if the computed solution is smaller than 1 then
        $c''$ is not dominated by $c'$.
    end if
end for
if after having tried every $\boldsymbol{x}_M \in \boldsymbol{o}_M$, no solution greater than 1 has been found then
    $c'$ dominates $c''$.
    /*alternatively, the minimum can be computed via an efficient polynomial-time procedure.*/
end if
```

instance (this is accomplished by considering all the possible replacements for the missing values); (ii) for each $\boldsymbol{x}_M \in \boldsymbol{o}_M$, assuming $\boldsymbol{x}_M$ to be the realization of the non-MAR part of the unit to classify, and then solving Problem (8); (iii) if the computed solution is smaller than or equal to 1, c''_M is not dominated by c'_M (and the computation can be interrupted); instead, if the solution is greater than 1 for each $\boldsymbol{x}_M$, c''_M is dominated by c'_M.

Although this procedure leads to the exact solution, it takes exponential time due to the number of the replacements of missing values in the instance to classify. We have hence designed [7, pp. 610-611] a more efficient procedure (not reported here for lack of space), which is still exact and solves Problem (9) in polynomial-time. The NCC2 procedures are summarized by Algorithms 1 and 2.

IV. Experiments

The goal of the experimental part is to compare NBC and NCC2 under a variety of different settings and data sets, to answer questions such as: is NCC2 truly able to isolate instances which are hard to classify for NBC? How does NBC behave on the instances which are classified determinately and indeterminately by NCC2? How frequently does NCC2 become indeterminate?

We start with a simple example referring to a single data set (Section IV-A), in order to make the reader familiar with the methodology adopted for comparing NBC and NCC2. Then we present exhaustive experiments performed on 18 UCI data sets; all data sets are complete, i.e., they do not contain missing data. In fact, missing data will be generated artificially. In particular, in (Section IV-B), we produce artificial missingness on the 18 data sets via a MAR MP, and we define as MAR all the features into NCC2; in Section IV-C, we produce artificial missingness on the 18 data sets via a non-MAR MP, and we define as non-MAR all the features into NCC2. Alternative settings (for instance, treating as non-MAR missing data which are actually MAR) are of interest but we do not present them in this paper for lack of space.

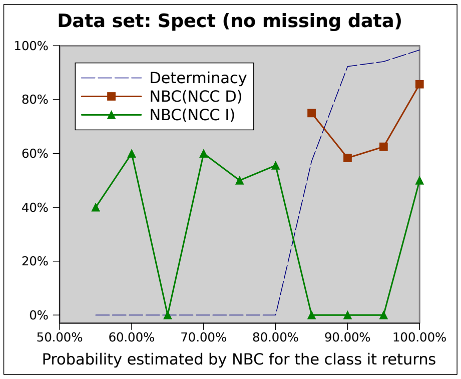


Fig. 2. Relationship between the posterior probabilities computed by NBC and the output of NCC2 on the spect data set.

A. An illustrative example

Firstly, we focus on a simple case study: the spect data set; it is made of 2 classes and 267 instances, and does not contain missing data. We split the data set into a training set of 67 instances, and a test set of 200 instances; we choose hence to work with a small training set, in order emphasize the effect of the prior. In this example prior ignorance is the only possible source of indeterminacy for NCC2. Moreover, as spect has only two classes, we can also clearly see what are the instance that are deemed doubtful by NBC, i.e., those classified with probability close to 50%.

For each instance of the test set we consider four pieces of information: the actual class, the class returned by NBC and its associated probability, and whether or not the instance has been classified by NCC2 in a determinate way.

We consider three indicators of performance: (a) *determinacy* of NCC2, i.e., the percentage of instances for which NCC2 returns a unique class; the accuracy achieved by NBC on the instances classified determinately and indeterminately by NCC2, which we denote respectively as (b) NBC(NCC D) and (c) NBC(NCC I).

The instances are then partitioned into subsets as follows: instances for which NBC estimates a probability in the range 50–55%, instances for which NBC estimates a probability in the range 55–60%, and so on (i.e., we use a step of 5% in probability to define the subsets). On each subset of instances, we compute the three mentioned indicators.

The results are shown in Figure 2. We see that there is a positive association between higher posterior probabilities computed by NBC and higher determinacy; indeed the choice of the prior is less likely to change the classification outcome when the probability computed by NBC for the most probable class increases. The output of NCC2 is indeterminate for all the instances as long as the probability estimated by NBC for the returned class is lower than 80%. In other words, here NCC2 deems that there is very little information about those instances in the learning set and suspends the judgment. Remarkably, such caution is justified as on those instances NBC is just guessing almost at random, or even worse. In fact, NBC classifies confidently a number of instances (assigning probabilities that go as up as 80%), but in practice such a confidence is not justified.

Moving on to greater probabilities, we see that the determinacy of NCC2 rises substantially when the probability estimated by NBC exceeds 80%; in this region NCC2 returns a mix of determinate and indeterminate classifications, and the drop of accuracy between NBC(NCC D) and NBC(NCC I) is large at any level of posterior probability. In fact, NCC2 returns indeterminate classification also on a non-negligible number of instances classified very confidently (for instance, with probability higher than 85%) by NBC, and over which the accuracy of NBC is bad indeed.

Summing up, NCC2 does not suspend the judgment only on instances that are deemed doubtful by NBC, i.e., those whose probability is for instance less than 55%. Moreover, while on the spect data set NCC2 is fully indeterminate on the instances on which NBC is doubtful, this is not always the case. For instance, the same analysis performed on the spambase data set (2 classes, 1300 training instances) shows that about 40% of instances in the range 50–55% of probability are classified determinately by NCC2 (while the behavior on the remaining ranges of probabilities followed a pattern similar to the one shown in Fig. 2).

B. *Results with MAR missingness*

Now, we analyze the behavior of NCC2 when there are MAR missing data. On each data set, we generate artificial missingness by turning each observation, apart from the classes, into missing with probability 5%; such a missingness process is actually MAR. On each data set, we then perform 100 training/testing experiments; the split between training and testing is 50%-50%. Into NCC2, we define all features variables as MAR both in training and test.

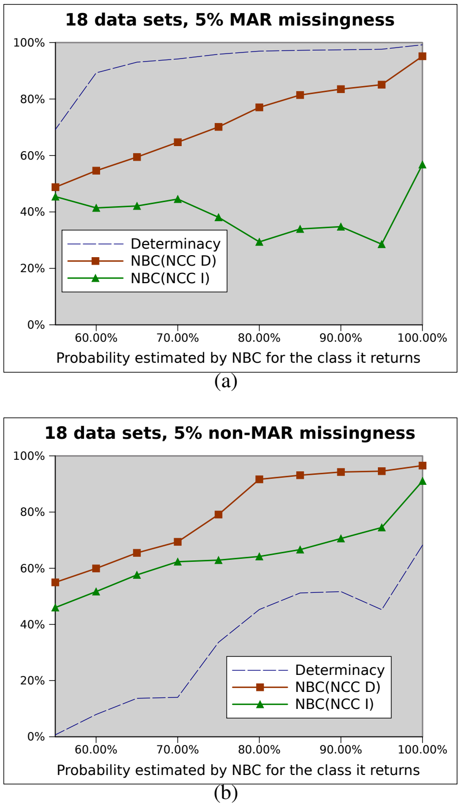
To summarize the results, we report in the following several indicators, averaged over the 18 data sets. The average accuracy of NBC is 82%; however, this is the result of 85% accuracy on the instances classified determinately by NCC2, and 36% accuracy on the instances classified indeterminately by NCC2. The average determinacy of NCC2 is 95%, i.e., in 5% of cases NCC2 returns more than one class. We see hence that prior ignorance and MAR missing data lead to little indeterminacy; yet, the accuracy of NBC is bad indeed on the instances over which NCC2 returns set-valued classifications.

When NCC2 is indeterminate, it returns about one third of the classes of the data set; in 84% of cases such an output contains the actual class.[4] Hence set-valued classifications do *preserve* the reliability of NCC2.

On each data set, moreover, we have found that the inequality NBC(NCC D) $>$ NBC(NCC I) is verified.

Eventually, we jointly analyze the predictions issued on all the data sets, via the methodology shown in the spect example; we focus in particular on the case when NBC is confident, i.e., on the instances with probability for the returned class greater than or equal to 55%. The results are shown in Figure 3a. Figure 3a shows a clear drop of accuracy of NBC, for the same level of posterior probability of the predicted class, from the instances classified in a determinate way by NCC2 to the others; the drop is especially striking on the instances classified very confidently by NBC, when the computed probability is for instance larger than 70%.

Moreover, there is a positive association between higher posterior probabilities computed by NBC and higher determinacy; both patterns were already present in Figure 2. The determinacy shown in Figure 3a is much higher than in the spect example, because the analysis now includes also several large data sets, such as kr-kp or spambase.



(a)

(b)

Fig. 3. Relationship between the posterior probabilities computed by NBC and the output of NCC2.

[4] Such indicators, referring to indeterminate classifications, are computed with reference to data sets having more than two classes; in fact, on data sets with two classes, NCC2 returns, when indeterminate, 100% of the classes with 100% set-accuracy.

C. Results with non-MAR missingness

We now evaluate the performance of NCC2 when the MP is non-MAR. Into NCC2, we set all features as non-MAR both in training and testing. In order to generate non-MAR missingness, we design the following non-MAR MP: (i) split the categorical values of each feature variable into two halves (remember that features variables have been already discretized); (ii) for each feature variable, turn into missing, with probability 5%, the observations falling into the first half of values, both on training and test. As in the previous section, we consider 18 UCI data sets and we perform 100 training/testing experiments on each data set.

NBC achieves an average accuracy of 82%; its performance hence is apparently not affected by the non-MAR missing data. In fact, the non-MAR MP we consider is very simple; more sophisticated non-MAR MPs can instead largely reduce the accuracy of NBC [7].

The determinacy of NCC2 is about 52%, i.e., much lower than in the MAR setup. In fact, the non-MAR setting leads to a much larger amount of instances that are isolated as difficult ones. On average the NBC accuracy drops from 95% on the instances determinately classified by NCC2, to 69% on those indeterminately classified; the drop between NBC(NCC D) and NBC(NCC I) is hence remarkable, though smaller than under the MAR setting. This can be partly explained by considering that NCC2 is more conservative than necessary in our non-MAR setup. In fact, it considers all the values of set $\mathcal{A}_j$ as possible replacement for the missing values of feature A_j; instead, given how the MP works, the possible replacements are only half the values contained in $\mathcal{A}_j$. This leads NCC2 to some unnecessary caution; the results we present are hence conservative in this respect.

Also in the non-MAR setup, the inequality NBC(NCC D) $>$ NBC(NCC I) is verified on each single data set.

When NCC2 is indeterminate, it returns about two thirds of the classes; in 96% of the cases such group contains the actual class;[5] this further confirms that set-valued classification preserve the reliability of NCC2.

Moreover, Figure 3b shows a clear drop of accuracy of NBC, for the same level of posterior probability of the predicted class, between NBC(NCC D) and NBC(NCC I); interestingly, NCC2 suspends the judgment frequently also on the instances classified by NBC with probability greater than 80%, and which are nevertheless doubtful indeed. In fact, missing data treated as non-MAR can lead frequently to indeterminate classifications even if the probability computed by NBC for the returned class is high: the reason is that a replacement for missing data especially unfavorable for the class predicted by NBC can well change the outcome of the classification with respect to the output of NBC, which instead marginalizes out the missing feature variable.

V. Conclusions

NCC2 generalizes naive Bayes by weakening some of its assumptions that play a critical role for robustness. Thanks to imprecise probabilities, NCC2 can return indeterminate classifications, made up by more classes, on instances which it recognizes as hard to classify, because of uncertainty arising from missing data, or because the outcome of the classification is dependent on the choice of the prior. This allows NCC2 to yield robust classifications also in cases of poor prior knowledge and poor knowledge about the MP, which are a commonplace in data mining.

NCC2 becomes indeterminate also on instances which are dealt with confidently by NBC (which might for instance return a probability larger than 70%), and over which nevertheless NBC achieves bad accuracy. In fact, the overall performance of NBC can be regarded as the average of a good accuracy on a subset of instances determinately classified by NCC2, and a much worse accuracy on the instances indeterminately classified by NCC2.

Remarkably, the set of classes returned by NCC2 when it is indeterminate contains the actual class with very high probability; hence, set-valued classifications do preserve the reliability of NCC2. Moreover, set-valued classifications are valuable, as they are informative (unlikely classes are dropped), and invite the domain expert to avoid overconfident statements.

JNCC2, i.e., the Java implementation of NCC2, is released under the GNU GPL license; it is available from http://www.idsia.ch/~giorgio/jncc2.html.

An extended version of this paper, including proofs and more algorithmic and experimental details, has been published [7] after the first submission to this conference.

Acknowledgments

The Authors gratefully acknowledge partial support by the Swiss NSF grants 200021-113820/1 and 200020-116674/1, and by the Hasler Foundation (Hasler Stiftung) grant 2233.

References

[1] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2/3, pp. 103–130, 1997.

[2] P. Walley, *Statistical Reasoning with Imprecise Probabilities*. New York: Chapman and Hall, 1991.

[3] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. New York: Wiley, 1987.

[4] M. Zaffalon, "Statistical inference of the naive credal classifier," in *ISIPTA '01: Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications* (G. de Cooman, T. L. Fine, and T. Seidenfeld, eds.), (The Netherlands), pp. 384–393, Shaker, 2001.

[5] M. Zaffalon, "Conservative rules for predictive inference with incomplete data," in *ISIPTA '05: Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications* (F. G. Cozman, R. Nau, and T. Seidenfeld, eds.), (Manno, Switzerland), pp. 406–415, SIPTA, 2005.

[6] P. Walley, "Inferences from multinomial data: learning about a bag of marbles," *J. R. Statist. Soc. B*, vol. 58, no. 1, pp. 3–57, 1996.

[7] G. Corani and M. Zaffalon, "Learning reliable classifiers from small or incomplete data sets: The naive credal classifier 2," *Journal of Machine Learning Research*, vol. 9, pp. 581–621, 2008.

[5] We recall that the two last indicators are computed only on data sets with more than two classes.