Learning Reliable Classifiers From Small or Incomplete Data Sets: The Naive Credal Classifier 2.

Article *in* Journal of Machine Learning Research · April 2008

DOI: 10.1145/1390681.1390701 · Source: DBLP

Giorgio Corani
Dalle Molle Institute for Artificial Intelligence

Journal of Machine Learning Research 9 (2008) 581-621 Submitted 1/07; Revised 2/08; Published 4/08

Learning Reliable Classifiers From Small or Incomplete Data Sets: The Naive Credal Classifier 2

Giorgio Corani GIORGIO@IDSIA.CH
Marco Zaffalon ZAFFALON@IDSIA.CH
IDSIA
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale
CH-6928 Manno (Lugano), Switzerland

Editor: Charles Elkan

Abstract

In this paper, the naive credal classifier, which is a set-valued counterpart of naive Bayes, is extended to a general and flexible treatment of incomplete data, yielding a new classifier called *naive credal classifier 2* (NCC2). The new classifier delivers classifications that are reliable even in the presence of small sample sizes and missing values. Extensive empirical evaluations show that, by issuing set-valued classifications, NCC2 is able to isolate and properly deal with instances that are hard to classify (on which naive Bayes accuracy drops considerably), and to perform as well as naive Bayes on the other instances. The experiments point to a general problem: they show that with missing values, empirical evaluations may not reliably estimate the accuracy of a traditional classifier, such as naive Bayes. This phenomenon adds even more value to the robust approach to classification implemented by NCC2.

Keywords: naive Bayes, naive credal classifier, imprecise probabilities, missing values, conservative inference rule, missing at random

1. Introduction

Is it possible to draw credible conclusions about a domain only looking at some data produced within the domain itself?

The answer to this question appears to be related to the problem of modeling *ignorance*.[1] In fact, there are at least two kinds of ignorance involved in the process of learning from data. The first is *prior ignorance* about the domain, as we are assuming that data are our only source of information. The second is ignorance arising from missing values, as data are often incomplete; in this case, ignorance is about the process that originates the missing values: that is, the *missingness process*. So, in principle, we should model both ignorances properly in order to deliver credible conclusions.

Let us consider *pattern classification*, which is the focus of this paper. Considering *Bayesian classifiers*, we see that prior ignorance is modeled in common practice by so-called non-informative *prior densities* (or just *priors*, for short). But such an approach can lead, especially when the learning set is *small*, to the known problem of prior-dependent classifications, whose reliability is

[1] When we use the word "ignorance" in this paper, we actually mean a condition of *near-ignorance*. Indeed, full ignorance is not compatible with learning, as it is well known (e.g., see Section 7.3.7 of Walley, 1991, Section 2.3 of Zaffalon, 2005b).

questionable. This appears to indicate that non-informative priors do not model prior ignorance satisfactorily.

A more objective-minded[2] model of prior ignorance has been proposed through a classifier called *naive credal classifier* (NCC), see Zaffalon (2001), which is an extension of *naive Bayes classifier* (NBC) to *imprecise probabilities* (Walley, 1991). NCC models prior ignorance by a *set* of prior densities (also called prior *credal set*), which is turned into a set of posteriors by element-wise application of Bayes' rule. The classification is eventually issued by returning all the classes that are *non-dominated* by any other class according to the posterior credal set, where class c_i is said to dominate c_j if *for all* the posteriors it holds that the probability of c_i is larger than that of c_j. This makes NCC naturally issue *set-valued* classifications (i.e., classifications made by more than one class) when faced with instances that are hard to classify, due to a combination of prior ignorance and poor information *about those specific instances* in the learning set. The shift of paradigm based on set-valued classifications allows NCC to deliver robust classifications in spite of small learning sets. NCC has indeed shown excellent accuracy in real-world case studies (Zaffalon, 2005a; Zaffalon et al., 2003), thus demonstrating the usefulness, for classification purposes, of modeling prior ignorance via a credal set. In the following, set-valued classifications are also called *indeterminate*. *Determinate* classifications correspond instead to the set being a singleton, and hence to the case usually considered by more traditional classifiers. Similarly, we say that a classifier is determinate when it outputs a single class and indeterminate otherwise.

As for the ignorance arising from missing data, we can think of the missingness process (MP) as a process that takes in input the complete data, which we cannot usually observe, and outputs the incomplete data, which we do observe. If data are our only source of information, we are ignorant about the MP because it is usually not possible to learn how it operates, from the observed, incomplete data.

In common practice, missing values are often *ignored*; this entails the idea that the MP is non-selective in producing them, or, in other words, that it is a *missing at random* (MAR) process (Little and Rubin, 1987; Jaeger, 2005). However, if one is ignorant about the MP, assuming MAR cannot be regarded as an objective-minded approach, as is well documented, for instance, by Manski (2003).

In its original formulation, NCC introduced also an initial attempt to deal with ignorance about the MP. The idea was to model ignorance about it by using a set of *likelihoods*: a likelihood per each complete learning set consistent with the incomplete one. A similar avenue was also implemented by *robust Bayes classifier* (Ramoni and Sebastiani, 2001).[3] These approaches are indeed valuable, but have two problems: (i) they implicitly still assume MAR for the missing values in the instance to classify, thus creating a peculiar asymmetry between learning and test set that is not of general validity in applications; (ii) they may well be too conservative, because for some feature variables one might know that the missingness is MAR, and they do not allow this information to be incorporated in the model. Furthermore, their treatment of missing values rests on intuitive arguments rather than on a principled derivation.

[2] Although we base our results on Walley's theory, which is a *subjective* theory of probability, we sometimes use the terminology "objective-minded." We do so to stress that using (very) weak assumptions leads to results that are much more determined by the data than by our prior beliefs (intended in a loose way, not only as prior probabilities), and in this sense are more objective. Yet, in the paper we deliberately avoid using the word "objective" alone, just because it is improper within the considered theory.

[3] When used with the so-called *strong dominance score*.

By this paper we extend NCC to a very general and flexible treatment of incomplete data, both in learning and testing. We call the resulting classifier *naive credal classifier 2* (NCC2), in order to emphasize the advancement made to deal with incomplete data, while keeping the original benefits of NCC on the front of prior ignorance.

By NCC2, it is possible to declare that some (possibly all or none) of the feature variables are subject to a MAR process, and the remaining ones are automatically assumed to be subject to an MP that is unknown to us. Remarkably, the set of feature variables subject to a MAR MP can be chosen differently from the learning to the test set. This is a key characteristic of NCC2: in fact, if the MP is unknown, it may well change its behavior from unit to unit for all we know (i.e., it may not be *identically distributed*), and we should act accordingly.

The development of NCC2 is based on a recently derived so-called *conservative inference rule* (CIR) to compute (imprecise) conditional expectations with incomplete data (Zaffalon, 2005b). After giving some notation and briefly recalling CIR in Section 2, we derive NCC2 in Section 3 by specializing CIR to the case of naive classification. In the end we obtain procedures to learn NCC2 and to do classifications with it that do not involve approximations and are computationally fast. (The software which implements NCC2 is released as open source; more details are provided in Section 3.5.)

Next, we concentrate on empirical evaluations: in Section 4 we analyze the behavior of NCC2 from a number of angles and on a number of publicly available data sets. The analysis turns out to be particularly meaningful when we compare NCC2 with its precise-probability counterpart, that is, naive Bayes. We do this by evaluating the accuracy of NBC on the instances of the test set where NCC2 issues a determinate classification separately from those where it does not. In fact, NCC2 is indeterminate on an instance when it deems that there is not enough knowledge in the learning set to make a determinate classification reliably; NBC, on the other hand, issues a determinate classification on such an instance (as well as on any other). Therefore we expect NBC to have different behaviors on the two kinds of instances isolated by NCC2. And this is indeed the case: the experiments show that NBC undergoes a major drop in accuracy moving from the instances classified in a determinate way by NCC2 to the indeterminate ones. The drop is observed on every data set, with no exception.

It is important to realize that such a drop points out a key question: the usual way to measure the performance of a classifier, that is, its predictive accuracy, which is an average over all the instances of the test set, may not help uncover a possible bad performance of the classifier on a subset of the test instances. These instances are precisely those that are hard to classify and that NCC2 isolates by delivering set-valued classifications.

But set-valued classifications help NCC2 to do more than just isolating the hard instances, they enable it to cope effectively with them: in fact, we show that set-valued classifications are often informative, as they usually lead to drop some unlikely classes; and that the measured *set-based* accuracy of NCC2 (i.e., the proportion of times the true class is contained in the output set) is often similar to the accuracy obtained on the instances classified in a determinate way.

At this point we should say that the mentioned experiments have been carried out with a variety of settings, obtained considering both MAR processes and non-MAR ones, and the mentioned outcomes are been confirmed over all of them (although sometimes this is due more to prior ignorance and some others more to the missing data).

To make our results stronger, we have also investigated whether NBC could take advantage of the posterior probabilities it computes in order to deal more successfully with the hard instances.

We have considered such posterior probabilities again by separating the cases where NCC2 is determinate from the others, and by comparing those probabilities with the measured accuracy that NBC actually achieves on the data. What we show is that the NBC probabilities are (also very) unreliable on the instances that are hard to classify, as isolated by NCC2, and definitely more unreliable than on the remaining instances. In other words, we observe another kind of drop that now is related to the quality of the posterior probabilities computed by NBC.

Overall, we show that NBC may well be too optimistic in dealing with small data sets and missing data, thus yielding unreliable predictions. It is useful to recall that NBC is known to be very robust to missing data. Therefore, it is not unlikely that the optimism on the front of missing data is even greater with more complex classifiers. This point appears to be worth of serious consideration on its own.

At the same time, and in contrast with naive Bayes, our experiments show that NCC2 may sometimes be too pessimistic (i.e., conservative) especially when dealing with missing data. This happens because by construction NCC2 implicitly considers the worst possible MP to have acted on the non-MAR part of the data, and in some cases this hypothesis may be too far from the MP that has actually produced the missing values.

In most of our experiments, for instance, we have deliberately used very simple MPs, and this has favored some excess of caution to show up. Yet, we have also considered an illustrative example of a more elaborated MP, in Section 4.6. In this case we show that the indeterminacy of NCC2 is fully justified, the NBC being completely unreliable in the area of indeterminacy. Even more important, we show that such an unreliability *cannot be uncovered by making empirical evaluations*: despite the predictive accuracy of NBC on a certain instance is measured properly by cross-validation, the actual accuracy on new instances of the same type can be significantly worse. This highlights the fact that modeling ignorance properly is important, even if there are data available for empirical evaluations. Indeed, in such an experiment NCC2 does not decrease its performance, nor do its empirical evaluations fail.

This is not to say that one should abuse assuming ignorance about the MP: when there are many missing values, especially in the instance to classify, one would obtain conclusions much too weak. This is shown in a setup where missingness increases in Section 4.5. Our view is that information about the MP should be incorporated in a model when available. In fact, we regard as an important research avenue the definition of classification models able to flexibly incorporate MP-related knowledge, which is usually not conveyed by the data at hand. With NCC2, incorporating knowledge is done by declaring that some variables are subject to a MAR MP, and we actually recommend doing so whenever possible; this has the potential, shown also experimentally in the above section, to yield strong enough conclusions in many cases. Eventually, in Section 4.7 we analyze the results obtained on the eucalyptus data sets; in fact, such results are quite peculiar and therefore worthy of a separate investigation. The conclusion we achieve in this case is that the use of coarsened rather than missing observations is another very effective means to incorporate knowledge, in case there is no support for declaring some variables as subject to a MAR MP.

2. Setup

In this paper, the variables that refer to complete data, which are in general not observable, are called *latent*, while those referring to incomplete data, which are the ones to which we have usually access, are called *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. Therefore, in a case where the data at hand are complete, the instances of the latent and the manifest variables coincide.

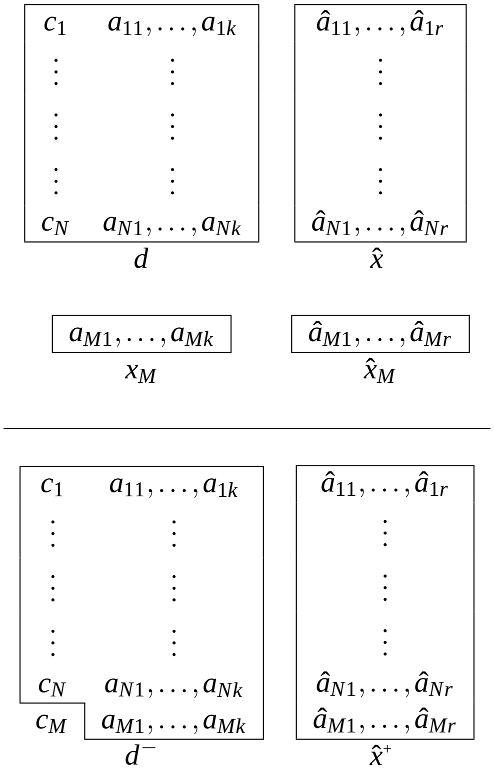

c_1 $a_{11},\dots,a_{1k}$
c_N $a_{N1},\dots,a_{Nk}$
d
$\hat{a}_{11},\dots,\hat{a}_{1r}$
$\hat{a}_{N1},\dots,\hat{a}_{Nr}$
$\hat{x}$
$a_{M1},\dots,a_{Mk}$
x_M
$\hat{a}_{M1},\dots,\hat{a}_{Mr}$
$\hat{x}_M$
c_1 $a_{11},\dots,a_{1k}$
c_N $a_{N1},\dots,a_{Nk}$
c_M $a_{M1},\dots,a_{Mk}$
d^-
$\hat{a}_{11},\dots,\hat{a}_{1r}$
$\hat{a}_{N1},\dots,\hat{a}_{Nr}$
$\hat{a}_{M1},\dots,\hat{a}_{Mr}$
$\hat{x}^+$

Figure 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.

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 modeled 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, that is, they are *not* assumed to be identically distributed.[4]

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

[4] See Zaffalon (2005b), Section 5, for a discussion about this point.

just the *i*-th unit, defining the vector $C := (C_1,\dots,C_N)$ and the matrices $X := (X_1,\dots,X_N)$, $\hat{X} := (\hat{X}_1,\dots,\hat{X}_N)$, $D := (D_1,\dots,D_N)$. The same grouped variables but with a "+" superscript, include also data of the *M*-th unit (i.e., the instance to be classified): $C^+ := (C_1,\dots,C_M)$, $X^+ := (X_1,\dots,X_M)$, $\hat{X}^+ := (\hat{X}_1,\dots,\hat{X}_M)$,$D^+ := (D_1,\dots,D_M)$. Let also define $D^- := (C,X^+)$.[5]

Observe that realizations of the random matrix $(D^+,\hat{X}^+)$ represent the possible complete data sets, which we cannot observe directly, and that realizations of $(D,\hat{X})$ 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* (IID) 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, that is, 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. Knowledge about θ is expressed by $p(\theta)$, which denotes an imprecise prior density for θ. This means that $p(\theta)$ is known to belong to a non-empty set $\mathcal{P}(\theta)$ of precise prior densities for θ. $\mathcal{P}(\theta)$ is referred to as *prior credal set*.

As for the manifest variables, we assume that we either observe a precise value, 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: $O := O_D = (O_1,\dots,O_N)$, $O^+ := O_{D^+} = (O_1,\dots,O_M)$, $O^- := O_{D^-} = (O_1,\dots,O_N,X_M)$, $\hat{O} := O_{\hat{X}} = (\hat{O}_1,\dots,\hat{O}_N)$, $\hat{O}^+ := O_{\hat{X}^+} = (\hat{O}_1,\dots,\hat{O}_M)$.

2.1 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 $(1,\dots,N)$ 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: that is, 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 Section 3.9.2 of Walley (1991), 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 computed posteriors densities, the posterior probability of c_i is greater than that of c_j; clearly, c_j is non-dominated if no class dominates c_j. The second procedure of Figure 2, based on pairwise comparison of classes, identifies the non-dominated classes. 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. Classifiers that issue set-valued classifications are called *credal classifiers* by Zaffalon (2002).

A key point is that non-dominated classes are *incomparable*;[6] 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.

[5] Since we assume the class variable to be always observed, C is a complete vector. Hence, it is possible to derive the NCC2 algorithms also by grouping C with $\hat{X}$.

[6] 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.

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

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

Actually, Equation (1) is the general form of the test of dominance for any classifier based on the conservative inference rule presented in Zaffalon (2005b); 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 $\hat{x}^+ \in \hat{o}^+$ refers indeed to the missing data of MAR feature variables 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. Equation (1) follows almost immediately from Theorem 4 in Zaffalon (2005b) after considering that for c'_M to dominate c''_M, it must hold that $p(c'_M | d^-, \hat{x}^+ \in \hat{o}^+) > p(c''_M | d^-, \hat{x}^+ \in \hat{o}^+)$ for all the possible posteriors, which we obtain considering any precise prior in the prior credal set, as well as any completion of the non-MAR missing values both in the sample and in the unit to classify.

CIR can be regarded as unifying two rules (Zaffalon, 2005b): 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 ($\min_{d \in o}$) realizes the conservative learning rule, by prescribing to loop on the completions of the non-MAR part of the learning set, that is, $d \in o$, 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.

NCC2 specializes the test of Equation (1) to the case of naive classification. In the following, we will move from the precise setting (corresponding in fact to naive Bayes) to NCC2 in four steps: in Section 3.1 we describe the precise setting (assuming hence that there is a single prior, and that there is a single likelihood as non-MAR data are complete); in Section 3.2 we extend the computation to manage a set of priors; in Sections 3.3 and 3.4 we finally relax the assumptions of completeness about non-MAR data in learning and testing respectively, thus managing a set of likelihoods and instances to classify.

3. Introducing NCC2

In this section, we introduce the NCC2 framework. In particular, we derive an expression that specializes the test of Equation (1) to the case of naive classification; such an expression realizes both the minimizations over possible priors and over possible unobserved values by distributing some pseudo-counts in a way that minimizes the ratio of the posterior probabilities of the two competing classes c'_M and c''_M.

Formal proofs of the findings derived in this section are provided in Appendix B.

3.1 Updating Precise Beliefs

In this section, we assume that a single prior is specified and that only the observations of the features affected by the MAR MP contain some missing data; the observations of the feature affected by the unknown MP, instead, do not contain any missing data. In practice, this corresponds to the naive Bayes setting, with the difference that we explicitly separate variables affected by the MAR MP and the unknown MP. In our setting, the naive hypothesis (i.e, the assumption of mutual independence of the latent attribute variables $A_{i1}, \ldots, A_{ik}, \hat{A}_{i1}, \ldots, \hat{A}_{ir}$ conditional on the class variable C_i) can be formalized as follows:

$$\vartheta_{(d_i,\hat{x}_i)} = \vartheta_{c_i} \prod_{j=1}^{k} \vartheta_{a_{ij}|c_i} \prod_{l=1}^{r} \vartheta_{\hat{a}_{il}|c_i} \quad \forall (d_i, \hat{x}_i) \in \mathcal{D} \times \hat{\mathcal{X}}, \tag{2}$$

where 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.

We focus on the precise probability $p(c_M | d^{-}, \hat{x}^{+} \in \hat{o}^{+})$. Let us consider the probability $p(c_M, d^{-}, \hat{x}^{+} \in \hat{o}^{+})$, which is proportional to the previous one. Write $p(c_M, d^{-}, \hat{x}^{+} \in \hat{o}^{+})$ as $\int_\Theta p(\vartheta) p(c_M, d^{-}, \hat{x}^{+} \in \hat{o}^{+} | \vartheta) d\vartheta$. Observe that $p(c_M, d^{-}, \hat{x}^{+} \in \hat{o}^{+} | \vartheta)$ is equal to $p(d, \hat{x} \in \hat{o} | \vartheta) p(c_M, x_M, \hat{x}_M \in \hat{o}_M | \vartheta)$ because of the IID assumption about the data generation mechanism. We obtain that

$$p(c_M | d^{-}, \hat{x}^{+} \in \hat{o}^{+}) \propto \int_\Theta p(\vartheta) p(d, \hat{x} \in \hat{o} | \vartheta) p(c_M, x_M, \hat{x}_M \in \hat{o}_M | \vartheta) d\vartheta. \tag{3}$$

The term $p(d, \hat{x} \in \hat{o} | \vartheta)$ in (3) is called *likelihood function*.

By sticking to the naive hypothesis, the likelihood can be expressed as follows:

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

Here $n(c)$ resp. $n(a_j, c)$ denote the number of occurrences of c resp. of joint occurrences of (a_j, c) in 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 $\hat{A}_l$. Technically, the likelihood function of Lemma 1 has the same functional form as a product of Dirichlet densities; in particular, the frequencies $n(\cdot)$ correspond to the Dirichlet hyperparameters usually denoted as $\alpha(\cdot) - 1$.

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

Lemma 2 $p(c_M, x_M, \hat{x}_M \in \hat{o}_M | \vartheta) = \vartheta_{c_M} \prod_{j=1}^{k} \vartheta_{a_{Mj}|c_M} \prod_{l=1}^{r'} \vartheta_{\hat{a}_{Ml}|c_M}$.

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.

3.1.1 IMPRECISE PRIOR (PRIOR CREDAL SET)

The remaining term in (3) is $p(\vartheta)$, that is, the prior. We define it so as to be *conjugate* to the likelihood, according to the following expression:

$$\begin{aligned} p(\vartheta | s, t) d\vartheta \propto \prod_{c \in \mathcal{C}} \{ \vartheta_c^{st(c)-1} d\vartheta_C [\prod_{j=1}^{k} \prod_{a_j \in \mathcal{A}_j} \vartheta_{a_j|c}^{st(a_j,c)-1} d\vartheta_{A_j|c}] \cdot \\ \cdot [\prod_{l=1}^{r} \prod_{\hat{a}_l \in \hat{\mathcal{A}}_l} \vartheta_{\hat{a}_l|c}^{st(\hat{a}_l,c)-1} d\vartheta_{\hat{A}_l|c}] \}, \end{aligned} \tag{4}$$

which is a product of Dirichlet densities. The density is named $p(\vartheta|s,t)$, so as to make it explicitly the hyperparameters on which it depends (here t denotes the vector of $t(\cdot)$-hyperparameters). In other words, the prior is defined by an expression that is similar to the likelihood function except that the frequencies $n(\cdot)$ are replaced everywhere by $st(\cdot)-1$. 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; Walley (1996) gives arguments to choose s in the interval $[1,2]$, and we will adopt $s:=1$ for the empirical experiments of Section 4. The real hyperparameter $t(\cdot)$ can instead be regarded as the proportion of units in which the variables in question take certain values (e.g., $t(c)$ is the proportion of units where the class variable is equal to c) in the hypothetical sample.

Now, remember that we want the prior to be imprecise, that is, 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 Walleys' *imprecise Dirichlet model* (Walley, 1996).

Informally, one could interpret $\mathcal{P}(\theta)$ in the following way: say that any single, *precise* Dirichlet prior, is a possible state of information about the process generating (latent) data. Then, our ignorance about this process is modeled by considering the set of all the states of information about the process, that is, by considering that all of them are actually possible.[7]

3.1.2 Probability of the Next Class

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

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

$$p(c_M|d^-,\hat{x}^+\in\hat{o}^+,s,t)=p(c_M|d,\hat{x}\in\hat{o},s,t)\prod_{j=1}^{k}p(a_{Mj}|c_M,d,\hat{x}\in\hat{o},s,t)\cdot$$
$$\cdot\prod_{l=1}^{r'}p(\hat{a}_{Ml}|c_M,d,\hat{x}\in\hat{o},s,t), \tag{5}$$

where

- $p(c_M|d,\hat{x}\in\hat{o},s,t):=[n(c_M)+st(c_M)]/(N+s)$*;*
- $p(a_{Mj}|c_M,d,\hat{x}\in\hat{o},s,t):=[n(a_{Mj},c_M)+st(a_{Mj},c_M)]/[n(c_M)+st(c_M)]$ *($j=1,\ldots,k$);*
- $p(\hat{a}_{Ml}|c_M,d,\hat{x}\in\hat{o},s,t):=[n(\hat{a}_{Ml},c_M)+st(\hat{a}_{Ml},c_M)]/[n_l(c_M)+st(c_M)]$ *($l=1,\ldots,r'$);*
- $n_l(c_M):=\sum_{\hat{a}_l\in\hat{\mathcal{A}}_l} n(\hat{a}_l,c_M)$.

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

[7] To be more precise, we do not consider all of them, only those with s fixed; this corresponds to fix the “strength” of our prior ignorance.

3.2 Credal 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 3.1.1.

Lemma 4 *Consider the problem* $\inf_{p(\theta)\in\mathcal{P}(\theta)} p(c'_M|d^-,\hat{x}^+\in\hat{o}^+)/p(c''_M|d^-,\hat{x}^+\in\hat{o}^+)$*, with the set of prior densities described in Section 3.1.1, and the probabilities in the function to optimize (also called* objective function *in the following) defined as in* (5)*. Such a problem is equivalent to the following:*

$$\begin{aligned}
&\inf_{0<t(c_{M''})<1}\{[\frac{n(c''_M)+st(c''_M)}{n(c'_M)+s-t(c''_M)}]^{k-1}\prod_{j=1}^{k}\frac{n(a_{Mj},c'_M)}{n(a_{Mj},c''_M)+st(c''_M)}\\
&\cdot\prod_{l=1}^{r'}[\frac{n_l(c''_M)+st(c''_M)}{n_l(c'_M)+s-t(c''_M)}\cdot\frac{n(\hat{a}_{Ml},c'_M)}{n(\hat{a}_{Ml},c''_M)+st(c''_M)}]\}\\
&=:\inf_{0<t(c_{M''})<1} h(t(c_{M''})).
\end{aligned}\tag{6}$$

The problem of finding the prior $p(\theta)\in\mathcal{P}(\theta)$ that minimizes the ratio of the posterior probabilities $p(c'_M|d^-,\hat{x}^+\in\hat{o}^+)/p(c''_M|d^-,\hat{x}^+\in\hat{o}^+)$ can be hence be solved by finding the value $0<t(c_{M''})<1$ that minimizes $h(t(c_{M''}))$, as follows.

Theorem 5 *The infimum of* $h(t(c_{M''}))$ *over* $(0,1)$ *is determined by the following procedure:*

- *if there is* j *such that* $n(a_{Mj},c'_M)=0$ *or* l *such that* $n(\hat{a}_{Ml},c'_M)=0$*,* $\inf h(t(c_{M''}))=0$*;*
- *if* $k=0$ *and* $r'=0$*,* $\inf h(t(c_{M''}))=h(1)$*;*
- *otherwise,* $h(t(c_{M''}))$ *can be shown to be convex over* $(0,1)$*; hence, it can be minimized by a convex optimization procedure (detailed in the proofs).*

3.3 Credal Dominance Test with an Incomplete, Non-MAR, Learning Set

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

$$\min_{d\in o}\inf_{p(\theta)\in\mathcal{P}(\theta)} p(c'_M|d^-,\hat{x}^+\in\hat{o}^+)/p(c''_M|d^-,\hat{x}^+\in\hat{o}^+). \tag{7}$$

Theorem 6 *The procedure of Theorem 5 solves Problem* (7) *after renaming* $n(a_{Mj},c'_M):=\underline{n}(a_{Mj},c'_M)$ *and* $n(a_{Mj},c''_M):=\overline{n}(a_{Mj},c''_M)$*, where* $\underline{n}(a_{Mj},c'_M):=\min_{d\in o} n(a_{Mj},c'_M)$ *and* $\overline{n}(a_{Mj},c''_M):=\max_{d\in o} n(a_{Mj},c''_M)$.

In other words, to solve Problem (7) we have to select the realization of the non-MAR part of the learning set, among the possible realizations $d\in o$ consistent with our observations, which minimizes the probability ratio.

For a non-MAR feature variable A_j, one can prove that:

- the probability of class c'_M is minimized by assuming the value of $\mathcal{A}_j$ to be different from a'_{Mj} whenever A_j is missing and the class of the instance is c'_M; this is the meaning of the lower counts $\underline{n}(a_{Mj},c'_M)$;[8]

[8] Note that, despite the different meaning, the counts $\underline{n}(a_j,c)$ for non-MAR feature variables are computed identically to the counts $n_l(a_j,c)$ for MAR ones.

- the probability of class c''_M is maximized by assuming the value of A_j to be a_{Mj} whenever A_j is missing and the class of the instance is c''_M; this is the meaning of the upper counts $\overline{n}(a_{Mj}, c''_M)$.

Once the realization of the non-MAR missing values is identified by upper and lower counts, the most unfavorable complete realization of the learning set has been chosen (i.e., the minimization over $d \in o$ has been accomplished), and we are in the case of Problem (6).

3.4 Credal Dominance Test with an Incomplete, Non-MAR, Unit to Classify

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

$$\min_{x_M \in o_M} \min_{d \in o} \inf_{p(\theta) \in \mathcal{P}(\theta)} \frac{p(c'_M | d^-, \hat{x}^+ \in \hat{o}^+)}{p(c''_M | d^-, \hat{x}^+ \in \hat{o}^+)}. \tag{8}$$

Problem (8) can be trivially solved as follows: (i) considering all the possible realizations x_M of the non-MAR part of the instance (this is accomplished by considering all the possible replacements for the missing values); (ii) for each $x_M \in o_M$, assuming x_M to be the realization of the non-MAR part of the unit to classify, and then solving Problem (7); (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 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. A more efficient polynomial-time procedure, which is still exact, can be designed to solve Problem (8) and is in fact given in Appendix A.

The underlying idea of such a procedure is to split the outer minimum in (8) in many minima, each one related to a different feature variable, and to distribute them at different places into the objective function. This makes is clear that the function to optimize is the lower envelope of a set of convex functions. In Appendix A we show that it is easy to compute the points where the function that determines the envelope changes, thus in fact obtaining a partition of the envelope function's domain with the property that in each of its elements the envelope function is convex. At that point, the function can be locally optimized efficiently on every element of the partition; and the global optimum is then just the minimum of the local optima so computed. The fact that the overall procedure is polynomial follows because the size of the partition is bounded by a polynomial.

A final remark is that, to make notation simpler, we refer to the possible realizations of the non-MAR variables in the instance to be classified as o_M; such a notation implies however that the non-MAR variables of the test set are the same of the training set. If instead the non-MAR variables of the test set are different from those of the training set, o_M should contain all the possible realizations of the variables which are non-MAR in the test set.

The NCC2 procedures are summarized in Fig. 2. Learning has linear complexity with respect to the number of attributes, while testing has roughly quadratic complexity, if the procedure carried out in Appendix A is adopted. Please refer to Appendix A also for the exact expression for the complexity.

3.5 Software Availability

The software JNCC2, implemented by the authors of this paper, implements the naive credal classifier 2. JNCC2 is open source, released under the GNU GPL license; it is hence freely available

```
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.

CLASSIFICATION OF AN INSTANCE

  1. set NonDominatedClasses := $\mathcal{C}$;
  2. for class $c' \in \mathcal{C}$
       • for class $c'' \in \mathcal{C}$, $c'' \neq c'$
           – if $c''$ is dominated by $c'$ (to be assessed via the below procedure), drop $c''$ from
             NonDominatedClasses;
           – exit;
       • exit
  3. return NonDominatedClasses.

DOMINANCE TEST BETWEEN TWO CLASSES ($c'$, $c''$)

  • list the attributes affected by a MAR MP or by an unknown MP in the instance;
  • for each $x_M \in o_M$ (i.e., for each possible realizations of the non-MAR part of the unit to
    classify):
      – assume $x_M$ to be the realization of the non-MAR part of the instance;
      – solve Problem (7) via Theorem 6;
      – if the computed solution is smaller than 1, $c''$ is not dominated by $c'$. STOP.
  • if, after having tried every $x_M \in o_M$, no solution greater than 1 has been found, $c'$ dominates
    $c''$.

/*alternatively to the above exhaustive procedure (for each $x_M \in o_M$) , the minimum can be computed more efficiently via the procedure presented in Appendix A.*/
```

Figure 2: Summary of NCC2 procedures.

together with user manual, sources and documentation. Being written in Java, it runs under any operating system; it has a command-line interface.

JNCC2 loads data stored in the ARFF format, which is a textual format (designed for classification problem), originally developed for WEKA (Witten and Frank, 2005), an open source software for data mining. Hence, the large public repositories of ARFF files can be used to extensively test NCC2.

For downloads and further information about JNCC2, see `www.idsia.ch/~giorgio/jncc2.html`.

4. Experiments

We consider 18 data sets from the UCI repository, and available in the ARFF format from the WEKA data sets page.[9] All the data sets are complete, that is, they do not contain any missing data.

We discretized the numerical features via MDL-based discretization (Fayyad and Irani, 1993), and then we split each data set into a training and a test set. Feature discretization is necessary, as NCC2 is designed to work with categorical variables. Discretizing the features on the entire data set introduces a slight optimistic bias in the evaluation of the classifiers accuracy (in principle, the discretization intervals should be computed on the training set, and then applied unchanged in the test set); yet, this is not a problem as our goal is to compare NBC and NCC2 in identical conditions, rather than to compare our findings with previous results obtained on the same data sets.

Also, since our goal is to fairly compare NBC and NCC2, and not to finely tune them for maximum performance, we did not perform feature selection (although, in fact, we remove numerical feature variables discretized into a single bin); yet, as both NBC and NCC2 are based on the naive hypothesis, redundant or mutually dependent feature variables might significantly bias the learning process; hence, in order to achieve maximum performance, one should consider selecting feature variables.

In the experiments presented in the following we generate artificial missingness on the original, complete data sets by using different MPs and then we compare NBC and NCC2 accuracy on the incomplete data sets.

4.1 Missingness Generation

We consider two different artificial MPs: (i) a MAR one, and (ii) a non-MAR, non-identically distributed one (nonMAR). The MPs we consider do not affect the class variable, as the class variable has been assumed to be always observed.

Note that we do not test mixed cases of MAR and non-MAR feature variables, which NCC2 is actually designed to treat. In fact, such settings would simply lead to results intermediate between those obtained under the MAR and the non-MAR settings. Yet, mixed settings are valuable and should be considered whenever possible when operating the classifier, as they allow for finely tuning the treatment of missing data to the characteristics of the MP.

The MAR MP turns into missing, with 5% probability, all the features of both the training and test sets. Such a missingness process actually meets not only the definition of MAR, but also the more restrictive definition called MCAR, that is, *missing completely at random*, see Little and Rubin (1987). We have taken into consideration also a non-MCAR, MAR MP; however, the results do not differ significantly from those obtained with the MAR MP (which satisfies also MCAR) described above.

The non-MAR MP works as follows : (i) it splits the categorical values of each feature variable into two halves; (ii) for each feature variable, it turns into missing, with probability 5%, the observations falling in the first half of values, on the training set; (ii) for each feature variable, it turns

[9] The URL is `http://www.cs.waikato.ac.nz/ml/weka/`.

into missing, with probability 5%, the observations falling in the second half of values, in the test set. Such MP is not identically distributed, as it follows a different pattern from training to test set.

We split each data set into equally sized training and test subsets. Using this training/test split, for each data set and for each MP we generate artificial missingness 100 times, producing hence 100 different training and test sets. The results we present are obtained as an average over 100 runs for each data set-MP pair.

4.2 Performance Measures

The performance of NBC is measured by its *accuracy*, that is, the percentage of correct classifications, while the evaluation of NCC2 requires a larger number of indicators: *determinacy*, that is, the percentage of classifications having as output a unique class; *single accuracy*, that is, accuracy of NCC2 when it is determinate; *indeterminate output size*, that is, the average number of classes returned when NCC2 is indeterminate; *set-accuracy*, that is, the percentage of indeterminate classifications that contain the true class. Note that if a data set has two classes, the output size is necessarily 2 and set-accuracy 100%; therefore, such two indicators are meaningful on data sets with more than two classes only.

We consider three classifiers: (i) NBC (with standard Laplace prior); (ii) NCC2-MAR, that is, NCC2 which assumes all missing values to be generated by a MAR MP; (iii) NCC2-nonMAR, that is, NCC2 which assumes all missing values to be generated by an unknown MP. Indeterminate classifications of NCC2-MAR are in fact due to prior uncertainty only, while indeterminate classifications of NCC2-nonMAR are due to both the prior and the missing values.

We then consider an additional set of indicators, which refer to the NBC accuracy on specific subsets of instances (note that such indicators, referring to subsets of instances, are written in italic):

- *NCC2-MAR D* and *NCC2-MAR I*, as the NBC accuracies on the instances classified respectively in a determinate and indeterminate way by NCC2-MAR; such indicators point out the effects of the prior specification on NBC;

- *NCC2-nonMAR D* and *NCC2-nonMAR I*, as the NBC accuracies on the instances classified respectively in a determinate and indeterminate way by NCC2-nonMAR; such indicators point out the joint effect of prior specification and missing values on NBC;

- Δ*NCC2-nonMAR*, as the NBC accuracies on the instances over which NCC2-nonMAR outputs a larger number of classes than NCC2-MAR; such an indicator points out the effect of missing values on NBC.

It turns out however that *NCC2-MAR D* coincides with the single-accuracy of NCC2-MAR, and that *NCC2-nonMAR D* coincides with the single-accuracy of NCC2-nonMAR: in fact, when determinate, NCC2-MAR and NCC2-nonMAR return the same output as NBC.[10] In the following we will hence only mention *NCC-MAR D* and *NCC-nonMAR D*; it is understood that the single-accuracy of respectively NCC2-MAR and NCC2-nonMAR coincides with such values.

[10] Differences might arise only in case that the NBC prior is not included in the credal set, and that such a prior leads to a classification which is different from the classification to which leads any single prior of the credal set. The frequency of such an event is however negligible and therefore the above indicators can be considered to be the same.

4.3 Overview of the Results

Table 1 reports the performance of NBC, NCC2-MAR and NCC2-nonMAR under the different MPs, averaged over all the data sets. The average of set-accuracy and indeterminate output size have been computed considering only data sets with more than two classes. The detailed results data set by data set are instead shown in Tables 6 and 7, which refer respectively to the MAR and non-MAR setting.

NBC achieves an average accuracy of about 80%; although its accuracy seems to be insensitive to the MP, we show later however, through a deeper analysis, that this is not the case.

The average determinacy of NCC2-MAR is about 91%, that is, NCC2-MAR yields set-valued classification in 9% of cases; this indicator is mainly influenced by the data set size and therefore it does not show great differences between the different MP settings. The point here is that when we declare all the feature variables to be MAR, imprecision in the probabilities is only originated by the imprecise prior density, as it follows from (1). Such a kind of imprecision strictly decreases with the size of the learning set because the prior, as in the precise Bayesian setting, counts less and less with more data.

The results detailed data set by data set (Tables 6 and 7) show that NCC2-MAR has considerably lower determinacy on the glass data set than on any other data set. This is a consequence of the low number of training instances (about 100) and high number of output classes (i.e., 7). In these conditions the prior has much weight and it originates more imprecision.

The determinacy of NCC2-nonMAR ranges from 33% under the MAR setting, to 50–55% under the non-MAR settings. The higher indeterminacy under the MAR setting for NCC2-nonMAR is due to the higher number of missing data: in fact, the MAR MP produces twice as many missing data as the non-MAR MPs, because it turns *all* the values of the feature variables into missing, with probability 5%, instead of half the values as the non-MAR MP does.

Looking at the results data set by data set, we can detect three factors that increase the indeterminacy of NCC2-nonMAR: (a) high prior uncertainty, that is, data sets over which NCC2-MAR is already quite indeterminate, and to which the uncertainty coming from missing data adds up; this is the case of glass; (b) high number of features variables (40–60), as in the case of kr-kp, optdigits, waveform, splice; in the case of the letter data set, the cause is instead (c) the high number of categories (10 on average) in which feature variables are discretized. Factors (b) and (c) increase the indeterminacy as they increase the number of complete data sets $d \in o$ consistent with the incomplete one. Increasing the number of the complete data sets makes it more likely to obtain lower values of the optimum in (1), eventually yielding a larger number of non-dominated classes. It is interesting to observe that the factors (a)–(c) above are the same that can lead NBC to overfitting. Hence, we can conjecture that classifiers based on precise and imprecise probability, respectively react in different ways to such critical characteristics of the data, the first ones by losing reliability, the second ones by becoming excessively cautious.

Let us focus now on the the size of the indeterminate output, which is the average number of classes in the set-valued classifications returned by NCC2 in a certain setting, and on set-accuracy, which denotes the percentage of times when set-valued classifications contain the actual class (we recall that the indeterminate output size and set-accuracy are measured only on data sets with more than two classes). The size of the indeterminate output should be compared with the (average) number of classes of the data sets under considerations, which is 8.8; hence, NCC2-MAR returns on average less than half the classes, and NCC2-nonMAR slightly more than half the classes. This

	MAR	non-MAR
	NBC	
Accuracy (%)	80.5	80.9
	NCC2-MAR	
Determinacy (%)	91.1	91.1
Set-Acc. (%)	84.0	84.3
Indeterminate Output Size	2.6	2.6
	NCC2-nonMAR	
Determinacy (%)	33.0	49.4
Set-Acc. (%)	97.9	96.6
Indeterminate Output Size	6.0	5.3
NBC accuracy (%) on subsets of instances		
	MAR	non-MAR
NCC2-MAR D	83.2	83.5
NCC2-MAR I	48.4	48.0
NCC2-nonMAR D	92.2	90.4
NCC2-nonMAR I	74.2	71.3
ΔNCC2-nonMAR	75.1	72.7

Table 1: Measured performance, averaged over all the 18 data sets. The average of set-accuracy and indeterminate output size has been computed considering only the data sets with more than two classes; the average number of classes of such data sets is 8.8.

shows that set-valued classifications are informative: they lead us to drop on average half of the classes, as sub-optimal, for the more doubtful instances, thus preventing an over-confident use of the issued judgment. Moreover, indeterminate classifications allow very high set-accuracy under any MP setting; about 85% for NCC2-MAR and about 96% for NCC2-nonMAR. Set-valued classifications appear then as a very effective way to maintain reliability while conveying informative content. This is especially appealing for contexts in which the classification outcome is very sensitive, such as, for instance, the medical area.

Probably the most interesting part of our results concerns the analysis of the accuracy of NBC made separately for the instances classified in a determinate way by NCC2 and for those whose classification is set-valued. Here the intuition is that we expect NBC to perform worse in the latter set of instances, as NCC2 deems that they are harder to classify. Indeed, by averaging over the two different MPs, we obtain that the accuracy of NBC drops of 33 points from *NCC2-MAR D* to *NCC2-MAR I*; of 20 points from *NCC2-nonMAR D* to *NCC2-nonMAR I* and of 19 points from *NCC2-nonMAR D* to Δ*NCC2-nonMAR*.

The overall performance of NBC can hence be regarded as the average of a good accuracy on the instances which are easier to classify, and a much lower accuracy on the instances which are harder, due to the fact that the available information about them is reduced with respect to the former ones. Such a reduction depends in part on the learning set, which in general contains different degrees of information in relationship with different units to classify; and in part on the number and type of missing feature variables in those units, which contributes to determine the information that the classifier can exploit to classify them.

We regard the alternate good-bad performance of NBC on different units as an important finding, which should be brought to light and properly taken into account. In contrast, such a finding can well remain hidden if we only measure the predictive accuracy, as it is common with traditional classifiers, because it is an average over all the instances in the test set.[11] NCC2, instead, recognizes the more difficult instances and preserves its reliability by issuing indeterminate classifications. (NCC2-MAR does the same, but not with respect to the hardness originated by missing data, both in learning and test set.)

In fact, the following relationships hold on the average indicators, but also on *every* data set-MP pair:

- *NCC2-MAR D* $>$ *NCC2-MAR I*,
- *NCC-nonMAR D* $>$ *NCC2-nonMAR I*,
- Δ*NCC2-nonMAR* $>$ *NCC2-nonMAR I*.

The intuition behind these inequalities is, as before, that on the instances where NCC2 is indeterminate, there is a drop of accuracy of NBC compared to the remaining instances. This happens not only on every data set but also under every experimental setup that we considered. This enforces the idea that NCC2 truly isolates instances that are hard to classify and where, as a consequence, NBC is less reliable.

Despite the inequalities remain the same in the different settings, one should not overlook the fact that the amount of instances isolated by NCC2 may be also very different in the different cases. In particular, looking at Table 1, we see that NCC2-nonMAR may produce even 5–6 times as much indeterminacy as NCC2-MAR. This is not surprising as prior uncertainty tends to vanish with larger samples whereas this does not need to be the case for the uncertainty originated by missing values. We see then that the large majority of indeterminate classifications issued by NCC2-nonMAR are due to missing data rather than prior uncertainty.[12]

It is useful to remark also that the indeterminacy generated by NCC2-nonMAR may sometimes be greater than necessary, or, in other words, that NCC2-nonMAR may show an excess of caution. This makes the drop in NBC accuracy be usually smaller from *NCC2-MAR D* to *NCC2-MAR I*, than from *NCC2-nonMAR D* to Δ*NCC2-nonMAR*. The excess of caution appears because the MP acting on the data is simpler than what NCC2-nonMAR expects. One reason is that for the MP described in Section 4.1 only half the values contained in $\mathcal{A}_j$ are actually possible replacements. This information is not accessible to NCC2-nonMAR, which then considers all the values in $\mathcal{A}_j$ as possible replacement for the missing values of A_j. The other, more important, reason is that we have

[11] There are some reasons why computing in addition the standard deviation of the predictive accuracy would not be very helpful either: (i) the standard deviation might be small simply because so is the subset of test-set instances on which the classifier performs badly; (ii) the standard deviation cannot be reliably estimated in the case of small samples; (iii) finally, even if computing the standard deviation might point to the problem, it would not help to isolate the critical instances, nor to know what to do with them, as opposed to what NCC2 makes it possible to do.

[12] With respect to the variability of the empirical measures collected, we can observe that *NCC2-MAR I* has usually larger standard deviation than the others (see Tables 6 and 7); this is expected, as NCC2-MAR tends to generate indeterminacy on relatively few instances. Anyway, the difference between *NCC2-MAR D* and *NCC2-MAR I*, and between *NCC2-NonMAR D* and *NCC2-NonMAR I*, is much larger than the standard deviations of the estimate of then indicators (actually, is it well above twice the standard deviation in most cases).

deliberately designed the non-MAR MPs so as to make them act in quite a naive way. We have done so to show that even those MPs can lead to appreciable problems for NBC, as confirmed by the fact that for the large majority of data sets it holds that the NBC accuracy on the ΔNCC2-nonMAR area decreases, moving from the MAR to the non-MAR setting. And, finally, when we consider MPs that are not naive, as the one described in Sect. 4.6, we see that the caution of NCC2-nonMAR is fully justified, NBC being totally unreliable on the area of indeterminacy.

4.4 NBC Probabilities Vs. Set-valued Classifications

We have shown that, thanks to imprecise probabilities, NCC2 delivers set-valued classifications on hard-to-classify instances, over which the accuracy of NBC clearly drops. In the following, we further compare the predictions of NBC and NCC2 by taking into consideration also the posterior probabilities computed by NBC for the classes. To make things clearer we carry out the analysis in two steps.

The first step, described in the next two sections, focuses on a simplified setup that allows us to make a very detailed analysis: a single data set containing two classes, and subject only to a MAP MP. The focus in this case is on prior ignorance, as the only possible source of indeterminacy for NCC2; we can therefore understand how strong assumptions about the prior can affect the ability of NBC to compute posterior probabilities. Thanks to the restriction to the binary case (i.e., two classes), we can then also clearly see that there is a difference between the cases that are deemed doubtful by NBC (i.e., with probability close to 50%) and those that are deemed so by NCC2 (i.e., that lead it to indeterminacy).

The second step extends the analysis to all the data sets and to ignorance originated by non-MAR missing data. In this case there is necessarily less detail but we have the advantage of having the full comparison between NBC probabilities and NCC2 indeterminacy, and where we observe that the situation basically resembles the one observed in the first step. This is done in Section 4.4.3.

4.4.1 An Illustrative Example: The Spect Data Set

We focus on the spect data set, which is made of 2 classes and 267 instances.

For the following analysis, we split the data set into training set of 67 instances and a test set of 200 instances. We choose a 25-75% split in this example because we want to clearly see the effect of the prior, which is well known to decrease with increasing sizes of the learning set.

We run the experimental framework of Section 4.3 with the MAR MP; that is, we generate 100 pairs of training and test sets (made artificially incomplete by the MP) from the original training/test split. The findings presented in the following are obtained by analyzing the predictions issued over the 100 test sets.

In particular, we consider four pieces of information for each classified instance: 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. The instances are then partitioned into subsets, according to the probability estimated by NBC for the returned class, that is, instances for which NBC estimates a probability in the range 50–55%, 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 measure: (a) the determinacy of NCC2-MAR; (b) the accuracy achieved by NBC on the instances classified determinately and (c) indeterminately by NCC2-MAR.

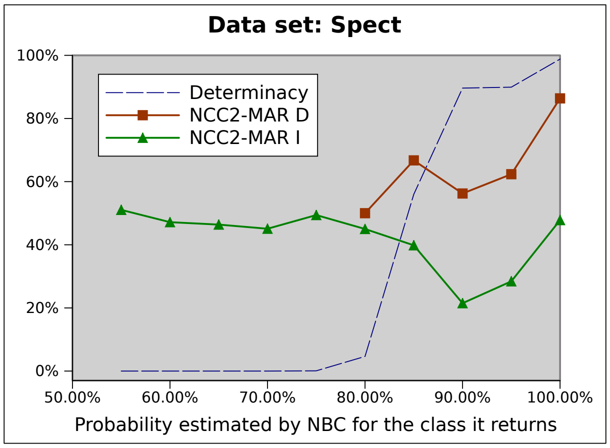


Figure 3: Relationship between the posterior probabilities computed by NBC and the output of NCC2-MAR on the spect data set.

The results are reported in Figure 3. The dashed line represents the percentage of instances in the subset (related to a certain range of probabilities) under consideration that is classified determinately by NCC2. We see then that there is a positive association between higher posterior probabilities computed by NBC and higher determinacy. This is a natural effect but it needs some words to be explained, and we postpone this task to the next section.

An interesting point is that 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 75%. 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 (uniformly) at random, at is follows from the line for NCC2-MAR I. The point here is that NBC believes to be quite confident on a number of instances (assigning probabilities that go as up as 75%) but in practice it is not. Stated differently, it believes it knows but it does not. NCC2 simply knows not to know from the very beginning and this enables it to avoid losing credibility.

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 *NCC2-MAR D* and *NCC2-MAR 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.

The last discussion points to an important question, which is useful to stress: that is, to the fact that NCC2 does not suspend the judgment only on instances that are deemed doubtful by NBC, that is, 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

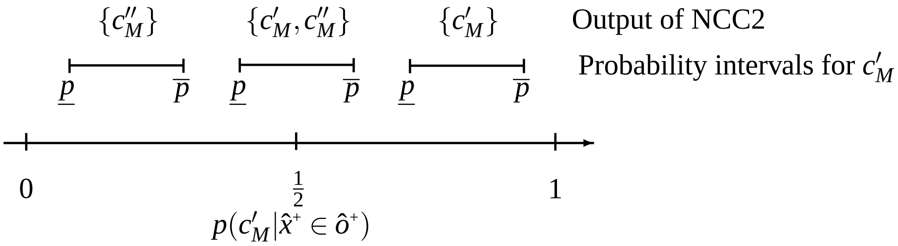


Figure 4: A graphical view of the test of dominance in the binary case. The output of the classifier is indeterminate if and only if the posterior probability interval for c'_M contains $\frac{1}{2}$.

determinately by NCC2 (while the behavior on the remaining ranges of probabilities followed a pattern similar to the one shown in Fig. 3).

4.4.2 ASSOCIATION BETWEEN NBC PROBABILITIES OF NCC2 DETERMINACY

In the following, we explain, with reference to a generic data set with two classes, why there is a positive association between increasing posterior probabilities computed by NBC and increasing determinacy of NCC2.

Re-consider the notation that we have introduced for the test of dominance in (1). Define the posterior *lower* and *upper probabilities* of class c'_M respectively as

$$\underline{p}(c'_M|\hat{x}^+ \in \hat{o}^+) := \inf_{p(\theta)\in\mathcal{P}(\theta)} p(c'_M|\hat{x}^+ \in \hat{o}^+)$$

and

$$\overline{p}(c'_M|\hat{x}^+ \in \hat{o}^+) := \sup_{p(\theta)\in\mathcal{P}(\theta)} p(c'_M|\hat{x}^+ \in \hat{o}^+).$$

Compared to (1), here we have removed the parts related to the non-MAR missingness process as in this moment we prefer to keep things simple by focusing on the MAR case, as in the previous section.

With the new definitions, the test of dominance (1) can be easily re-written in the case of a data set with two classes as a simple test that checks the position of the interval $[\underline{p}(c'_M|\hat{x}^+ \in \hat{o}^+), \overline{p}(c'_M|\hat{x}^+ \in \hat{o}^+)]$ relative to the probability value $\frac{1}{2}$. This is shown graphically in Fig. 4. Such a figure shows three possible positions of the posterior intervals, which are represented as segments. When the interval contains the value $\frac{1}{2}$, as in the middle case, then the output of the classifier is indeterminate: both classes are returned. Here we have complete indeterminacy because it is not possible to know if one of the two classes has posterior probability larger than the other (note that in this case the interval for c''_M contains $\frac{1}{2}$ as well). In the remaining two cases, the value $\frac{1}{2}$ is either greater or less than all the points in the interval and this allows the classifier to know that one of the two classes is dominated, thus returning a determinate classification, as shown above the intervals. Now, assume temporarily the width of the NCC2 interval to be constant across all the instances. By construction, the posterior interval computed by NCC2 is typically "close" to the posterior probability given by NBC.[13] Hence, the NBC probability and the NCC2 interval tend to move together; if the NBC

[13] In our experiments the posterior probability is actually very often in the interval, but we cannot say this is always the case because the Laplace prior used for NBC is not contained in the prior credal set of NCC2.

probability grows, then the NCC2 interval reaches a point that makes it exclude the value $\frac{1}{2}$, leading to a determinate classification. This explains the basic mechanism behind the association between NBC probabilities and NCC2 determinacy.

Let us now move on to the realistic scenario that involves dropping the simplifying assumption about the width of the NCC2 interval to be constant.[14] In fact, it practice it happens that the length of the NCC2 interval varies from unit to unit, as it depends on the amount of information that the learning set contains about the *specific unit under consideration*: the less the information the wider the interval, and vice versa. The fact that the learning set contains different amounts of information for different units to classify should not be surprising: a unit to classify can be put in correspondence with the subset of the learning set characterized by the values taken by the feature variables for the unit under consideration; and these subsets have (also very) different sizes (remember that we are dealing only with the MAR case; in the non-MAR case, we should consider also the effect of missing data and the interpretation would be slightly more complicated, but still of a similar kind).

Therefore, it may well happen that for a certain unit to classify to which NBC assigns a high posterior probability, the width of the NCC2 interval dilates so much that $\frac{1}{2}$ gets included in the interval, leading to an indeterminate classification. In this case, NCC2 deems that there is little information in the learning set about the unit under consideration, and expresses this by a large width of the interval. NBC does not have such expressive means and, moreover, may generate such a high probability mostly because of its prior density, drawing then conclusions that are not supported by the data.

In any case, it is clear that increasing the NBC probability makes it less likely that the NCC2 interval is as large as to be able to contain the value $\frac{1}{2}$. This is the reason why we observe the, very natural at this point, association between increasing NBC probabilities and increasing NCC2 determinacy.

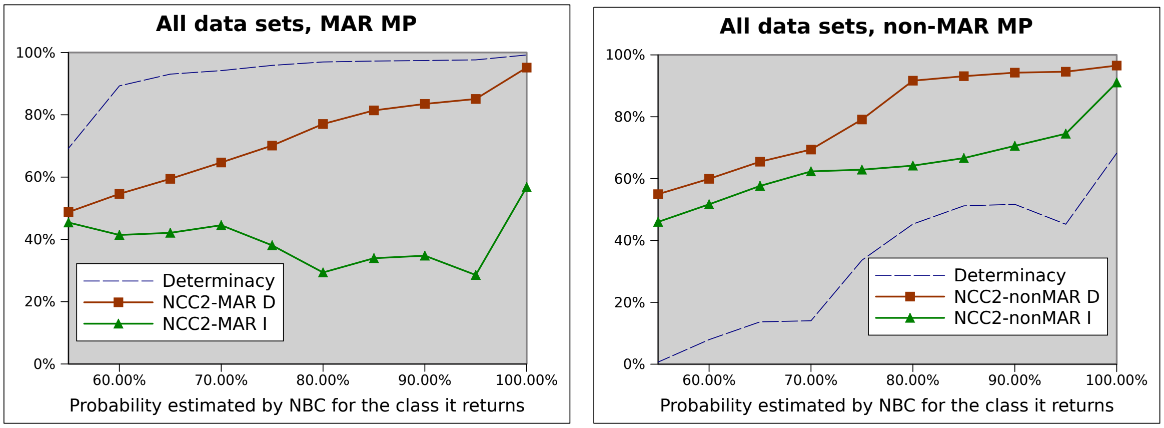
4.4.3 Results on All the Data Sets

In this section we consider all the usual 18 data sets with the same training/test splits of Section 4, that is, 50%-50%. We consider two settings: the first compares NBC and NCC2-MAR when the MP is MAR, and the second compares NBC and NCC2-nonMAR when the MP is non-MAR. We analyze jointly the predictions issued on all the data sets, respectively under the first (Figure 5a) and second setting (Figure 5b).

One point to consider is that in data sets with two classes, one can spot the instances doubtful for NBC looking at those with probability around 50% for the returned class. However, for data sets with more than two classes, the instances that are doubtful for NBC are not as easy to recognize as before; for instance, with four classes both the posterior mass functions [40%, 40.5%, 15%, 4.5%] and [24.5%, 25.5%, 25%, 25%], lead to doubtful classifications for NBC. For this reason, we only focus now on the case when NBC is confident, that is, on the instances with probability for the returned class greater than or equal to 55%.

Looking at the figures we see that the determinacy is much higher in the MAR setup than in the non-MAR; in fact, we have already seen that the non-MAR case leads in general to a much larger amount of instances that are isolated as difficult ones. Nevertheless, both Figures 5a and 5b show a clear drop of accuracy of NBC, for the same level of posterior probability of the predicted class, from

[14] If it were really so, the behavior of NCC2 would be quite trivial, and could be reproduced by that of an NBC that uses a threshold larger that $\frac{1}{2}$ to decide when a class should be said to dominate the other.



(a) NBC, NCC2-MAR and MAR MP

(b) NBC, NCC2-nonMAR and non-MAR MP

Figure 5: Relationship between the posterior probabilities computed by NBC and the output of NCC2.

the instances classified in a determinate way by NCC2 to the others. The drop is especially striking on the instances classified confidently by NBC, when the computed probability is for instance larger than 70%. This is more evident in Fig. 5a; on the other hand, the drop observed in Fig. 5b applies to a much larger set of instances. In fact, NCC2-nonMAR suspends the judgment frequently also on the instances classified by NBC with probability greater than 80%; 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.

The instances for which NBC returns a probability higher than 90% are of particular interest, also because they constitute 62% of the total instances. On this area, NCC2-MAR returns 1.5% of indeterminate classifications on which it achieves the following remarkable performance: *NCC2-MAR D*: 94%; *NCC2-MAR I*: 54%. The performance of NCC2-nonMAR in this area is as follows: determinacy 66%, *NCC2-NonMAR D*: 96%; *NCC2-NonMAR I*: 89%. The drop is in this case significant, yet smaller than under MAR: in fact, as already pointed out NCC2-nonMAR is more conservative than necessary under this setting.

Summing up, hence, uncertainty related to the choice of the prior manifest itself more evidently, but not only, on the instances in which NBC is less confident; yet, only part of such doubtful instances lead to indeterminate classifications. On the other hand, missing data treated as non-MAR can lead to many indeterminate classifications even if the probability computed by NBC for the returned class is high. This is clear by comparing the dashed lines in Figures 5a and 5b.

4.5 Results with Increasing Missingness

Next, we analyze how NCC2-nonMAR deals with very high levels of missingness. We adopt the non-MAR MP, with missingness rate at 15% per feature variable; given how the MP works, it will produce about 7.5% missing data on each feature. We focus on two data sets: segment-challenge

		NBC	NCC2-nonMAR				Subsets	
	MP rate (%)	Accuracy (%)	Determinacy (%)	Accuracy (%)	Set-Accuracy (%)	Output Size (%)	NCC2-nonMAR D	NCC2-nonMAR I
segment	5	91.6	54.8	97.7	99.4	5.5/7	97.7	84.1
segment	15	91.4	18.3	99.4	99.9	6/7	99.4	89.6
segment-pruned	5	93.8	77.8	98.0	99.7	4.6/7	98.0	79.1
segment-pruned	15	93.0	48.8	99.1	99.5	5.1/7	99.1	87.7

Table 2: *Segment-challenge* data set: impact of feature selection on the performance of NBC and NCC2-nonMAR. Standard deviation of the indicators is always smaller than 2%, that is, all indicators ± (at maximum) 2 percentage points.

(classes: 7; feature variables: 19; average number of bins per discretized feature variable: 5.6)[15] and waveform (classes: 3; feature variables: 40; average number of bins per feature variable: 8.4).[16]

Let us stress that in our intentions the results we present constitute a kind of worst-case analysis in terms of achieved determinacy, since (a) the amount of missing data generated by the MP is far higher than those usually found in real data sets; (b) NCC2-nonMAR treats all feature variables as non-MAR, both in training and testing (while usually, the investigator is able to isolate some MAR feature variables); (c) the considered data sets contain a high number of feature variables, classes and bins per feature variable, and these characteristics work together towards rising the indeterminacy of NCC2-nonMAR. Moreover, as already discussed, given how the non-MAR MP works, NCC2-nonMAR is more conservative than necessary under this setting.

Tables 2 and 3 report the results, displaying also the outcomes previously obtained with missingness rate 5%. In fact, while the performance of NBC is substantially insensitive to the amount of missing data, the determinacy of NCC2-nonMAR clearly decreases; with rate of missingness per feature variable equal to 15%, the determinacy of NCC2-nonMAR drops to about 1% on waveform and 18% on segment. As a side-effect, we see also that the indicator NCC2 I increases with the level of missingness; this is due to the fact that the number of instances on which NCC2 I is computed approaches the overall test set, so that NCC2 I approximates the average accuracy of NBC. Remarkably, however, even in such a difficult setting, there is a clear drop of accuracy (some 15 points), on both data sets, between NCC2 D and NCC2 I.

Is there a way to reduce the indeterminacy of NCC2-nonMAR? A first way to improve the situation is feature selection. On the one hand, feature selection is necessary if one wants to get maximum performance from NBC and NCC2, since redundant feature variables, which violate the naive assumption, might skew the learning process of both classifiers; on the other hand, reducing the set of feature variables reduces the amount of missing data, which is beneficial for the deter-

[15] Two feature variables, out of the 19, are discretized into a single bin; they are not considered when computing the average number of bins.

[16] Twenty feature variables, out of the 40, are discretized into a single bin; they are not considered when computing the average number of bins.

		NBC	NCC2-nonMAR				Subsets	
	MP rate (%)	Accuracy (%)	Determinacy (%)	Accuracy (%)	Set-Accuracy (%)	Output Size (%)	NCC2-nonMAR D	NCC2-nonMAR I
waveform	5	81.3	54.3	89.5	99.9	2.1/3	89.5	71.7
waveform	15	81.1	1.2	100	100	2.7/3	100	81.1
waveform-pruned	5	82.3	65.9	88.1	99.9	2.1/3	88.2	70.9
waveform-pruned	15	81.7	10.2	97.0	100	2.5/3	96.9	79.9

Table 3: *Waveform* data set: impact of feature selection on the performance of NBC and NCC2-nonMAR. Standard deviation of the indicators is always smaller than 2%, that is, all indicators ± (at maximum) 2 percentage points.

minacy of NCC2-nonMAR. We performed feature selection by cross-checking the suggestions of different methods implemented in WEKA (Witten and Frank, 2005); eventually, we selected 6 feature variables for segment-challenge and 14 for waveform.

In Table 2 and 3, the data sets after feature selection are referred to as *pruned*. The effectiveness of the feature selection we performed is demonstrated by a slight yet sensible improvement of NBC accuracy; as for NCC2-nonMAR, it achieves a satisfactory 50% of determinate classifications on segment-challenge, and a less satisfactory 10% on waveform. Still, it is remarkable that such a 10% of instances is naturally classified in a determinate way under such weak assumptions and with such an amount of missing data; this should make us quite confident about the reliability of the classification, and indeed the measured accuracy is 97%. The drop between NCC2 D and NCC2 I after feature selection does not show major changes compared to before feature selection.

So far the analysis in the most conservative possible conditions. Let us recall at this point that a strength of NCC2 is the possibility of declaring that some feature variables are MAR (possibly changing this from learning to test set). This is likely to be often possible to do in practice if we only consider that currently most classifiers take for granted that all the feature variables should be MAR. We considered then a new setup for waveform-pruned obtained by declaring that only half the feature variables are non-MAR both in training and testing; and the determinacy of NCC2 raised up to 58.5%.

4.6 A More Sophisticated MP

So far, we have considered a very simple MP, which in particular works independently on each feature variable. It is interesting at this point to consider a more complex MP, working for instance on the joint values of the feature variables. We show that this can have much bigger effects: in fact, it can lead not only to severe misclassifications, but also to erroneous empirical evaluations of the accuracy of classifiers which assume MAR.

We consider now the vote data set, also taken from the UCI repository. The data set contains information about the United States Congressional Voting; the classification task is to guess whether a certain congressman is republican or democrat, by looking at his/her votes about some critical laws. The data set has 435 instances, 16 features variables and 2 classes; by feature selection, we reduce to 3 the number of features variables. The pruned data set is complete, that is, it does not contain any missing data. In the following, we denote by the symbol “*” a missing value generated by the non-MAR MP. Let us name as “type A” the instances with values (*n, y, n, class1*) and as “type B” the instances with values (*y, n, n, class0*). The data set contains about 26% type A and 26% type B instances.

The malicious MP that we consider turns the type A instances of the training set into (*, *, *n*, *class1*), and the type B instances of the test set into (*, *, *n*, *class0*); it hence affects about 26% of the training set and 26% of the test set.

We measured the following on the test set:

- NBC accuracy 71.4%;
- NCC2-MAR: determinacy 100%; single accuracy 71.4%;
- NCC2-nonMAR: determinacy 55.3%, single accuracy 99.2%.

Remarkably, in this case NCC2-MAR is always determinate, and therefore it behaves identically to NBC. With reference to the previous results on the same data set, obtained with a simpler MP, there is a major drop of accuracy (some 25 points) for NBC (and consequently for NCC2-MAR, too); on the other hand, NCC2-MAR becomes more indeterminate, but it preserves a high single-accuracy.

Going more deeply in the analysis, the accuracy of NBC and NCC2-MAR is 99.2% on the instances NCC2-nonMAR D, and 37.1 only (lower than random guessing!) on the instances NCC2-nonMAR I. Hence, this MP heavily deteriorates the reliability of the classifiers which assume MAR. NCC2-nonMAR, being based on CIR, behaves instead robustly even against such a malicious MP.

However, there is a further point of interest, concerning the failure of the empirical evaluations of classifiers that are always determinate, when data are made missing by a non-MAR MP. Let us assume that the only available data are the instances of the training set. Evaluating by cross-validation the classifiers on the instances of the training set, we measure:

- NBC accuracy 88.2%;
- NCC2-MAR: determinacy 89.9%; single accuracy 93.9%;
- NCC2-nonMAR: determinacy 49.5%, single accuracy 100%.

The evaluation of the accuracy of NBC and NCC2-MAR is hence heavily biased when compared against their actual performance on the test set. This phenomenon is discussed at some length in Section 6 of Zaffalon (2005b). On the other hand, NCC2-nonMAR is reliably evaluated.

Of course, this kind of “extreme” example heavily relies on the fact that the MP is not identically distributed. Yet, note that if one is ignorant about the MP, such a behavior should be considered as a possibility, which is just what NCC2 does. On the other hand, imposing the assumption that the MP is identically distributed should be done on a case-by-case basis, as doing it in general is questionable (unlike imposing the data generation process to be so, given that MPs are usually processes of a very different kind). Alternative assumptions, between the two extremes of assuming ignorance and assuming that the MP is identically distributed, are not so easy to envisage if we also require that they are widely applicable as one would like them to be in the field of data mining.

Algorithm	Determinacy (%)	Accuracy (%)	Output Size	Set-accuracy (%)
NBC	-	62.4 ± 5.1	-	-
NCC2-MAR	91.8 ± 3.3	64.2 ± 5.5	2.2 ± 0.2	83.9 ± 15.4
NCC2-nonMAR	65.1 ± 5.5	60.1 ± 7.7	3.3 ± 0.3	89.8 ± 5.7

Table 4: Results on the eucalyptus data set. Accuracy refers to determinate classifications only, while output size to indeterminate classifications only.

	Values of feature variable "vigour"						
Output class	*1*	*2*	*3*	*4*	*5*	*6*	*7*
c_1 ("*none*")	24	10	14	1	6	1	0
c_2 ("*low*")	0	3	24	17	8	2	0
c_3 ("*average*")	0	0	2	17	31	14	1
c_4 ("*good*")	0	0	0	6	28	57	16
c_5 ("*best*")	0	0	0	0	8	21	23

Table 5: Counts $n(a_j, c_k)$ for feature variable "vigour" in the eucalyptus data set.

In summary, we conclude that the conservative approach provided by NCC2 can be considered as a valuable avenue to reliably cope with situations where we miss substantial information about (part of) the MP. Such an approach might become even more effective by exploiting coarsened observations, as a way to insert knowledge about the MP. This is discussed in the next section.

4.7 The Eucalyptus Data Set: A Difficult One For NCC2-nonMAR

The *eucalyptus* data set is available from the Weka (Witten and Frank, 2005) repository[17] of public data sets. It contains 736 instances; there are 19 feature variables and 5 classes. After feature selection, there are 8 feature variables left; about 3% of the values in the data set are missing. NBC, NCC2-MAR and NCC2-nonMAR have been validated via 10 runs of 10 folds cross-validation; feature variables have been discretized via MDL-based discretization (Fayyad and Irani, 1993). Results are reported in Table 4.

Strikingly, NCC2-MAR has higher single-accuracy than NCC2-nonMAR. To investigate this issue, we split in a stratified way (i.e., keeping the proportion of the classes as close as possible between training and test set) the data set into a training and a test set; this enables us to detect a set of 35 critical instances, which are classified accurately and in a determinate way by NCC2-MAR, and in a totally indeterminate way by NCC-nonMAR, which returns five classes out of five.

[17] The URL is `http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html`.

A key point is that all these instances lack the value of feature variable vigour. The frequencies $n(a_j, c_k)$, reported in Table 5, show that, for any class c_k, there is at least a value $\tilde{a_j}$ such that $n(\tilde{a}_j, c_k)$=0. This is due to different reasons, such as the high number of classes in which the feature variable is discretized; the skew of the distribution of the counts $n(a_j, c_k)$, for any class c_k, towards certain values of the feature variable; the high number of output classes (i.e., 5), and also the relatively small size of the data set.

In fact, if the value of vigour is missing and the MP is supposed to be non-MAR, for any pair of classes (c_i, c_j) there exists a replacement $\tilde{a_j}$ for the missing value such that $n(\tilde{a_j}, c_i) = 0$, and hence $p(c'_M|d^{-}, \hat{x}^{+} \in \hat{o}^{+})/p(c''_M|d^{-}, \hat{x}^{+} \in \hat{o}^{+}) = 0$, thus preventing the existence of any credal-dominance relationship between any two classes. In this condition, hence, NCC2-nonMAR has to output all the classes.

On the other hand, the conditional counts of the remaining feature variables, strongly skewed towards class c_1, allow NCC2-MAR (and NBC as well) for predicting all these instances accurately and in a determinate way. This phenomenon is the reason why NCC2-MAR is both more accurate and determinate than NCC2-nonMAR on the eucalyptus data set.

4.7.1 SOLUTION VIA COARSENING (OR SET-BASED OBSERVATIONS)

Yet, NCC2-nonMAR is processing correctly the eucalyptus instances: if the MP affecting the vigour feature variable is non-MAR, by definition all the possible replacements for the missing values should be taken into consideration. However, NCC2-nonMAR is unnecessarily cautious, as demonstrated by both NCC2-MAR and NBC, which are able to classify the critical instances both accurately and in a determinate way. It appears hence that some further knowledge should be put into the classifier, to mitigate its unnecessary caution; yet, we try to avoid assuming MAR. A step forward in this direction can be accomplished by introducing *set-based* (or *coarsened*) observations.

Up to now, we have assumed that manifest values are either equal to the latent ones, or set equal to the symbol of missing data. In the former case, the value is known exactly; in the latter, it is known in the most indeterminate way, that is, we only know that it belongs to the set of possible values of the feature variable. Coarsened observations represent an intermediate situation of knowledge, in which the observation, for instance referring to feature variable A_j, is known to belong to a set $\mathcal{A}_j^{'} \subseteq \mathcal{A}_j$. The size and composition of $\mathcal{A}_j^{'}$ can vary between different set-based observations of the same feature variable, thus allowing for a high degree of flexibility. Actually, the possible values of $\mathcal{A}_j^{'}$ are all the possible non-empty subsets of $\mathcal{A}_j$.

If A_j is assumed to be affected by a non-MAR MP, set-based observations restrict the set of possible replacements from $\mathcal{A}_j$ to $\mathcal{A}_j^{'}$ (we recall that $\mathcal{A}_j^{'}$ will generally be different for each set-based observation). Therefore, set-based observations provide, without implying any strong assumption about the MP, a higher degree of knowledge than missing data, and they tend to generate less indeterminacy than missing data.

Remarkably, real MPs often produce set-based observations rather than missing data. For instance, let us consider a doctor that recommends a patient to go through exam B only if exam A is positive. In fact, the domain knowledge of the doctor makes it possible to conclude that if A is negative, the outcome of B will range within a restricted set of values, not influential for the diagnosis. On the contrary, if A is positive, the value of B cannot be predicted. However, when B is not performed because A is negative, we do have a set-based observation rather than a missing value.

In many cases, it is possible to provide coarsening sets rather than missing values at the time data are collected, by excluding unsuitable values from the set of the possible replacements.

Let us stress that using set-based observations and assuming MAR are two fundamentally different ways of inserting knowledge into the classifier. In fact, set-based observations incorporate specific domain knowledge (obtained at data collection or data validation time) into the data set and eventually into the classifier. On the other hand, MAR is an assumption typically done by the modeler, whose tenability is only rarely checked with domain experts.

What happens using set-based observations on the eucalyptus data set? On the 35 critical instances we tried to substitute the missing values of vigour with set-based observations, using a coarsening set $\mathcal{A}_j^{'} = \{1,2,3,4,5\}$, which contains 5 values out of the 7 possible. Using such set-based observations, NCC-nonMAR classifies 32/35 critical instances correctly and in a determinate way; on the remaining 3 instances, it outputs 4 classes, including the true one. In this way, the usual inequalities (*NCC-MAR D* $>$ *NCC-MAR I*, *NCC-nonMAR D* $>$ *NCC-nonMAR I*, *ΔNCC2-nonMAR* $>$ *NCC-nonMAR I*) hold also on the eucalyptus data set.

In our example, set-based observations constitute an effective way to insert knowledge about the MP into the classifier without assuming MAR; they appear hence as a promising approach for credal classifiers. It is useful to observe that coarsened observations have received considerable attention in the traditional literature of statistical inference from incomplete observations. A work of paramount importance on this topic is the popular coarse-data model of Heitjan and Rubin (1991), which generalizes the missing-data model of Rubin (1976). This appears to enforce the potential of coarsening for classification in general, and to make it worth of further investigation. From a practical viewpoint, the use of set-based observations would require some changes in the data archiving formats, which are usually designed to store precise observations only.

5. Conclusions

There are usually different amounts of information in the data that a classifier can exploit to classify different units. This depends in part of the fact that a learning set can be more informative about some units than some others, and in part on the type and amount of missing values in the units to classify. As a consequence, a classifier's predictions may be more uncertain on some units than on some others. But if we impose that the classifier is inferred using a single prior density and that it considers all the missing data as ignorable, we lose much of the capability to distinguish that some units are harder to classify than others: we cannot see anymore that some of the strong conclusions that we obtain are determined by our strong assumptions rather than the supposedly strong information in the data, and are hence actually fragile. Nor can help much in this respect the experimental evaluations of the classifier as they are, somewhat necessarily, designed to yield its average performance. And yet, we do care about distinguishing the hardness of different units in many real-world applications: in the case of medical diagnosis, for instance, we do not want a classifier to be good only *on average*; we want it to make reliable predictions on every single patient.

In this paper we have tried to pursue such a goal just by weakening the assumptions that are traditionally made by classifiers. We have modified the naive Bayes classifier so as to model prior ignorance in an objective-minded way by using a set of prior densities, and by giving the opportunity to treat some of the missing values as originated by a missingness process that we do not know (and the others by a MAR one). The resulting model, called naive credal classifier 2, departs from the more traditional classifiers in a number of ways, the more substantial one being the fact that NCC2 makes set-valued classifications in general: it issues a determinate classification (i.e., a singleton) only when it deems that there it has enough information to do so.

Extensive empirical evaluations have shown that NCC2 has high accuracy when it issues determinate classifications, and that when it is more cautious, it is very often justified: NBC is clearly shown to considerably decrease its classification accuracy on the instances classified in a set-valued way by NCC2, as well as its ability to compute predictive posterior probabilities. This indicates that set-valued classifications do indeed isolate the instances of the test set that are hard to classify. And they actually do more than just isolating them: they are in fact still informative, as unlikely classes are dropped anyway, and invite the domain expert (for instance, a doctor that has to issue the ultimate diagnosis) to avoid over-confident statements. We have also pointed out that in some extreme cases of missingness processes, the empirical evaluations made for naive Bayes can be completely biased, so that not even its average predictive accuracy can be evaluated reliably. In these cases, NCC2 was instead still reliably evaluated.

Dealing effectively with the hard instances is not an easy task in general, and we have shown that NBC fails on this a number of times by yielding unreliable classifications due to its inherent optimism. The way NCC2 sometimes fails on this is different, as it leads to an excess of caution; this is less critical although still not desirable. In these cases, the classifier is too pessimistic and the information used to build it up should be strengthened in order to obtain stronger conclusions. The more obvious way to do so with NCC2, and that—provided that one stays within the realm of tenable assumptions—is also what we recommend, is to minimize the number of missing values that are declared to be subject to an unknown missingness process. A less obvious but still very important way to do so is to simplify the model as much as possible, for example by doing feature selection. In fact we have conjectured that the same factors that can lead NBC to overfitting (e.g., high number of features or feature variables) can well result in excessive caution of NCC2. When also this is not enough, another approach may prove to be very helpful: using coarsened rather than missing values in data sets. Coarsened, or set-based, observations carry more information than missing ones in general because they naturally exclude some values as possible replacement for the missing information. Such extra information might be often available in real applications, and it seems to have the potential to lead to strong enough conclusions in a number of cases. It is also a kind of information that is typically provided by a domain expert rather than by the modeler (as opposed to MAR, for example) and as such it is a good candidate to be tenable.

More generally speaking, we regard the problem of providing a classifier with knowledge about the missingness process, which usually cannot be obtained from data, as a serious and an important topic of research. This seems to have much to do with identifying general and tenable assumptions about missingness processes of which classifiers could take advantage.

NCC2 can be regarded as the outcome of having made a first step in this direction. Doing such a step has led to a classifier that offers us a range of new opportunities to deal robustly with predictions in case of small or incomplete data sets: in particular, it allows us to check whether in very weak states of prior information we can draw strong enough conclusions for our aims, thus avoiding us to worry about doing mistaken assumptions; it lets us work incrementally towards stronger assumptions and conclusions; it enables us to uncover situations that otherwise could not be, not even experimentally. And it makes all of this possible by means of fast and exact computations.

Acknowledgments

We are grateful to all the reviewers for their constructive criticism, which has led us to expand and deepen the empirical analysis eventually making the paper gain insight, as well as to improve the paper at the level of presentation. Marco Zaffalon is also grateful to Peter Walley because of the joint work that led in 2001 to an inferential method for the naive credal classifier that was the starting point of the extension presented here. Work for this paper has been partially supported by the Swiss NSF grants 200021-113820/1 and 200020-116674/1, and by the Hasler Foundation (Hasler Stiftung) grant 2233.

Appendix A. A Polynomial-time Procedure for Incompleteness in the Instance to Classify

Although the procedure sketched in Section 3.4 solves Problem (8), a substantially more efficient procedure can be designed to solve the same problem. However, to avoid adding further complexity to the presentation of NCC2 carried out in the main body of the paper, we present it here, in Figure 6. The correctness of the procedure is stated by the following theorem, whose proof is in Appendix B.

Theorem 7 *The optimum of Problem* (8) *is obtained by the procedure in Figure 6.*

1. Build the set $\mathcal{U} := \{x_e = \frac{\overline{n}(a_{Me'},c''_M)\underline{n}(a_{Me''},c'_M)-\underline{n}(a_{Me'},c'_M)\overline{n}(a_{Me''},c''_M)}{s[\underline{n}(a_{Me'},c'_M)-\underline{n}(a_{Me''},c'_M)]} : e = 1,\dots,k', a_{Me'}, a_{Me''} \in \mathcal{A}_e, \underline{n}(a_{Me'},c'_M) \neq \underline{n}(a_{Me''},c'_M), x_e \in (0,1)\}$, where the lower and upper counts are defined as in Theorem 6.

2. Use the points in $\mathcal{U}$ to define a partition $\mathcal{J}$ of $(0,1)$.

3. For each interval $\mathcal{I} \in \mathcal{J}$, determine an associated tuple $(a^{\mathcal{I}}_{M1},\dots,a^{\mathcal{I}}_{Mk'}) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_{k'}$ by selecting, for each $e = 1,\dots,k'$ an element among those yielded by $\text{argmin}_{a^{\mathcal{I}}_{Me} \in \mathcal{A}_e} \frac{\underline{n}(a^{\mathcal{I}}_{Me},c'_M)}{\overline{n}(a^{\mathcal{I}}_{Me},c''_M)+x^{\mathcal{I}}}$, where $x^{\mathcal{I}}$ denotes the middle point of the interval $\mathcal{I}$.

4. For each of the intervals in $\mathcal{J}$, minimize the function defined by the tuple associated with the interval, by applying of Theorem 6 with suitable changes to adapt it to $\mathcal{J}$.

5. Take the minimum of the values provided by the previous step.

Figure 6: The solution procedure for Problem (8).

The intuitive idea behind the procedure is presented in the following. Let us assume that among the k attribute variables there are k' ($1 \leq k' \leq k$) that are not observed in the instance to classify. Assume, without loss of generality, that they are indexed from 1 to k'. The objective function of the problem can be re-written as (let $x := st(c''_M)$):

$$\inf_{0<x<s}\{[\frac{n(c''_M)+x}{n(c'_M)+s-x)}]^{k-1}\prod_{e=1}^{k'}[\min_{a_{Me}\in\mathcal{A}_e}\frac{\underline{n}(a_{Me},c'_M)}{\overline{n}(a_{Me},c''_M)+x}]\cdot$$
$$\cdot\prod_{j=k'+1}^{k}\frac{\underline{n}(a_{Mj},c'_M)}{\overline{n}(a_{Mj},c''_M)+x}\prod_{l=1}^{r'}[\frac{n_l(c''_M)+x}{n_l(c'_M)+s-x}\cdot\frac{n(\hat{a}_{Ml},c'_M)}{n(\hat{a}_{Ml},c''_M)+x}]\}, \tag{9}$$

where the functions $\underline{n}(a_{Me},c'_M)/[\overline{n}(a_{Me},c''_M)+x]$ can be minimized, given x, separately over each a_{Me}, as shown by the minima over $a_{M1},\cdots,a_{Mk'}$.
For variable A_e, the functions of this type obtained for each different value $a_{Me}\in\mathcal{A}_e$ are compared over the interval $[0,s]$; when the value a_{Me}, which minimizes them, changes, there is a so-called *partition point*. Now, let us consider the set $\mathcal{U}$, which collects the partition points of all the variables $A_1,\ldots,A_{k'}$: the tuple $\{a_1,\cdots a_{k'}\}$ which minimizes the first factor of Problem (9) changes over the interval $[0,s]$ at every point belonging to $\mathcal{U}$. Since the function is convex on each sub-interval, we can solve a minimization problem on each sub-interval; the global minimum is finally found by comparing the local minima obtained over each sub-interval.

Let us analyze briefly the computational complexity of the procedure. The procedure is based on the determination of the partition $\mathcal{J}$, whose size is at most $1+\sum_{e=1}^{k'}(|\mathcal{A}_e|^2-|\mathcal{A}_e|)/2$, which is also the number of minimizations in the worst case. Each minimization is done according to Theorem 6. This is based on the computation of the logarithmic derivatives of the function of interest. Each evaluation of the logarithmic derivative is a task linear in $k+r'$, where we recall that r' is the number of non-missing MAR variables in the instance to classify. If we regard the complexity of Newton-Raphson's method as a constant, which is reasonable as it is an extremely fast algorithm, we obtain that the procedure works in time $O([1+\sum_{e=1}^{k'}(|\mathcal{A}_e|^2-|\mathcal{A}_e|)/2](k+r'))$. To obtain a simpler expression, we can focus on an upper bound: $O(\overline{\mathcal{A}}^2(k+r')^2)$, where $\overline{\mathcal{A}}:=\text{argmax}_{e=1}^{k'}|\mathcal{A}_e|$. The complexity is then roughly quadratic in the number of attribute variables and quadratic also in the worst-case number of attributes per variable.

Appendix B. Proofs

Proof [Lemma 1] The likelihood is easily re-written as $\prod_{i=1}^{N}\vartheta_{(d_i,\hat{x}_i\in\hat{o}_i)}$ thanks to the IID assumption, where $\vartheta_{(d_i,\hat{x}_i\in\hat{o}_i)}:=\sum_{\hat{x}_i\in\hat{o}_i}\vartheta_{(d_i,\hat{x}_i)}$. By Assumption (2) we have that

$$\begin{aligned}
p(d,\hat{x}\in\hat{o}|\vartheta) &= \prod_{i=1}^{N}[\sum_{\hat{x}_i\in\hat{o}_i}\vartheta_{(d_i,\hat{x}_i)}]\\
&= \prod_{i=1}^{N}(\sum_{\hat{a}_{i1}\in\hat{o}_{i1}}\cdots\sum_{\hat{a}_{ir}\in\hat{o}_{ir}}\vartheta_{c_i}\prod_{j=1}^{k}\vartheta_{a_{ij}|c_i}\prod_{l=1}^{r}\vartheta_{\hat{a}_{il}|c_i})\\
&= \prod_{i=1}^{N}\{\vartheta_{c_i}[\prod_{j=1}^{k}\vartheta_{a_{ij}|c_i}][\prod_{l=1}^{r}\sum_{\hat{a}_{il}\in\hat{o}_{il}}\vartheta_{\hat{a}_{il}|c_i}]\}.
\end{aligned}$$

Let us focus on $\prod_{i=1}^{N}\vartheta_{c_i}$. By grouping the same chances over the N units, that expression can be re-written as $\prod_{c\in\mathcal{C}}\vartheta_c^{n(c)}$, where $n(c)$ denotes the number of occurrences of c in d. In the same way, we re-write $\prod_{i=1}^{N}\vartheta_{a_{ij}|c}$ for all j and a given c as $\prod_{a_j\in\mathcal{A}_j}\vartheta_{a_j|c}^{n(a_j,c)}$, where $n(a_j,c)$ denotes the number of joint occurrences (a_j,c) in d. Now consider $\sum_{\hat{a}_{il}\in\hat{o}_{il}}\vartheta_{\hat{a}_{il}|c}$, for a given c. Whenever $\hat{o}_{il}$ coincides with $\hat{\mathcal{A}}_l$, $\sum_{\hat{a}_{il}\in\hat{o}_{il}}\vartheta_{\hat{a}_{il}|c}=1$; and in all the other cases the sum degenerates to a single term, as the observation of the variable value has been precise. Indeed, as clarified in Section 2, according to the definition of missing data we either observe a value exactly, or we do not observe it at all.

This means that we are allowed to drop the missing values from the learning set. In other words, $\prod_{i=1}^{N}(\sum_{\hat{a}_{il}\in\hat{o}_{il}}\vartheta_{\hat{a}_{il}|c})$ becomes, for all l, $\prod_{\hat{a}_l\in\hat{\mathcal{A}}_l}\vartheta_{\hat{a}_l|c}^{n(\hat{a}_l,c)}$, where $n(\hat{a}_l,c)$ denotes the number of joint occurrences $(\hat{a}_l,c)$ in the learning set after dropping the units with missing values of $\hat{A}_l$. ■

We skip the proof of Lemma 2 as it follows a pattern analogous (and easier) to that of the previous proof.

Proof [Theorem 3] Lemmas 1 and 2, together with the chosen prior in (4), allow us to re-write $p(c_M|d^{\text{-}},\hat{x}^{+} \in \hat{o}^{+})$ in Equation (3), that is, $p(c_M|d^{\text{-}},\hat{x}^{+} \in \hat{o}^{+},s,t)$, as follows:

$$
\begin{aligned}
& p(c_M|d^{\text{-}},\hat{x}^{+} \in \hat{o}^{+},s,t) \propto \\
& \int_{\Theta_C} \vartheta_{c_M} \prod_{c \in \mathcal{C}} \vartheta_c^{n(c)+st(c)-1} d\vartheta_C \\
& \cdot \prod_{j=1}^{k} \int_{\Theta_{A_j|c_M}} \vartheta_{a_{Mj}|c_M} \prod_{a_j \in \mathcal{A}_j} \vartheta_{a_j|c_M}^{n(a_j,c_M)+st(a_j,c_M)-1} d\vartheta_{A_j|c_M} \\
& \cdot \prod_{\substack{l=1 \\ \hat{o}_{Ml} \neq \hat{\mathcal{A}}_l}}^{r} \int_{\Theta_{\hat{A}_l|c_M}} \vartheta_{\hat{a}_{Ml}|c_M} \prod_{\hat{a}_l \in \hat{\mathcal{A}}_l} \vartheta_{\hat{a}_l|c_M}^{n(\hat{a}_l,c_M)+st(\hat{a}_l,c_M)-1} d\vartheta_{\hat{A}_l|c_M}.
\end{aligned}
\tag{10}
$$

Expression (10) is obtained by the following steps: (i) substituting in (3) the expressions for the prior, the likelihood and the probability of the next class obtained in Section 3.1; (ii) observing that the Dirichlet densities involved in the integration are independent, so that they can be integrated separately; (iii) dropping the integrals related to classes different from c_M, as each of them equals 1 given that it is missing the term of which to take the expectation; and (iv), for similar reasons, dropping the integrals related to missing attribute values in the unit to classify. Observe that in (10) we have used the following notation: we have denoted by $\vartheta_C \in \Theta_C$ the vector of chances ϑ_c, $c \in \mathcal{C}$. Similarly, $\vartheta_{A_j|c_M} \in \Theta_{A_j|c_M}$ (for all $j = 1,\dots,k$) resp. $\vartheta_{\hat{A}_l|c_M} \in \Theta_{\hat{A}_l|c_M}$ (for all $l = 1,\dots,r$) denote the vector of chances $\vartheta_{a_j|c_M}$, $a_j \in \mathcal{A}_j$, resp. $\vartheta_{\hat{a}_l|c_M}$, $\hat{a}_l \in \hat{\mathcal{A}}_l$.

Each integral in (10) represents the expectation of a chance with respect to a Dirichlet density. It is well known that such a calculation can be solved exactly (Kotz et al., 2000, Chap. 49), leading to the statement of the theorem. ■

Proof [Lemma 4] Using (5) we first re-write the function to optimize as follows:

$$
\begin{aligned}
& [\frac{n(c''_M)+st(c''_M)}{n(c'_M)+st(c'_M)}]^{k-1} \prod_{j=1}^{k} \frac{n(a_{Mj},c'_M)+st(a_{Mj},c'_M)}{n(a_{Mj},c''_M)+st(a_{Mj},c''_M)} \cdot \\
& \cdot \prod_{l=1}^{r'} [\frac{n_l(c''_M)+st(c''_M)}{n_l(c'_M)+st(c'_M)} \cdot \frac{n(\hat{a}_{Ml},c'_M)+st(\hat{a}_{Ml},c'_M)}{n(\hat{a}_{Ml},c''_M)+st(\hat{a}_{Ml},c''_M)}].
\end{aligned}
$$

Then we simplify the optimization problem according to the following considerations.

- The objective function is only concerned with the r' non-missing MAR attribute variables.
- For each $j \in \{1,\dots,k\}$ and $c \in \mathcal{C}$, the objective function contains only one term $t(a_j,c)$. We can choose its value freely in the interval $[0,t(c)]$, as the constraints of type $\sum_{a_j \in \mathcal{A}_j} t(a_j,c) = t(c)$ are always satisfied by assigning the value $[t(c)-t(a_j,c)]$ to any term $t(a_{j'},c)$ ($j' \in \{1,\dots,r\}$, $j' \neq j$) that does not appear in the objective function, and zero to the others. It follows that constraints of type $\sum_{a_j \in \mathcal{A}_j} t(a_j,c) = t(c)$ can be re-written as $t(a_j,c) \leq t(c)$ for all $j \in \{1,\dots,k\}$, $c \in \mathcal{C}$. Analogous considerations hold with constraints of type $\sum_{\hat{a}_l \in \hat{\mathcal{A}}_l} t(\hat{a}_l,c) = t(c)$.

- The optimum of the problem is obtained when all the terms $t(a_{Mj},c'_M)$ and $t(\hat{a}_{Ml},c'_M)$ go to zero, and all the terms $t(a_{Mj},c''_M)$ and $t(\hat{a}_{Ml},c''_M)$ go to $t(c''_M)$. This follows from the preceding observation and because the objective function is made of non-negative terms. Therefore, in the objective function we can replace the mentioned terms with the corresponding values at the optimum.

- Constraint $\sum_{c\in\mathcal{C}} t(c) = 1$ can be replaced by $t(c_{M'})+t(c_{M''}) = 1$, because the objective function is only concerned with those two classes, as because such a constraint naturally holds at a global optimum. Suppose the last statement is false, that is, that $t(c_{M'})+t(c_{M''}) < 1$ at a global optimum point. Then we might keep $t(c_{M''})$ fixed and increase $t(c_{M'})$ up to $1-t(c_{M''})$ (making the $t(\cdot)$ terms of the remaining classes go to zero), so decreasing the optimum value.

These lead to the statement of the lemma once we also remove the variable $t(c'_M)$ from consideration. ■

Proof [Theorem 5] To demonstrate Theorem 5, it is necessary first to introduce two lemmas, whose proofs are given later.

Lemma 8 *Consider Problem* (6)*, assuming that the following two conditions hold: (i) there is no j such that $n(a_{Mj},c'_M)=0$ nor l such that $n(\hat{a}_{Ml},c'_M)=0$; and (ii) $k+r'>0$. Then the objective function $h(t(c_{M''}))$ is (strictly) convex over* $(0,1)$.

Lemma 9 *Consider Problem* (6)*, and the same assumption used in Lemma 8. Then the logarithmic derivative of the objective function in $t(c''_M)=0$ is $-\infty$ if and only if any of the following cases hold: $n(c''_M)=0$; there is $j\in\{1,\dots,k\}$ s.t. $n(a_{Mj},c''_M)=0$; there is $l\in\{1,\dots,r'\}$ s.t. $n(\hat{a}_{Ml},c''_M)=0$ and $n_l(c''_M)\neq 0$.*

Having introduced Lemmas 8 and 9, we can now prove Theorem 5. The first two steps of the procedure deal with special cases. The first step is justified since in the stated conditions the objective function attains the value zero, and this is a global minimum because the function is non-negative. The second step considers a degenerate problem of classification, in which there are no attribute variables; in this case the minimum is at $t(c_{M''})=1$, as the objective function is strictly decreasing, as it can be shown by differentiating.

Otherwise, the function is convex and the local minimum is also the global minimum. The minimum is found by the following procedure:

1. If:

   - $n(c''_M)=0$ or
   - there is $j\in\{1,\dots,k\}$ s.t. $n(a_{Mj},c''_M)=0$ or
   - there is $l\in\{1,\dots,r'\}$ s.t. $n(\hat{a}_{Ml},c''_M)=0$ and $n_l(c''_M)\neq 0$,

   then let $(\ln h(t(c''_M)))'|_{t(c''_M)=0} := -\infty$, else compute $(\ln h(t(c''_M)))'|_{t(c''_M)=0}$.

2. Compute $(\ln h(t(c''_M)))'|_{t(c''_M)=1}$.

3. If $(\ln h(t(c''_M)))'|_{t(c''_M)=0} \geq 0$, let $\inf_{0<t(c''_M)<1} h(t(c''_M)) := h(0)$. Stop.

4. If $(\ln h(t(c''_M)))'|_{t(c''_M)=1} \leq 0$, let $\inf_{0<t(c''_M)<1} h(t(c''_M)) := h(1))$. Stop.
5. If $(\ln h(t(c''_M)))'|_{t(c''_M)=0} < 0$ and $(\ln h(t(c''_M)))'|_{t(c''_M)=1} > 0$, approximate the minimum numerically by Newton-Raphson's algorithm as in p. 366 of Press et al. (1993).

The procedure is basically a test based on the values of the logarithmic derivative at the extremes of the interval $[0,1]$. If the function were bounded, the last three points would obviously identify the minimum. The preceding points allow the test to be extended to the case of $h(0)$ being unbounded.[18] This happens according to the conditions stated in Lemma 9. We account for these cases by introducing the value $-\infty$ for the derivative in the solution procedure, and treating it as one of the possible values.

As far as Newton-Raphson's method is concerned, it can be applied as the first and second derivatives are available. Note that if the Newton-Raphson is combined with bracketing as in the cited implementation, then its convergence is guaranteed. ■

Proof [Lemma 8] Consider the following definitions to simplify the notation: $\alpha_j := n(a_{Mj}, c'_M)$, $\beta_j := n(a_{Mj}, c''_M)$, $\tilde{\alpha} := n(c'_M)$, $\tilde{\beta} := n(c''_M)$, $\gamma_l := n(\hat{a}_{Ml}, c'_M)$, $\delta_l := n(\hat{a}_{Ml}, c''_M)$, $\tilde{\gamma}_l := n_l(c'_M)$, $\tilde{\delta}_l := n_l(c''_M)$, $x := st(c''_M)$. Re-write Problem (6) accordingly as

$$\begin{aligned}&\inf_{0<x<s} (\frac{\tilde{\beta}+x}{\tilde{\alpha}+s-x})^{k-1} \prod_{j=1}^{k} \frac{\alpha_j}{\beta_j+x} \prod_{l=1}^{r'} (\frac{\tilde{\delta}_l+x}{\tilde{\gamma}_l+s-x} \cdot \frac{\gamma_l}{\delta_l+x})\\ &= \inf_{0<x<s} h(x).\end{aligned}$$

The objective function is positive on the domain of definition, so we can compute the logarithmic derivative of $h(\cdot)$:

$$\begin{aligned}&\frac{d \ln h(x)}{dx} =\\ &= \frac{k-1}{\tilde{\beta}+x} + \frac{k-1}{\tilde{\alpha}+s-x} - \sum_{j=1}^{k} \frac{1}{\beta_j+x} + \sum_{l=1}^{r'} \frac{1}{\tilde{\delta}_l+x} +\\ &- \sum_{l=1}^{r'} \frac{1}{\delta_l+x} + \sum_{l=1}^{r'} \frac{1}{\tilde{\gamma}_l+s-x}.\end{aligned} \tag{11}$$

Another differentiation leads to the following expression:

$$\begin{aligned}&\frac{d^2 \ln h(x)}{dx^2} =\\ &= -\frac{k-1}{(\tilde{\beta}+x)^2} + \frac{k-1}{(\tilde{\alpha}+s-x)^2} + \sum_{j=1}^{k} \frac{1}{(\beta_j+x)^2} +\\ &- \sum_{l=1}^{r'} \frac{1}{(\tilde{\delta}_l+x)^2} + \sum_{l=1}^{r'} \frac{1}{(\delta_l+x)^2} + \sum_{l=1}^{r'} \frac{1}{(\tilde{\gamma}_l+s-x)^2}.\end{aligned} \tag{12}$$

[18] Note that the function is always defined in $t(c_{M''}) = 1$ as the cases that arise when either $n(c'_M) = 0$ or when there is l such that $n_l(c'_M) = 0$, are ruled out by the initial assumption prescribing that there is neither j such that $n(a_{Mj}, c'_M) = 0$, nor l such that $n(\hat{a}_{Ml}, c'_M) = 0$.

There are three possible cases.

1. $k = 0$ and $r' > 0$. In Expression (12), the sum on j disappears, and the other terms are positive except for two of them: $-\frac{1}{(\tilde{\alpha}+s-x)^2}$ and $-\sum_{l=1}^{r'} \frac{1}{(\tilde{\delta}_l+x)^2}$.

   Consider the first one. By definition, $\tilde{\gamma}_l \leq \tilde{\alpha}$ for each l, so that $\frac{1}{(\tilde{\alpha}+s-x)^2} \leq \frac{1}{(\tilde{\gamma}_l+s-x)^2}$, whence $-\frac{1}{(\tilde{\alpha}+s-x)^2} + \sum_{l=1}^{r'} \frac{1}{(\tilde{\gamma}_l+s-x)^2} \geq 0$.

   Now consider the second negative term in Expression (12). By definition, it holds that $\delta_l \leq \tilde{\delta}_l$ for each l, whence $\sum_{l=1}^{r'} \frac{1}{(\tilde{\delta}_l+x)^2} \leq \sum_{l=1}^{r'} \frac{1}{(\delta_l+x)^2}$, or, in other words, $-\sum_{l=1}^{r'} \frac{1}{(\tilde{\delta}_l+x)^2} + \sum_{l=1}^{r'} \frac{1}{(\delta_l+x)^2} \geq 0$.

   This shows that the negative terms in Expression (12) together with the considered positive ones produce non-negative results. Since there are positive terms left in Expression (12), the overall expression is positive. In other words, the function $\ln h(\cdot)$ is (strictly) convex, and so is $h(\cdot)$ (by applying the exponential function).

2. $k > 0$ and $r' = 0$. In this case, there is only one negative term in Expression (12): $-\frac{k-1}{(\tilde{\beta}+x)^2}$. By definition, $\beta_j \leq \tilde{\beta}$ for each j, so that $-\frac{k-1}{(\tilde{\beta}+x)^2} + \sum_{j=1}^{k} \frac{1}{(\beta_j+x)^2} \geq 0$. Since there are positive terms left in Expression (12), the function $h(\cdot)$ is (strictly) convex.

3. $k > 0$ and $r' > 0$. In this case there are two negative terms in Expression (12) (the others being positive): $-\frac{k-1}{(\tilde{\beta}+x)^2}$ and $-\sum_{l=1}^{r'} \frac{1}{(\tilde{\delta}_l+x)^2}$. We have already shown in the two preceding cases that $-\sum_{l=1}^{r'} \frac{1}{(\tilde{\delta}_l+x)^2} + \sum_{l=1}^{r'} \frac{1}{(\delta_l+x)^2} \geq 0$ and that $-\frac{k-1}{(\tilde{\beta}+x)^2} + \sum_{j=1}^{k} \frac{1}{(\beta_j+x)^2} \geq 0$. As there are positive terms left in Expression (12), the overall expression is positive. Also in this case, the function $h(\cdot)$ is (strictly) convex.

■

Proof [Lemma 9] Consider Expression (11) for the logarithmic derivative, and the notation introduced there. Let us focus on the expression obtained from Expression (11) by dropping the terms that are bounded when x approaches zero:

$$\frac{k-1}{\tilde{\beta}+x} - \sum_{j=1}^{k} \frac{1}{\beta_j + x} + \sum_{l=1}^{r'} \frac{1}{\tilde{\delta}_l + x} - \sum_{l=1}^{r'} \frac{1}{\delta_l + x}. \tag{13}$$

Of course Expression (11) goes to $-\infty$ when x approaches zero if and only if Expression (13) does the same, so that we can concentrate on the latter.

($\Leftarrow$) Let us now show that if any of the conditions in the statement holds, Expression (13) goes to $-\infty$ with x approaching zero.

First, consider the case when $\tilde{\beta} = n(c''_M) = 0$, and suppose that $k = 0$. The first term in (13) becomes $-1/x$, the second disappears, and the last two terms sum up to zero. Indeed, when $\tilde{\beta} = 0$, $\tilde{\delta}_l = \delta_l = 0$. The expression becomes $-1/x$, and the implication is verified. In the case when $r' = 0$, it must be $k > 0$ and hence the first two terms in (13) yield $-1/x$ as $\beta_j = 0$ for all j when $\tilde{\beta} = 0$; and the last two terms disappear. Again, the expression becomes $-1/x$, and the implication is verified. The case when both $k > 0$ and $r' > 0$ is verified on the basis of the previous two cases.

Second, consider the case when there is $j \in \{1,\ldots,k\}$ s.t. $\beta_j = n(a_{Mj}, c''_M) = 0$. If also $\tilde{\beta} = 0$, we would fall in the previous case and the implication would be true, so we can assume that $\tilde{\beta} > 0$. In the case when $r' = 0$, the implication would follow immediately. Consider $r' > 0$. The only possibility for Expression (13) not to go to $-\infty$ with x going to zero, is that there is l s.t. $\tilde{\delta}_l = 0$. In this way the term $-1/(\beta_j + x) = -1/x$ would sum up to zero together with the term $1/(\tilde{\delta}_l + x) = 1/x$. But this is not possible: in fact, $\tilde{\delta}_l = 0$ implies $\delta_l = 0$, so that $1/(\tilde{\delta}_l + x) = 1/x$ cancels out together with $-1/(\delta_l + x) = -1/x$. The implication is true also in this case.

Finally, consider the case when there is $l \in \{1,\ldots,r'\}$ s.t. $\delta_l = n(\hat{a}_{Ml}, c''_M) = 0$ and $\tilde{\delta}_l = n_l(c''_M) \neq 0$. Observe that the term $-1/(\delta_l + x) = -1/x$ cannot cancel out because of other terms: in fact, with the l-th terms, we have that $\tilde{\delta}_l > 0$; if there is another l' s.t. $\tilde{\delta}_l = 0$, also $\delta_l = 0$ and the related two terms sum up to zero. The only remaining possibility to cancel $-1/(\delta_l + x) = -1/x$ out is that $\tilde{\beta} = 0$. But in this case we know that the implication must be true. It follows that also in this case the implication is verified.

($\Rightarrow$) Let us now consider the reverse implication. If Expression (13) goes to $-\infty$ with x going to zero, at least one of the terms in such an expression must do the same. By considering the terms one by one, the implication is proved in a trivial way. ■

Proof [Theorem 6]

Problem

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

where the set of prior densities is defined according to Expression (4), the constraints are those described in Section 3.1.1, and the probabilities in the function to optimize are defined as in (5), is equivalent to the following:

$$\min_{d \in o} \inf_{0 < t(c''_M) < 1} \{ [\frac{n(c''_M) + st(c''_M)}{n(c'_M) + s - st(c''_M)}]^{k-1} \prod_{j=1}^{k} \frac{n(a_{Mj}, c'_M)}{n(a_{Mj}, c''_M) + st(c''_M)} \cdot$$
$$\cdot \prod_{l=1}^{r'} [\frac{n_l(c''_M) + st(c''_M)}{n_l(c'_M) + s - st(c''_M)} \cdot \frac{n(\hat{a}_{Ml}, c'_M)}{n(\hat{a}_{Ml}, c''_M) + st(c''_M)}] \}.$$

The problem can be then re-written as follows:

$$\inf_{0 < t(c''_M) < 1} \min_{d \in o} \{ [\frac{n(c''_M) + st(c''_M)}{n(c'_M) + s - st(c''_M)}]^{k-1} \prod_{j=1}^{k} \frac{n(a_{Mj}, c'_M)}{n(a_{Mj}, c''_M) + st(c''_M)} \cdot$$
$$\cdot \prod_{l=1}^{r'} [\frac{n_l(c''_M) + st(c''_M)}{n_l(c'_M) + s - st(c''_M)} \cdot \frac{n(\hat{a}_{Ml}, c'_M)}{n(\hat{a}_{Ml}, c''_M) + st(c''_M)}] \}. \tag{14}$$

This is obtained by inverting the order of the optimizations and using Expression (6). Note that the inner minimization only affects the product over j in the objective function. Moreover, the counts in such a product attain the same value at the optimum for any choice of $t(c''_M)$, leading to the following product: $\prod_{j=1}^{k} \underline{n}(a_{Mj}, c'_M) / [\overline{n}(a_{Mj}, c''_M) + st(c''_M)]$. Note that the pairwise counts are optimized separately as they do not affect each other. These arguments enable us to reduce

Problem (14) as:

$$\inf_{0<t(c''_M)<1}\Big\{\left[\frac{n(c''_M)+st(c''_M)}{n(c'_M)+s-st(c''_M)}\right]^{k-1}\prod_{j=1}^{k}\frac{\underline{n}(a_{Mj},c'_M)}{\overline{n}(a_{Mj},c''_M)+st(c''_M)}\cdot \tag{15}$$

$$\cdot\prod_{l=1}^{r'}\left[\frac{n_l(c''_M)+st(c''_M)}{n_l(c'_M)+s-st(c''_M)}\cdot\frac{n(\hat{a}_{Ml},c'_M)}{n(\hat{a}_{Ml},c''_M)+st(c''_M)}\right]\Big\},$$

with $\underline{n}(a_{Mj},c'_M):=\min_{d\in o}n(a_{Mj},c'_M)$ and $\overline{n}(a_{Mj},c'_M):=\max_{d\in o}n(a_{Mj},c'_M)$. At this point, Problem (15) is an instance of Problem (6) obtained for a specific choice of a complete data set d in o. ■

Proof [Theorem 7] Let us focus the attention on the term $\min_{a_{Me}\in\mathcal{A}_e}\frac{\underline{n}(a_{Me},c'_M)}{\overline{n}(a_{Me},c''_M)+x}$. As a function of x, this is the lower envelope of the set of functions $\{\frac{\underline{n}(a_{Me},c'_M)}{\overline{n}(a_{Me},c''_M)+x}:a_{Me}\in\mathcal{A}_e\}=:\mathcal{F}_e$. Consider a value x such that the lower envelope coincides with two different functions in $\mathcal{F}_e$ before and after x. This implies that the two functions must cross at x, because the function that was over the other before x must be under it after x. In other words, the set of points where any two different functions in $\mathcal{F}_e$ cross (let us call them *partition points*), contains the points where the function matched by the lower envelope changes.

We can identify the partition points in the following way. Consider any two different functions in $\mathcal{F}_e$: $\underline{n}(a_{Me'},c'_M)/[\overline{n}(a_{Me'},c''_M)+x]$ and $\underline{n}(a_{Me''},c'_M)/[\overline{n}(a_{Me''},c''_M)+x]$. Assume that $\underline{n}(a_{Me'},c'_M)\neq\underline{n}(a_{Me''},c'_M)$: in the opposite case the only way for the two functions to cross would be that also $\overline{n}(a_{Me'},c''_M)=\overline{n}(a_{Me''},c''_M)$, but the two functions would be the same, which is excluded a priori. The two functions cross when $\underline{n}(a_{Me'},c'_M)/[\overline{n}(a_{Me'},c''_M)+x]=\underline{n}(a_{Me''},c'_M)/[\overline{n}(a_{Me''},c''_M)+x]$, and this happens if and only if

$$x=\frac{\overline{n}(a_{Me'},c''_M)\underline{n}(a_{Me''},c'_M)-\underline{n}(a_{Me'},c'_M)\overline{n}(a_{Me''},c''_M)}{\underline{n}(a_{Me'},c'_M)-\underline{n}(a_{Me''},c'_M)}. \tag{16}$$

In other words, there is at most one partition point for any two different functions,[19] and we can identify it easily by (16). The maximum number of partition points is also the number of distinct pairs of different functions in $\mathcal{F}_e$, which is at most $(|\mathcal{A}_e|^2-|\mathcal{A}_e|)/2$.[20]

The crucial observation here is that the lower envelope matches a single function of $\mathcal{F}_e$ in any sub-interval of $(0,s)$ that does not contain partition points; and we are always able to select an element of $\mathcal{A}_e$ that gives rise to the matched function. In other words, the partition points can be used to define a partition of $(0,s)$, in the sub-intervals of which the lower envelop matches a single function that we are able to characterize by an element of $\mathcal{A}_e$. This argument is easily extended to the entire product over e in (9): we compute the set of partition points for each term of the product, and take their union. The union defines a partition $\mathcal{J}$ of $(0,s)$, in which every sub-interval $\mathcal{I}$ is associated with a single tuple $(a^{\mathcal{I}}_{M1},\dots,a^{\mathcal{I}}_{Mk'})\in\mathcal{A}_1\times\cdots\times\mathcal{A}_{k'}$, so that

$$\prod_{e=1}^{k'}[\min_{a_{Me}\in\mathcal{A}_e}\frac{\underline{n}(a_{Me},c'_M)}{\overline{n}(a_{Me},c''_M)+x}]=\prod_{e=1}^{k'}\frac{\underline{n}(a^{\mathcal{I}}_{Me},c'_M)}{\overline{n}(a^{\mathcal{I}}_{Me},c''_M)+x}.$$

[19] Note that it could be outside $[0,s]$.

[20] Note that some partition points may be such that the lower envelope does not change the matched function in $\mathcal{F}_e$, and so they could be discarded. As an example, assume that the value x identified by (16) is a partition point only for the two mentioned functions, and that there is a third function below the other two at x. In this case the former two functions are not involved in the determination of the lower envelope at x.

Problem (9) then becomes the following:

$$\begin{aligned}
&\min_{\mathcal{I}\in\mathcal{J}} \inf_{x\in\mathcal{I}} \{[\frac{n(c''_M)+x}{n(c'_M)+s-x}]^{k-1} \prod_{e=1}^{k'} \frac{\underline{n}(a^{\mathcal{I}}_{Me},c'_M)}{\overline{n}(a^{\mathcal{I}}_{Me},c''_M)+x} \cdot \\
&\cdot \prod_{j=k'+1}^{k} \frac{\underline{n}(a_{Mj},c'_M)}{\overline{n}(a_{Mj},c''_M)+x} \cdot \\
&\cdot \prod_{l=1}^{r'} [\frac{n_l(c''_M)+x}{n_l(c'_M)+s-x} \cdot \frac{n(\hat{a}_{Ml},c'_M)}{n(\hat{a}_{Ml},c''_M)+x}]\},
\end{aligned}$$

where the inner optimization can be solved by the procedure given in Section (3.2), applying it to the interval $\mathcal{I}$ rather than $(0,s)$.

Now it is easy to show that the procedure in Fig. 6 solves Problem (8). The first step of the procedure builds the set of partition points for the functions in $\mathcal{F}_e$, and the next one defines the partition of $(0,s)$. Since the function is convex on each sub-interval, the remaining steps solve a minimization problem on each sub-interval; finally, the global minimum is selected. ■

Appendix C. Experimental Results Data Set by Data Set

Detailed results by data set are shown in Tables 6 and 7, which refer respectively to the MAR and non-MAR setting.

Data set	*NBC*	*NCC2-MAR*				*NCC2-nonMAR*				*Subsets of instances*				
	Acc. (%)	Det. (%)	Single-Acc. (%)	SetAcc. (%)	Indet.OutSize	Det. (%)	Single-Acc. (%)	SetAcc. (%)	Indet.OutSize	NCC2-MAR D (%)	NCC2-MAR I (%)	NCC2-nonMAR D (%)	NCC2-nonMAR I (%)	ΔNCC2-nonMAR (%)
ecoli (8 cl.)	85.0	84.2	88.5	92.2	3.8	54.4	95.0	96.6	5.2	88.5	66.6 (6.4)	95.0	72.9 (3.4)	74.7 (4.3)
glass (6 cl.)	67.5	52.1	80.4	79.5	2.4	22.4	81.9	94.8	3.7	80.4	53.6 (3.9)	81.9	63.3 (2.6)	61.6 (4.9)
haberman (2 cl.)	72.0	91.8	74.2	100.0	2.0	66.7	80.8	100.0	2.0	74.2	47 (10.9)	80.8	54.1 (4.8)	56.5 (6.2)
kr-kp (2 cl.)	86.8	98.4	87.4	100.0	2.0	5.4	100.0	100.0	2.0	87.4	48 (10.2)	100.0	86.1 (0.4)	86.7 (0.4)
letter (26 cl.)	72.4	91.6	77.0	57.7	2.7	13.3	99.0	97.9	18.7	77.0	22.0 (1.2)	99.0	68.3 (0.2)	68.4 (0.2)
monks1 (2 cl.)	70.0	87.4	72.7	100.0	2.0	39.7	82.5	100.0	2.0	72.7	51.2 (5.8)	82.5	61.6 (3.0)	64.3 (3.6)
monks2 (2 cl.)	62.2	86.6	63.4	100.0	2.0	15.3	81.2	100.0	2.0	63.4	54.6 (6.8)	81.2	58.9 (2.2)	59.7 (2.3)
monks3 (2 cl.)	95.3	99.2	95.5	100.0	2.0	67.3	96.7	100.0	2.0	95.6	64 (25)	96.6	91.7 (3.2)	92.5 (3.1)
nursery (5 cl.)	87.0	97.5	87.4	78.9	2.0	49.0	98.4	99.6	2.5	87.4	66.1 (5.3)	98.4	76.0 (1.0)	76.0 (0.9)
optdigits (10 cl.)	91.2	95.2	93.5	86.9	2.7	14.8	99.9	99.9	8.0	93.5	46.9 (3.6)	99.9	89.7 (0.2)	89.8 (0.2)
pendigits (10 cl.)	87.2	96.3	89.5	81.8	2.5	29.2	97.1	99.1	7.2	89.5	26.7 (2.7)	97.1	83.1 (0.3)	83.2 (0.3)
segment (7 cl.)	91.0	89.3	95.6	96.7	3.7	31.7	98.8	99.9	6.1	95.6	52.5 (3.9)	98.8	87.4 (0.7)	88.7 (0.7)
sonar (2 cl.)	84.4	90.9	86.9	100.0	2.0	41.8	97.9	100.0	2.0	86.9	59 (14.3)	97.9	74.6 (3.1)	77.4 (2.6)
spambase (2 cl.)	89.1	99.5	89.4	100.0	2.0	42.1	97.2	100.0	2.0	89.4	31 (13.4)	97.2	83.3 (0.5)	83.7 (0.4)
spect (2 cl.)	76.1	90.8	79.3	100.0	2.0	53.9	90.2	100.0	2.0	79.3	44 (12.6)	90.2	59.8 (4.1)	63.7 (4.3)
splice (3 cl.)	94.5	97.2	96.4	96.2	2.2	0.0	100.0	100.0	3.0	96.4	49.5 (7.6)	100.0	94.9 (0.2)	94.5 (0.2)
waveform (3 cl.)	81.3	99.0	81.6	99.9	2.0	19.7	94.5	100.0	2.3	81.6	50.3 (7.6)	94.5	78.0 (0.4)	78.2 (0.2)
yeast (10 cl.)	56.6	91.9	58.6	70.0	2.3	28.0	69.1	91.1	3.7	58.5	34.9 (6.7)	69.1	51.7 (1.7)	51.7 (1.7)

Table 6: Results data set by data set under the MAR-MP setting. For each data set, the number of classes is reported aside the name. For NccMarI, NccI, ΔNCC2-nonMAR, the standard deviation is reported into brackets; for the remaining indicators, the standard deviation is smaller than 2 percentage points in the large majority of cases.

Data set	*NBC*	*NCC2-MAR*				*NCC2-nonMAR*				*Subsets of instances*				
	Acc. (%)	Det. (%)	Single-Acc. (%)	SetAcc. (%)	Indet.OutSize	Det. (%)	Single-Acc. (%)	SetAcc. (%)	Indet.OutSize	NCC2-MAR D (%)	NCC2-MAR I (%)	NCC2-nonMAR D (%)	NCC2-nonMAR I (%)	ΔNCC2-nonMAR (%)
ecoli (8cl.)	85.2	83.2	88.9	92.6	3.7	69.6	94.9	94.1	3.8	88.9	67.2 (5.7)	94.9	62.9 (3.0)	60.4 (6.7)
glass (6 cl.)	67.5	47.5	81.7	79.7	2.4	29.6	83.5	91.1	3.0	81.7	54.8 (3.4)	83.5	60.7 (2.2)	58.7 (6.9)
haberman (2 cl.)	72.0	93.5	73.8	100.0	2.0	79.4	78.5	100.0	2.0	73.8	47 (12.2)	78.5	46.6 (5.7)	47.1 (9.6)
kr-kp (2 cl.)	88.0	98.5	88.6	100.0	2.0	52.9	96.8	100.0	2.0	88.6	49.1 (9.4)	96.8	78.1 (1.0)	79.0 (0.9)
letter (26 cl.)	72.9	91.9	77.4	58.5	2.8	35.3	96.7	91.8	11.7	77.4	22.0 (1.1)	96.7	59.9 (0.3)	60.0 (0.3)
monks1 (2 cl.)	71.0	87.5	73.7	100.0	2.0	58.8	80.1	100.0	2.0	73.7	51.8 (5.5)	80.1	57.9 (2.4)	60.6 (3.2)
monks2 (2 cl.)	62.2	86.7	63.5	100.0	2.0	40.2	72.5	100.0	2.0	63.5	53.5 (6.0)	72.5	55.6 (2.2)	56.3 (2.5)
monks3 (2 cl.)	95.5	99.8	95.6	100.0	2.0	90.5	97.2	100.0	2.0	96.3	64 (31.3)	87.6	73 (14.1)	73 (14.1)
nursery (5 cl.)	87.8	97.5	88.3	80.3	2.0	68.8	97.2	99.6	2.3	88.3	67.8 (4.5)	97.2	67.1 (1.5)	66.9 (1.2)
optdigits (10 cl.)	91.3	95.3	93.5	86.2	2.6	54.6	99.0	98.4	4.9	93.5	45.2 (3.1)	99.0	82.0 (0.4)	82.4 (0.4)
pendigits (10 cl.)	87.4	96.3	89.7	81.8	2.5	57.1	95.5	96.8	5.4	89.7	25.7 (2.5)	95.5	76.5 (0.5)	77.3 (0.5)
segment (7 cl.)	91.6	89.5	95.9	97.6	3.7	54.8	97.7	99.3	5.5	95.9	54.9 (3.0)	97.7	84.2 (0.9)	87.8 (1.0)
sonar (2 cl.)	83.8	91.8	87.1	100.0	2.0	56.5	94.4	100.0	2.0	87.1	46 (17)	94.4	69.9 (3.2)	75.2 (3.2)
spambase (2 cl.)	88.9	99.5	89.2	100.0	2.0	76.1	94.1	100.0	2.0	89.2	39 (12.8)	94.1	72.6 (0.8)	73.3 (0.8)
spect (2 cl.)	73.4	88.5	78.1	100.0	2.0	67.4	83.5	100.0	2.0	78.1	36.5 (8.8)	83.5	52.3 (4.0)	60.9 (4.7)
splice (3 cl.)	95.1	97.6	96.8	96.3	2.2	0.1	100.0	100.0	2.8	96.2	52.1 (5.5)	100.0	95.1 (0.2)	95.0 (0.2)
waveform (3 cl.)	81.4	99.1	81.6	99.8	2.0	54.1	89.6	100.0	2.1	81.6	49.2 (5.5)	89.6	71.7 (0.6)	72.0 (0.6)
yeast (10 cl.)	57.2	91.5	59.1	72.1	2.3	55.1	67.3	87.5	2.9	59.0	36.7 (6.7)	67.3	44.8 (1.8)	45.1 (1.9)

Table 7: Results data set by data set under the MAR-MP setting. For each data set, the number of classes is reported aside the name. For NccMarI, NccI, ΔNCC2-nonMAR, the standard deviation is reported into brackets; for the remaining indicators, the standard deviation is smaller than 2 percentage points in the large majority of cases.

References

U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, San Francisco, CA, 1993. Morgan Kaufmann.

D. F. Heitjan and D. B. Rubin. Ignorability and coarse data. *The Annals of Statistics*, 19(4):2244–2253, 1991.

M. Jaeger. Ignorability in statistical and probabilistic inference. *Journal of Artificial Intelligence Research*, 24:889–917, 2005.

S. Kotz, N. Balakrishnan, and N. L. Johnson. *Continuous Multivariate Distributions, Volume 1: Models and Applications*. Series in Probability and Statistics. Wiley, New York, 2000.

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

C. F. Manski. *Partial Identification of Probability Distributions*. Springer-Verlag, New York, 2003.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1993. 2nd edition.

M. Ramoni and P. Sebastiani. Robust Bayes classifiers. *Artificial Intelligence*, 125(1–2):209–226, 2001.

D. B. Rubin. Inference and missing data. *Biometrika,* 63(3):581–592, 1976.

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

P. Walley. Inferences from multinomial data: learning about a bag of marbles. *J. R. Statist. Soc. B*, 58(1):3–57, 1996.

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, 2005.

M. Zaffalon. Statistical inference of the naive credal classifier. In G. de Cooman, T. L. Fine, and T. Seidenfeld, editors, *ISIPTA '01: Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications*, pages 384–393, The Netherlands, 2001. Shaker.

M. Zaffalon. The naive credal classifier. *Journal of Statistical Planning and Inference,* 105(1):5–21, 2002.

M. Zaffalon. Credible classification for environmental problems. *Environmental Modelling and Software*, 20(8):1003–1012, 2005a.

M. Zaffalon. Conservative rules for predictive inference with incomplete data. In F. G. Cozman, R. Nau, and T. Seidenfeld, editors, *ISIPTA '05: Proceedings of the Fourth International Symposium on Imprecise Probabilities and Their Applications*, pages 406–415, Manno, Switzerland, 2005b. SIPTA.

M. Zaffalon, K. Wesnes, and O. Petrini. Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data. *Artificial Intelligence in Medicine*, 29(1–2):61–79, 2003.