The Multilabel Naive Credal Classifier☆

Alessandro Antonucci[a], Giorgio Corani[a]

[a]*Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno (Lugano), Switzerland.*

Abstract

A credal classifier for multilabel data is presented. This is obtained as an extension of Zaffalon's naive credal classifier to the case of non-exclusive class labels. The dependence relations among the labels are shaped with a tree topology. The classifier, based on a polynomial-time algorithm to compute whether or not a class label is optimal, returns a compact description of the set of optimal sequences of labels. Extensive experiments on real multilabel data show that the classifier gives more robust predictions than its Bayesian counterpart. In practice, when multiple sequences are returned in output, the Bayesian model is more likely to be inaccurate, while the sequences returned by the credal classifier are more likely to include the correct one.

Keywords: Credal classification, imprecise Dirichlet model, naive credal classifier, multilabel classification.

1. Introduction

A classifier represents the relationship between the characteristics of an object (*features*) and its category (*class*). A traditional *classifier* predicts the *class* variable given the value of the features. *Credal classifiers* generalise traditional classifiers, allowing for set-valued predictions, possibly including more than a single class. A credal classifier drops the non-optimal classes returning the classes that are potentially optimal given the information available. Depending on the data, there can be one or multiple optimal classes. Credal classifiers are thus less informative but more reliable than traditional classifiers [17]. Both credal and traditional classifiers assume the classes to be mutually *exclusive*.

Multilabel classification is a modern type of classification, in which an object is allowed to have multiple *relevant* classes (or *labels*). Multilabel classification naturally appears in many domains. E.g., a news article discussing international treaties could be labeled as politics *and* finance *and* environment, this

☆This paper is an extended and revised version of material originally presented in [1].

*Corresponding author.

Email addresses: alessandro@idsia.ch (Alessandro Antonucci), giorgio@idsia.ch (Giorgio Corani)

making its categorization a multilabel task. More generally speaking, lots of information retrieval tasks such as tagging of photos or videos or texts can be regarded as multilabel problems. In bioinformatics, the identification of the best mix of drugs for curing HIV has been addressed as a multilabel problem [32]. Similar considerations have been done, among many other examples, for protein classification [26].

Binary relevance is the simplest way to approach multilabel classification. Given q labels, binary relevance learns q independent single-label classifiers. The main shortcoming of binary relevance is that it ignores the dependencies among the different classes, which in many cases are important [24]. Different approaches have been proposed for modelling the dependencies among classes; see for instance [6, 31, 38].

Probabilistic graphical models allow for a direct modelling of the dependencies between class labels [2, 4, 7, 45]. Each label is represented by a Boolean variable. The i-th Boolean variable represents whether the i-th label is relevant or not for the current instance. The inference task is to detect the most probable joint configuration of the labels. A joint configuration of the labels is a *sequence* of zeros and ones. Given q labels, there are 2^q possible sequences.

Since the number of possible answers increases exponentially with the number of labels, some research has been devoted to compute robust multilabel classifications. In the related field of label ranking [29], some algorithms have been proposed for estimating robust partial (instead of complete) orders between labels [11]. A multilabel classifier which computes an interval of grades for the relevance of each label has been proposed by [35]. In [25] a credal approach is adopted to compute an outer approximation of the marginal posterior probability of each label.

We focus on probabilistic graphical models and we study the sensitivity of the multilabel prediction on parameter perturbations. We propose a graphical model which generalises the naive Bayes to the multilabel setting. We represent our uncertainty about the model parameters as a convex set of distributions, using the *imprecise Dirichlet model* (IDM) [5, 47]. To take decisions with models of this kind, we take decisions based on the optimality criterion of *maximality* [42]. We propose a polynomial-time algorithm which detects whether there is an optimal sequence with a class label in a given state. By iterating the procedure for each class and state we obtain a compact description of (an outer approximation of) the set of optimal sequences.

The paper is organised as follows. We review some basics about Bayesian networks and the IDM in Section 2. We indeed show how the IDM applies to Bayesian networks quantification in Section 3. The (single-label) classical naive credal classifier is reviewed in Section 4. The new model we present for multilabel data is described in Section 5. A discussion about how to perform classification with this model together with the technical theorems behind the inference algorithms is in Section 6. A demonstrative example is discussed in Section 7. The results of extensive numerical experiments and the conclusions are reported in Sections 8 and 9, while the proofs of the theorems are in the appendix.

2. Preliminaries

We denote random variables by uppercase letters, generic values by lowercase letters, and the sets of possible values, always assumed to be finite, by calligraphic letters. E.g., X is a variable whose generic value is $x \in \mathcal{X}$. For a Boolean variable X, the negation of each $x \in \mathcal{X} := \{0, 1\}$ is denoted as $\neg x$.

We denote by $P(X)$ a probability mass function over X. Given a set of variables $\boldsymbol{X} := (X_1, \ldots, X_n)$, arranged into a directed acyclic graph, a *Bayesian network* is a set of conditional probability tables $\{P(X_i|\mathrm{Pa}(X_i))\}_{i=1}^n$ where $\mathrm{Pa}(X_i)$ are the parents of X_i, i.e., the immediate predecessors of X_i within the graph. If the graph depicts conditional independence relations according to the Markov condition for directed graphs, the joint mass function $P(\boldsymbol{X})$ factorises as $P(\boldsymbol{x}) = \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))$, where the values x_i and $\mathrm{pa}(X_i)$ of X_i and $\mathrm{Pa}(X_i)$ are those consistent with $\boldsymbol{x}$ [34].

A *credal set* over X is a (convex) set of probability mass functions over X. Given a credal set, the *maximality* criterion allows to choose the optimal (i.e., most probable) states as follows: $x'' \in \mathcal{X}$ is *maximal* if and only if there is no $x' \in \mathcal{X}$ s.t. $P(x') > P(x'')$ for each $P(X)$ in the credal set [46].

The *imprecise Dirichlet model* (IDM) is a standard approach to learn credal sets from multinomial data [47]. Given a variable X, a Dirichlet prior distribution $P(\theta_x) \propto \theta^{st(x)-1}$ induces a probability $\theta_x = \frac{n(x)+st(x)}{N+s}$ for $X = x$, where $n(x)$ is the number of observations such that $X = x$ and N the total one.[1] The IDM approach consists in considering all the Dirichlet prior distributions such that $\sum_{x\in\mathcal{X}} t(x) = 1$, thus allowing θ_x to vary between $\frac{n(x)}{N+s}$ and $\frac{n(x)+s}{N+s}$ for each $x \in \mathcal{X}$.

In the next section we determine the form of the IDM constraints in the multivariate case when the relations among the different variables are described by a Bayesian network.

3. IDM-based Learning with Independence

In this section we discuss the particular problem of learning a set of multivariate distributions through the IDM under specific independence assumptions. This is done in the case where the independence relations are described within the framework of Bayesian networks. We extend Zaffalon's ideas stated in [50] for the naive Bayes case to general topologies. Let us begin with an example.

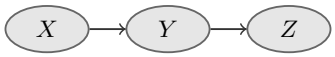

Figure 1: A chain topology

[1] We adopt Walley's parametrisation of the Dirichlet prior distribution, which highlights the role of the equivalent sample size s (see Section 3).

Example 1. *Consider a Bayesian network over three Boolean variables X, Y, and Z with the topology in Figure 1. According to the Markov condition for directed graphs [34], this models the conditional independence between X and Z given Y, corresponding to the following factorization of the joint probability mass function: $P(x,y,z) = P(x) \cdot P(y|x) \cdot P(z|y)$. Accordingly, the likelihood of a data set $\mathcal{D}$ of joint observations for the three variables is:*

$$L(\boldsymbol{\theta}) := P(\mathcal{D}|\boldsymbol{\theta}) = \prod_x \left(\theta_x^{n(x)} \left(\prod_y \theta_{y|x}^{n(x,y)} \left(\prod_z \theta_{z|y}^{n(y,z)} \right) \right) \right) , \tag{1}$$

where $\theta_x := P(x)$, $\theta_{y|x} := P(y|x)$, and $\theta_{z|y} := P(z|y)$, for each x, y, z, and $n(\cdot)$ is the counting function. A conjugate prior over the parameters $\boldsymbol{\theta}$ is:

$$P(\boldsymbol{\theta}) \propto \prod_x \left(\theta_x^{st(x)-1} \left(\prod_y \theta_{y|x}^{st(x,y)-1} \left(\prod_z \theta_{z|y}^{st(y,z)-1} \right) \right) \right) , \tag{2}$$

where s and the $t(\cdot)$ are nonnegative parameters. The first term in Eq. (2) is proportional to a Dirichlet prior distribution. We set $\sum_x t(x) = 1$. Considering the corresponding (structural) constraint for the counts in the likelihood, i.e., $\sum_x n(x) = N$, we can regard s as the equivalent sample size *(ESS) of this prior distribution.*

Let us identify the additional constraints required to regard s as an ESS even for the whole distribution in Eq. (2). The (again, structural) constraints on the likelihood $\sum_{xy} n(x,y) = \sum_{yz} n(y,z) = N$ correspond to:

$$\sum_{x,y} t(x,y) = \sum_{y,z} t(y,z) = 1 .$$

The posterior values of the parameters become therefore:

$$\begin{aligned} \theta_x &= \frac{n(x) + st(x)}{N + s} , \\ \theta_{y|x} &= \frac{n(x,y) + st(x,y)}{n(x) + st(x)} , \\ \theta_{z|y} &= \frac{n(y,z) + st(y,z)}{n(y) + st(y)} , \end{aligned}$$

with $t(x) = \sum_y t(x,y)$ and $t(y) := \sum_z t(y,z)$. An IDM-based model is therefore obtained by considering all *the specifications of the parameters consistent with the following constraints over $t(x)$, $t(x,y)$, and $t(y,z)$:*

$$\begin{aligned} \sum_x t(x) &= 1 , \\ \sum_y t(x,y) &= t(x), \forall x , \\ \sum_z t(y,z) &= \sum_x t(x,y), \forall y . \end{aligned}$$

Such a model can be regarded as induced by a set of prior distributions made of Dirichlet components and with ESS s. This is the way we generalise the IDM to multivariate models with independence. To check that the constraints are sufficient, consider all the (structural and not all independent) constraints satisfied by the count function $n(\cdot)$ in Eq. (1), i.e., $\sum_x n(x) = \sum_{xy} n(x,y) = \sum_{yz} n(y,z) = N$, $\sum_y n(x,y) = n(x)$, $\sum_z n(y,z) = n(y)$, $\sum_x n(x,y) = n(y)$. It is a trivial exercise to check that the $t(\cdot)$ parameters satisfy the analogous relations (with one in the place of N).

The above example deals with a node which is a child of a child of another variable. This pattern does not appear in Zaffalon's original work for the naive topology, neither in other papers about more connected topologies [52]. The procedure for Bayesian networks with generic topologies is just a straightforward extension of that outlined in the above example. The specifications over X apply to the root (i.e., parentless) nodes with Y replaced by the whole children set, the specifications over Z apply to any leaf (i.e., childless) node with Y replaced by the whole parent set, and those for Y apply to any non-root non-leaf node with the parents and children playing the role of X and Z. This discussion should be regarded as a guideline for the learning of the parameters of a Bayesian networks based on the IDM. The resulting model is a *credal network* [18], with the local parameters taking their values from different credal sets, but with the constraints over the parameters of the prior distribution inducing a *non-separate* specification of the local credal sets [3].

In the literature this approach is also known as the *global* IDM; the credal sets of the different random variables are linked by structural constraints. A different approach is constituted by the *local* IDM, in which an independent credal set is established for each random variable. The credal sets in this case are separately specified. See [28] for a detailed discussion of the differences between extensively and separately specified models. Separately specified credal sets allow to compute more easily the inferences; yet such inferences might be unnecessarily imprecise, due to the lack of constraints between the different variables [16].

4. The Naive Credal Classifier

In this section we briefly review the credal version of the naive Bayes classifier as proposed by Zaffalon in [50]. We denote the class variable as C and the feature variables as $\boldsymbol{F} := (F_1, \ldots, F_m)$. A data set of N complete i.i.d. joint observations of $(C, \boldsymbol{F})$ is available together with a counting function $n(\cdot)$. The features are assumed to be conditionally independent given the class. This corresponds to the topology in Figure 2, that induces the factorization $P(c, \boldsymbol{f}) = P(c) \cdot \prod_{i=1}^m P(f_i|c)$, for each $c \in \mathcal{C}$ and $\boldsymbol{f} := (f_1, \ldots, f_m) \in \prod_{i=1}^m \mathcal{F}_i$.

Following the same procedure as in Example 1, given a particular specification of the prior distribution (i.e., of $t(\cdot)$ and s), the parameters to be quantified

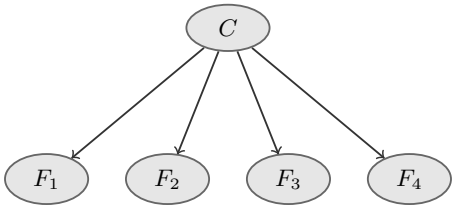

Figure 2: A naive topology

are:

$$P(c) = \frac{n(c) + st(c)}{N + s}\,,$$
$$P(f_i|c) = \frac{n(c, f_i) + st(c, f_i)}{n(c) + st(c)}\,,$$

for each $f_i \in \mathcal{F}_i$, $c \in \mathcal{C}$, $i = 1, \ldots, m$. The class labels assigned to an unannotated instance $\boldsymbol{f}$ of the features are those corresponding to $\arg\max_{c \in \mathcal{C}} P(c, \boldsymbol{f})$. The IDM constraints on the above positive[2] parameters are: $\sum_c t(c) = 1$ and $\sum_{f_i} t(c, f_i) = t(c)$, for each $i = 1, \ldots, m$ and $c \in \mathcal{C}$.[3] We denote as $\boldsymbol{t}$ a generic value for the joint variable of these parameters associated to the prior distribution, and by $\boldsymbol{\mathcal{T}}$ the corresponding feasible region.

The class labels assigned to $\boldsymbol{f}$ by this credal classifier are the *undominated* ones according to the maximality criterion (see Section 2). Given $c', c'' \in \mathcal{C}$, c' dominates c'' if $P(c', \boldsymbol{f}) > P(c'', \boldsymbol{f})$ for any specification consistent with the IDM constraints. This is equivalent to the following condition:

$$\inf_{\boldsymbol{t} \in \boldsymbol{\mathcal{T}}} \left[\frac{n(c'') + st(c'')}{n(c') + st(c')}\right]^{m-1} \prod_{i=1}^{m} \frac{n(c', f_i) + st(c', f_i)}{n(c'', f_i) + st(c'', f_i)} > 1\,, \tag{3}$$

where the exponent is the result of the simplification between the m contributions of the features and that of the class. The optimisation of the second term can be achieved independently by setting $t(c', f_i) = 0$ and $t(c'', f_i) = t(c'')$ for each $i = 1, \ldots, m$. The objective function rewrites therefore as:

$$\left[\frac{n(c'') + st(c'')}{n(c') + st(c')}\right]^{m-1} \prod_{i=1}^{m} \frac{n(c', f_i)}{n(c'', f_i) + st(c'')}\,,$$

with the remaining constraints being only $t(c') + t(c'') = 1$, with $t(c'), t(c'') > 0$ (set to zero the other variables). In other words, we can express the objective function as a function of a single variable. Its logarithmic derivative is a linear fractional variable, and the second derivative is always positive. Overall, the minimization can be efficiently achieved by bracketing (see [50]).

[2]Strict positivity is required, otherwise the corresponding prior would be improper.

[3]Here and in the following, if there is no risk of ambiguity, the arguments of the sums and the products are omitted for sake of notation. E.g., $\sum_c$ is a shortcut for $\sum_{c \in \mathcal{C}}$.

The *naive credal classifier* (NCC) repeats the test of dominance for each pair of labels. In this way it detects the *undominated* classes. For a given instance there can be a single undominated class or a set of undominated classes. When NCC returns a set of labels it shows an epistemic lack of information which prevents to identify with certainty the most probable class. The NCC is not a multilabel classifier: it assumes the class labels to be mutually exclusive. In this paper we extend the NCC to the multilabel framework by modifying its topology to capture the dependencies among the different labels.

5. The Multilabel Naive Credal Classifier (MNCC)

In order to extend the credal approach to classification, described in the previous section, to the multilabel case, let us define some more notations and introduce a toy example to be used in Section 7 to clarify the procedures we develop. Let q be the cardinality of the class C. In the multilabel framework, C is replaced by q (Boolean) class labels $\boldsymbol{C} := (C_1, \ldots, C_q)$. This is a standard way to cope with non-exclusivity: if the j-th label in $\mathcal{C}$ is relevant then $C_j = 1$, otherwise $C_j = 0$. The example here below has three non-exclusive labels.

Example 2. *We want to predict the national languages mastered by a Swiss citizen. The options are German (C_1), French (C_2), and Italian (C_3), we neglect Romansh, the fourth Swiss national language spoken by less than 1% of the population. These are non-exclusive options with eight (or seven) possible joint states.*[4] *To predict $\boldsymbol{C}$ we collect joint data about this variable and about the official language in the canton where the citizen and the parents of the citizen live. These data are reported in Table 1.*

The Swiss citizen speaks			F_1	F_2	F_3
German	French	Italian	Father	Mother	Citizen
C_1	C_2	C_3	lives in a canton		
$\surd$	$-$	$-$	German	German	Italian
$-$	$\surd$	$\surd$	Italian	Italian	French
$\surd$	$-$	$\surd$	Italian	French	German
$\surd$	$\surd$	$-$	German	German	Italian
$-$	$\surd$	$-$	French	Italian	French
$\surd$	$-$	$\surd$	German	German	Italian

Table 1: Data about the national languages spoken by six Swiss citizens and their relations with the official languages in the cantons where the parents and the citizen live

Without lack of generality let us call the first label C_1 the *superclass*, and the other class labels *subclasses*. We assume the conditional independence of

[4] As already emphasised in [14], in multilabel tasks, mutual exclusivity is relaxed, but not exhaustivity. This implies that the joint state with all the Boolean variables in their false state, which corresponds to the empty set, is usually not regarded as a possible outcome.

the subclasses given the superclass. Simplistically we set as superclass the class which is more frequently observed as relevant [14]. E.g., for the data in Table 1, German is properly indexed as C_1 as it is the most observed class label.

A data set of N joint observations of $(\boldsymbol{C}, \boldsymbol{F})$ is available together with a counting function $n(\cdot)$. Each feature is *replicated* q times. For each $k = 1, \ldots, m$, $\{F_k^j\}_{j=1}^q$ are replicas of F_k. For each $j = 1, \ldots, q$, the replicated features $\{F_k^j\}_{k=1}^m$ are assumed to be independent given C_j. This is a simplifying assumption, already formulated in other papers [4].[5] Accordingly, the joint mass function over the class labels and the features factorises as follows:

$$P(\boldsymbol{c}, \boldsymbol{f}) = P(c_1) \left[\prod_{i=2}^{q} P(c_i|c_1) \right] \prod_{j=1}^{q} \prod_{k=1}^{m} P(f_k^j|c_j)\,, \tag{4}$$

where the values of the class labels and of the features are those consistent with $\boldsymbol{c}$ and $\boldsymbol{f}$. Parameters in Eq. (4) can be learned from the data through a procedure similar to that in the previous sections, i.e.,

$$\begin{aligned} P(c_1) &= \frac{n(c_1) + st(c_1)}{N + s}\,, \\ P(c_i|c_1) &= \frac{n(c_1, c_i) + st(c_1, c_i)}{n(c_1) + st(c_1)}\,, \\ P(f_k^j|c_j) &= \frac{n(c_j, f_k) + st(c_j, f_k^j)}{n(c_j) + st(c_j)}\,. \end{aligned}$$

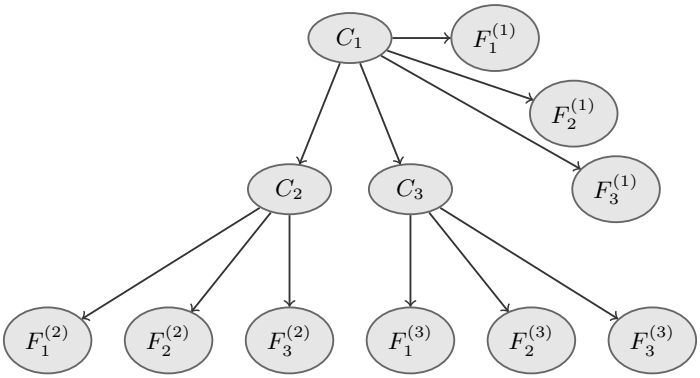

Figure 3: The multilabel naive topology for the variables in Example 2

The above considered model is a Bayesian network with a singly connected topology which we call *multilabel naive* topology (e.g., see Figure 3). A multilabel classifier based in this model can be implemented by assigning to a test

[5]Strictly speaking, an additional dummy child modelling the fact that all the replicas corresponds to the same variable should have been added.

instance of the features (replicated for each class label), the most probable configuration of the class labels, i.e.,

$$\boldsymbol{c}^* := \arg\max_{\boldsymbol{c}\in\{0,1\}^q} P(\boldsymbol{c},\boldsymbol{f})\,. \tag{5}$$

This is a MAP (maximum a posteriori) inference task on a singly connected Bayesian network, which can be efficiently solved by standard algorithms.[6]

Following the guidelines in Section 3, an IDM quantification of the above model is obtained by considering all the quantifications consistent with the following constraints:

$$\begin{aligned}\sum_{c_1} t(c_1) &= 1\,,\\ \sum_{c_i} t(c_1,c_i) &= t(c_1),\forall i\\ \sum_{f_k^j} t(c_j,f_k^j) = \sum_{c_1} t(c_1,c_j) &= t(c_j),\forall j,c_j\,,\end{aligned}$$

together with the strict positivity of all the parameters. Even in this case we denote by $\boldsymbol{t}$ the generic value of the joint variable including all these parameters and by $\boldsymbol{\mathcal{T}}$ the corresponding feasible region. The imprecision in this model can be regarded as induced by s missing observations, which we are completely ignorant about.

In the next section we show how the decision task in Eq. (5) can be extend to the IDM-based quantification considered here by mean of the maximality criterion defined in Section 2.

6. Inference with the MNCC

6.1. Maximal Sequences and Maximal Labels

Consider a complete observation $\boldsymbol{f}$ of the features and two sequences of labels $\boldsymbol{c}'$ and $\boldsymbol{c}''$. According to the maximality criterion, the second sequence is undominated by the first if and only if there is (at least) a prior distribution, and hence a value of $\boldsymbol{t}$, consistent with the constraints such that the first sequence is less (or equally) probable than the second, i.e.,[7]

$$\inf_{\boldsymbol{t}\in\boldsymbol{\mathcal{T}}} \frac{P_{\boldsymbol{t}}(\boldsymbol{c}',\boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'',\boldsymbol{f})} \leq 1\,. \tag{6}$$

[6] Solving a MAP task as in Eq. (5) for this particular topology is trivial. It is sufficient to compute the most probable configuration of the subclasses if the superclass is relevant and if it is irrelevant. Then the resulting MAP configuration is the most probable among the two.

[7] This is an alternative, but equivalent, formulation with respect to that in Eq. (3).

In Section 6.2 we show how to ascertain the dominance test in Eq. (6) in linear time with respect to the number of class labels and features. To detect the optimal sequences, the test should be iterated over all the possible pairs of distinct sequences. This adds to the complexity a factor quadratic in the number of possible sequences, which is in turn exponential in the number q of class labels.

An alternative strategy might be to directly ascertain whether sequence $\boldsymbol{c}''$ is optimal. This corresponds to evaluate if the condition in Eq. (6) is satisfied for each possible specification of $\boldsymbol{c}'$, i.e.,

$$\max_{\boldsymbol{c}'} \inf_{\boldsymbol{t}} \frac{P_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'', \boldsymbol{f})} \leq 1\,. \tag{7}$$

Yet, a procedure to detect the maximality of a sequence as in Eq. (7) has to be iterated for each possible specification of $\boldsymbol{c}''$, i.e., 2^q times. Accordingly, an explicit identification of the optimal sequences based on the above procedure is feasible only if the number of classes is limited, for instance $q < 10$.

Thus, we devise a different approach to deal with data sets containing many labels. We analyse, for each label, whether there are maximal sequences in which the label is relevant and non-relevant. We accomplish this task through the algorithm shown in the next section. This approach is however less informative than detecting the maximal sequences. Consider having detected k labels whose maximal states are both relevant and non-relevant. The 2^k sequences obtained combining their states in all possible ways contain the maximal sequences and, in general, others non-maximal sequences. To decide which ones of the 2^k sequences are maximal we should apply the test in Eq. (7); yet this is feasible only if k is limited.

The above idea corresponds to iterate the test in Eq. (7) for all the sequences with a given label, say l, in a given state $\tilde{c}_l$. Thus, if this is the case, we have the equivalent condition:

$$\min_{\boldsymbol{c}'' : c''_l = \tilde{c}_l} \max_{\boldsymbol{c}'} \inf_{\boldsymbol{t}} \frac{P_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'', \boldsymbol{f})} \leq 1\,, \tag{8}$$

showing that there is at least an optimal sequence with C_l in the state $\tilde{c}_l$. Note also that, removing the constraint $c''_l = \tilde{c}_l$ from Eq. (8), we have a test for the existence of at least a maximal sequence, which is true by definition. Thus, if the inequality in Eq. (8) is not satisfied for $c''_l = 1$, then it should be satisfied for $c''_l = 0$, and vice versa.

The technical results allowing for an efficient implementation of the optimization tasks in Eq. (6) and Eq. (8) are provided in the next section. We call this approach, based on the joint model in Eq. (4) and the corresponding IDM constraints, *multilabel naive credal classifier* (MNCC). The characterization of the maximal sequences we consider uses ideas analogous to those proposed by De Bock and de Cooman for hidden Markov models [20].

6.2. Solving the Optimization

In this section we present the technical results behind our implementation of the MNCC. Let us start from the maximality-based dominance test among

two sequences, which can be performed as follows.

Theorem 1. *Given two sequences $\boldsymbol{c}'$ and $\boldsymbol{c}''$ and an instance of the features $\boldsymbol{f}$, the decision task in Eq.* (6) *is equivalent to:*

$$\prod_{i=2}^{q} \delta(c'_i, \neg c''_i) \frac{n(c'_1, c'_i) \cdot g_i(c'_i, c''_i, \boldsymbol{f})}{n(c''_1, c''_i) + s} \leq 1\,, \tag{9}$$

if $c'_1 = c''_1$, with δ denoting the Kronecker delta function, and to

$$\inf_{0<t_1<1} h(c'_1, c''_1, t_1, \boldsymbol{f}) \prod_{i=2}^{q} \frac{n(c'_1, c'_i)\tilde{g}_i(c'_i, c''_i, \boldsymbol{f})}{n(c''_1, c''_i) + st_1} \leq 1\,, \tag{10}$$

if $c'_1 = \neg c''_1$, with

$$g_i(c'_i, c''_i, \boldsymbol{f}) := \inf_{0<t_i<1} \prod_{k=1}^{m} \frac{\frac{n(c'_i, f_k)}{n(c'_i)+s(1-t_i)}}{\frac{n(c''_i, f_k)+st_i}{n(c''_i)+st_i}}\,, \tag{11}$$

$\tilde{g}_i(c'_i, c''_i, \boldsymbol{f}) := g_i(c'_i, c''_i, \boldsymbol{f})$ if $c'_i = \neg c''_i$ and one otherwise, and

$$h(c'_1, c''_1, t_1, \boldsymbol{f}) := \left[\frac{n(c''_1) + st_1}{n(c'_1) + s(1-t_1)}\right]^{q+m-2} \prod_{k=1}^{m} \frac{n(c'_1, f_k)}{n(c''_1, f_k) + st_1}\,.$$

Moreover, the objective functions in the left-hand side of Eq. (10) *and the right-hand side of Eq.* (11) *are convex with respect to, respectively, t_1 and t_i.*

The proof of this theorem is in the appendix. Theorem 1 can be used to decide whether or not $\boldsymbol{c}'$ undominate $\boldsymbol{c}''$. Because of the convexity results (see the proof), the optima in Eq. (10) and Eq. (11) can be evaluated by bracketing (e.g., bisection) in a constant number of evaluations of the objective functions (assuming that we work with finite precision). This, in turn, takes only linear time in the number of labels and features. Overall, the dominance test only takes $O(qm)$ time. Moreover, to avoid numerical issues when dealing with many features and/or labels, the implementation of the test uses the logarithms of the above functions. The same can be done also for the task considered here below.

The optimization task in Eq. (8) is more involved. Given an observation of the features, a class label and a state of this label, Algorithm 1 provides a computation scheme which takes $O(qm)$ time exactly as that in Theorem 1. Note that the g_i functions should be computed as in Theorem 1, and the arg operator in lines 11 and 21 is intended to return the values of both the optimization variables. The following result gives a justification for this algorithm by proving that its output returns an upper bound for the left-hand side of Eq. (8).

Theorem 2. *Given an observation of the features $\boldsymbol{f}$, a label index l and a state $\tilde{c}_l$ of C_l, let $\gamma_l(\tilde{c}_l, \boldsymbol{f})$ be the output of Algorithm 1 with these inputs. Then:*

$$\min_{\boldsymbol{c}'':c''_l=\tilde{c}_l} \max_{\boldsymbol{c}'} \inf_{\boldsymbol{t}} \frac{P_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'', \boldsymbol{f})} \leq \gamma_l(\tilde{c}_l, \boldsymbol{f})\,.$$

The proof of the theorem is in the appendix. As a simple corollary of Theorem 2, $\gamma_l(\tilde{c}_l, \boldsymbol{f}) \leq 1$ is a sufficient condition for the inequality in Eq. (8) being satisfied. Accordingly, we can characterise the optimal sequences by iterating Algorithm 1 for both the states of each label. If the algorithm returns a number smaller than, or equal to, one, we conclude that there is at least an optimal sequence with the label in that state. If this is not the case for both states, it means that the outer bound in Theorem 2 is too loose and, in our framework, it is not possible to decide whether or not all the optimal sequences have that label in the same state. If this is the case, we cautiously assume that both states are possible. The overall procedure, demonstrated by Algorithm 2, gives therefore an outer approximation for the set of maximal sequences and its complexity is affected by an additional factor linear in the number of labels, thus resulting $O(q^2m)$. A binary relevance approach achieved by applying the NCC separately for each label would take only $O(qm)$. MNCC is therefore q times slower, this being a potential issue when coping with a huge number of labels.

7. A Demonstrative Example

To illustrate how the MNCC works in practice, let us consider the multilabel data in Table 1. We use the MNCC to decide whether a person with both parents living in a German-speaking canton and living in an Italian-speaking canton speaks German. For sake of brevity, we model this joint observation of the three features as a state f of a single feature F. The marginal counts about the citizens speaking German, French and Italian are $n(C_1 = 1) = 4$, $n(C_2 = 1) = 3$, and $n(C_3 = 1) = 3$, while the joint counts about French and Italian for the considered state of the feature are $n(C_2 = 1, f) = 1$ and $n(C_3 = 1, f) = 1$. The counts for the labels being non-relevant are obtained by complement considering that $N = 6$ and $n(f) = 3$.

Let us preliminary compute the functions defined in Eq. (11) with $s = 1$. For Italian, i.e., the third label, we have:

$$g_3(0,1,f) := \inf_{0<t<1} \frac{\frac{n(C_3=0,f)}{n(C_3=0)+1-t}}{\frac{n(C_3=1,f)+t}{n(C_3=1)+t}} = \inf_{0<t<1} \frac{\frac{2}{4-t}}{\frac{1+t}{3+t}} \simeq 1.319\,.$$

We similarly obtain $g_3(1,0,f) \simeq .375$ and the same values for g_2. To decide if the citizen speaks German, we consider a task as in Eq. (8), for which the execution of Algorithm 1 with $l = 1$ and $\tilde{c}_l = 1$ gives an upper bound because of Theorem 1. As German is the superclass, we consider the pseudocode in lines 19-25. For the term in line 19, we have that $\Phi(1,1)$ (and, because of line 24, even $\Phi(0,0)$) is obtained by multiplying:

$$\min\left\{\max\left\{1, \frac{n(C_1{=}1, C_2{=}1)g_2(1,0,f)}{n(C_1{=}1, C_2{=}0)+1}\right\}, \max\left\{\frac{n(C_1{=}1, C_2{=}0)g_2(0,1,f)}{n(C_1{=}1, C_2{=}1)+1}, 1\right\}\right\}$$

for the analogous term associated to C_3. The counts required to evaluate the first term are $n(C_1 = 1, C_2 = 1) = 1$, $n(C_1 = 1, C_2 = 0) = 3$, and the term

Algorithm 1 MNCC: outer bound computation as in Theorem 2

Input: Observation of the features $\boldsymbol{f}$, label index l, and state $\tilde{c}_l$ of C_l
Output: $\gamma_l(\tilde{c}_l, \boldsymbol{f})$

if $l > 1$ **then**
  for $c'_1 \leftarrow 0, 1$ **do**
    for $c''_1 \leftarrow 0, 1$ **do**
      if $c'_1 = c''_1$ **then**
        $\Phi(c'_1, c''_1) \leftarrow \prod_{i \neq l} \min\left\{\max\left\{1, \frac{n(c'_1, c'_i=1) g_i(1,0,\boldsymbol{f})}{n(c'_1, c''_i=0)+s}\right\}, \max\left\{\frac{n(c'_1, c'_i=0) g_i(0,1,\boldsymbol{f})}{n(c'_1, c''_i=1)+s}, 1\right\}\right\}$
        $\Phi(c'_1, c''_1) \leftarrow \Phi(c'_1, c''_1) \cdot \max\left\{1, \frac{n(c'_1, \neg\tilde{c}_l) \cdot g_l(\neg\tilde{c}_l, \tilde{c}_l, \boldsymbol{f})}{n(c'_1, \tilde{c}_l)+s}\right\}$
      else
        $(\tilde{c}'_l, \tilde{c}''_l) \leftarrow \left(\arg\max_{c'_l} n(c_1, c'_l) \cdot \tilde{g}_l(c'_l, \tilde{c}_l, \boldsymbol{f}), \tilde{c}_l\right)$
        for $i \leftarrow 2, q$ **do**
          if $i \neq l$ **then**
            $(\tilde{c}'_i, \tilde{c}''_i) \leftarrow \arg\min_{c''_i} \frac{\max_{c'_i} n(c'_1, c'_i) \tilde{g}_i(c''_i, c'_i, \boldsymbol{f})}{n(c''_1, c''_i)}$
          end if
        end for
        $\Phi(c'_1, c''_1, \boldsymbol{f}) := \inf_{\boldsymbol{t} \in \mathcal{T}} \frac{P_{\boldsymbol{t}}(c'_1, \tilde{c}'_2, \ldots, \tilde{c}'_q, \boldsymbol{f})}{P_{\boldsymbol{t}}(c''_1, \tilde{c}''_2, \tilde{c}''_q, \boldsymbol{f})}$ ▷ By Theorem 1
      end if
    end for
  end for
else
  $\Phi(\tilde{c}_l, \tilde{c}_l) \leftarrow \prod_{i>1} \min\left\{\max\left\{1, \frac{n(\tilde{c}_l, c'_i=1) g_i(1,0,\boldsymbol{f})}{n(\tilde{c}_l, c''_i=0)+s}\right\}, \max\left\{\frac{n(\tilde{c}_l, c'_i=0) g_i(0,1,\boldsymbol{f})}{n(\tilde{c}_l, c''_i=1)+s}, 1\right\}\right\}$
  for $i \leftarrow 2, q$ **do**
    $(\tilde{c}'_i, \tilde{c}''_i) \leftarrow \arg\min_{c''_i} \frac{\max_{c'_i} n(\neg\tilde{c}_l, c'_i) \tilde{g}_i(c''_i, c'_i, \boldsymbol{f})}{n(\tilde{c}_l, c''_i)}$
  end for
  $\Phi(\neg\tilde{c}_l, \tilde{c}_l, \boldsymbol{f}) := \inf_{\boldsymbol{t} \in \mathcal{T}} \frac{P_{\boldsymbol{t}}(\neg\tilde{c}_l, \tilde{c}'_2, \ldots, \tilde{c}'_q, \boldsymbol{f})}{P_{\boldsymbol{t}}(\tilde{c}_l, \tilde{c}''_2, \tilde{c}''_q, \boldsymbol{f})}$ ▷ By Theorem 1
  $\Phi(\neg\tilde{c}_l, \neg\tilde{c}_l) \leftarrow \Phi(\tilde{c}_l, \tilde{c}_l)$ ▷ Replicated value
  $\Phi(\tilde{c}_l, \neg\tilde{c}_l) \leftarrow \Phi(\neg\tilde{c}_l, \tilde{c}_l)$ ▷ Replicated value
end if
$\gamma_l(\tilde{c}_l, \boldsymbol{f}) \leftarrow \min_{c''_1} \max_{c'_1} \Phi(c'_1, c''_1, \boldsymbol{f})$
return $\gamma_l(\tilde{c}_l, \boldsymbol{f})$

Algorithm 2 MNCC: global procedure

Input: observation of the features $\boldsymbol{f}$

Output: Boolean function $\mu(l, c)$ true iff there is a maximal seq with $C_l = c$

1: **for** $l \leftarrow 1, q$ **do**
2: **for** $c \leftarrow 0, 1$ **do**
3: **if** $\gamma_l(c, \boldsymbol{f}) \leq 1$ **then** ▷ By Algorithm 1
4: $\mu(l, c) = 1$
5: **else**
6: $\mu(l, c) = 0$
7: **end if**
8: **end for**
9: **if** $\mu(l, 0) = 0$ **and** $\mu(l, 1) = 0$ **then**
10: $\mu(l, 0) \leftarrow 1$
11: $\mu(l, 1) \leftarrow 1$
12: **end if**
13: **end for**
14: **return** μ

give therefore $1 = \min\{\max\{1, \frac{.375}{4}\}, \max\{\frac{3 \cdot 1.319}{2}, 1\}\}$. For the second term we similarly obtain a contribution one. Accordingly $\Phi(1, 1) = \Phi(0, 0) = 1$. Afterwards, we discuss the computations in line 21. For $i = 2$ the objective function is:

$$\min \left\{ \frac{\max\{n(C_1 = 0, C_2 = 0), n(C_1 = 0, C_2 = 1) g_2(1, 0, f)\}}{n(C_1 = 1, C_2 = 0)}, \right.$$
$$\left. \frac{\max\{n(C_1 = 0, C_2 = 0) g_2(0, 1, f), n(C_1 = 0, C_2 = 1)\}}{n(C_1 = 1, C_2 = 1)} \right\}$$

As $n(C_1 = 0, C_2 = 0) = 0$, $n(C_1 = 0, C_2 = 1) = 2$, $n(C_1 = 1, C_2 = 0) = 3$, and $n(C_1 = 1, C_2 = 1) = 1$, both maxima are attained on the second term, while the minimum of the two is the first. Thus $\tilde{c}'_2 = 0$ and $\tilde{c}''_2 = 0$. We similarly identify $\tilde{c}'_3$ and $\tilde{c}''_3$. Then, we solve the dominance test in line 23 with the procedure described in Theorem 1. The resulting value is smaller than one. Finally, the evaluation in line 27 gives $\gamma_1(1) = 1$, which implies that there is at least a maximal sequence with the citizen speaking German.

By iterating the same calculations for all labels according to the scheme in Algorithm 2, we obtain that all the maximal sequences have the citizen speaking German and not speaking French, while a condition of indecision holds about Italian. In summary, the possible outputs for $\boldsymbol{C}$ consistent with those results are the sequences {German} and {German, Italian}. The person with both parents living in a German-speaking canton and living in an Italian-speaking canton can speak German only or both German and Italian.

1

8. Experiments

We compare MNCC algorithm against the three competitors (MNBC, NCC, and NBC) described in the following.

The Competitors. MNBC is the Bayesian counterpart of MNCC; thus it is a Bayesian network classifier with the same topology of MNCC (Figure 3), quantified using the BDeu prior [34, Chap.17]. We perform inference by solving the MAP task of Eq. (5).

NCC is binary relevance implemented on the basis of the naive credal classifier [17]. This means to train a binary NCC for each label and then merging the outputs of these credal binary classifiers. This yields a compact description of a set of optimal sequences analogous to that in Algorithm 2.

NBC is instead binary relevance based on the training of a naive Bayes separately for each label. This is the Bayesian counterpart of NCC.

The sets of optimal sequences detected by MNCC and NCC include those returned by respectively MNBC and NBC.

The Data Sets. We consider fifteen data sets, mostly related to information retrieval tasks. This includes sound (music for EMOTIONS and CAL500, animal calls for BIRDS), images (SCENE, FLAGS, and NUS-WIDE), text (e-mails for ENRON, restaurant reviews for YELP, radiology reports for MEDICAL, movie plot summaries for IMDB,[8] web posts for SLASHDOT), and video (MEDIAMILL). Bioinformatics data related to protein genomics are also considered (YEAST and GENBASE). The E-MOBILITY data set is taken from a mobility study. It tracks the means of transport (car, train, bus, etc.) used by a person during a trip on the basis of the length and duration of the trip, hour and day of the week, reason of the trip, and others. Table 2 reports the main characteristics of these data sets: the number of labels, the number of features before and after feature selection (see the paragraph below), the number of instances and the label density, i.e., the ratio of positive outcomes over the whole set of labels.

Preprocessing. We validate the classifiers by ten-fold cross validation. Before training, some preprocessing actions are performed. First, we discretise numerical features into four equally sized bins.

Secondly, we perform feature selection as follows. We adopt the correlation-based feature selection (CFS) [48, Chap. 7.1], often used in traditional classification. We perform CFS on the replicated features for each different label, and retain the union of all individually selected features. This is a useful preprocessing step which reduces the number of features, removing the non-relevant ones. Feature selection is helpful as our models have no links between features and would thus compute biased probabilities when dealing with correlated features. Feature selection for multilabel classification is however an open problem, and more sophisticated approaches can be designed.

[8] Courtesy of IMDb (`http://www.imdb.com`).

#	Data Set	Ref.	Classes	Density	Features	Instances
1	Emotions	[43]	6	0.311	72/44	593
2	Scene	[8]	6	0.179	294/224	2407
3	Flags	[30]	7	0.484	19/14	194
4	E-mobility	[10]	8	0.121	13/12	3218
5	Yelp	[39]	8	0.295	668/203	1951
6	Birds	[9]	11	0.116	260/90	435
7	Genbase	[26]	13	0.087	1185/68	662
8	Yeast	[27]	13	0.325	103/92	2417
9	Medical	[36]	14	0.084	1449/292	888
10	Slashdot	[38]	14	0.084	1079/462	3663
11	Nus-Wide	[12]	16	0.114	498/408	1981
12	Imdb		19	0.097	1001/590	8792
13	Enron	[33]	24	0.131	257/214	1696
14	Mediamill	[41]	25	0.157	120/91	4857
15	Cal500	[44]	119	0.201	68/67	502

Table 2: Characteristics of the data sets

Splitting multilabel data in folds well-stratified with respect to all the labels simultaneously is a problematic task for multilabel data [40]. Training sets with no positive occurrences of a particular label can therefore appear. As IDM-based credal classifiers become unnecessarily imprecise with zero counts [15], we eventually discard rare class labels (less than two percent of positive outcomes) from the benchmark. Instances with no active labels after the removal of the rare classes are consequently removed (see comment in Footnote 4). A specific treatment of multilabel data with rare labels in the credal case should be regarded as a necessary future work.

Indicators for Credal Classification. Various indicators can be considered to characterise the performance of a credal classifier. A classifier is determinate if a single output is returned and indeterminate otherwise. The *determinacy* is the percentage of instances on which the classifier is determinate. To summarise the trade-off between informativeness and accuracy, we use the utility measures proposed by Zaffalon et al. [51]. Such measures yield a unique performance descriptor to be compared to the accuracy of a traditional classifier. We consider the u_{65} and the u_{80} functions, which are quadratic transformations of a *discounted* utility paying $\frac{1}{k}$ if one of the k outputs returned by the credal classifier is correct, and zero otherwise. The transformations are such that, as in the discounted case, the utility of a wrong classification is zero and that of a correct and determinate classification is one. If the classifier gives two options, one of the two being the correct one, the transformations increase the discounted value

.50 to, respectively, .65 and .80.[9]

Joint versus Marginal. The accuracy of a multilabel classifier is usually measured in a *joint* way, i.e., the classifier is correct if the set of active labels in output exactly matches the correct one and wrong otherwise. In Table 3 we report the accuracy and the u_{80} measures for the traditional and the credal classifiers, measured in a joint way. We do not report the u_{65} for the sake of space. Yet, we do statistically analyse later even the results using u_{65}. Note that with many class labels, the value of the joint indicator becomes negligible and thus not meaningful. For this reason the three data sets with more than 20 labels are not reported in this table: each algorithm has accuracy (or u_{80} utility in the credal case) which is practically zero.

A different viewpoint can be obtained by looking at the *marginal* accuracies computed separately for each label and then averaging them. In Table 4 we report the marginal accuracy and the marginal u_{80} for all the data sets. This corresponds to the classical Hamming loss.

Further indicators. Future studies might inspect also further indicators of performance for multilabel classification. It is well known that each loss function requires a different inference in order to be optimised [22]. For instance another important indicator is the F-measure. Recently it has been proposed an inference which optimises the value of the F-measure based on the posterior probability of the labels being relevant [23]. This is a fairly complex algorithm and we leave its extension to the imprecise case for future work. Other commonly used indicators are the F-micro and F-macro; however there are currently no inferences able to maximise their values. They are thus out of our scope.

Rejecting Non-Maximal Sequences. The joint performance of the credal classifiers displayed in Table 3 refers to the output returned by the MNCC as in Algorithm 2, and similarly for the NCC. Yet, if the number k of labels for which an indeterminate output is returned is not huge, it is possible to enumerate all the 2^k consistent sequences and check the maximality of each sequence as in Eq. (7). We perform such a deeper analysis when $k \leq 10$. This is observed to give more informative results in the sense that the procedure reject some consistent sequences and, in our experiments, the actual sequence was never rejected. Yet, the effect on the u_{80} (and the same for the u_{60}) accuracy is not significant.

Statistical Analysis. We compare the different algorithm across multiple data sets using the signed-rank test (α=0.05). As for the joint accuracy, no significant difference can be detected between algorithms. This might also due to the reduced sample size, as three data sets are excluded from this analysis.

When we move to the marginal accuracy, we detect the following significant differences. MNCC is significantly more accurate than NBC, using both u_{65} and

[9] This corresponds to $u_{65}(x) := x(1.6 - 0.6x)$ and $u_{80}(x) := x(2.2 - 1.2x)$, where x is the discounted utility contribution.

#	Data Set	u_{80}				accuracy			
		MNCC (2.4)		NCC (2.7)		MNBC (2.1)		NBC (2.5)	
1	Emotions	26.57	(2)	26.81	(1)	26.10	(3)	25.42	(4)
2	Scene	30.48	(1)	28.32	(3)	29.08	(2)	27.92	(4)
3	Flags	16.31	(1)	16.19	(2)	13.68	(3)	12.63	(4)
4	E-mobility	43.72	(1)	30.11	(3)	43.71	(2)	29.66	(4)
5	Yelp	22.19	(1)	21.94	(2)	20.41	(3)	19.33	(4)
6	Birds	27.37	(3)	27.29	(4)	31.86	(1)	31.16	(2)
7	Genbase	94.64	(4)	94.67	(3)	98.64	(2)	98.65	(1)
8	Yeast	9.60	(4)	9.89	(2)	9.75	(3)	10.00	(1)
9	Medical	36.16	(4)	36.71	(3)	44.77	(2)	48.86	(1)
10	Slashdot	40.27	(4)	40.48	(3)	43.20	(1)	41.64	(2)
11	Nus-Wide	7.13	(3)	6.27	(4)	12.93	(1)	11.46	(2)
12	Imdb	8.93	(1)	7.26	(3)	8.73	(2)	6.76	(4)

Table 3: Joint performance with ranks (in parentheses) of the algorithms

#	Data Set	u_{80}				accuracy			
		MNCC (2)		NCC (1.8)		MNBC (3.3)		NBC (2.8)	
1	Emotions	77.82	(2)	78.03	(1)	76.95	(4)	77.09	(3)
2	Scene	83.86	(1)	82.92	(3)	83.35	(2)	82.74	(4)
3	Flags	77.98	(1)	77.32	(2)	74.06	(4)	74.29	(3)
4	E-mobility	87.03	(3)	88.42	(1)	86.82	(4)	88.32	(2)
5	Yelp	82.22	(2)	82.28	(1)	79.57	(4)	80.04	(3)
6	Birds	87.77	(2)	87.91	(1)	84.33	(4)	84.69	(3)
7	Genbase	99.29	(3)	99.28	(4)	99.90	(1)	99.89	(2)
8	Yeast	69.01	(2)	69.87	(1)	67.84	(4)	68.89	(3)
9	Medical	93.54	(3)	93.51	(4)	94.16	(2)	94.62	(1)
10	Slashdot	94.50	(2)	94.64	(1)	93.90	(4)	94.06	(3)
11	Nus-Wide	78.10	(1)	77.79	(2)	77.05	(3)	76.87	(4)
12	Imdb	88.76	(2)	89.18	(1)	87.63	(4)	88.47	(3)
13	Enron	81.06	(2)	81.09	(1)	77.69	(4)	77.84	(3)
14	Mediamill	78.10	(1)	77.79	(2)	77.05	(3)	76.87	(4)
15	Cal500	72.87	(1)	72.25	(2)	70.03	(4)	70.08	(3)

Table 4: Marginal performance with ranks (in the parentheses) of the algorithms

u_{80}. MNCC is significantly more accurate also than MNBC, but this holds only when the u_{80} is considered. Dealing with u_{65}, which values less favorably the indeterminate classifications, there is no significant difference between MNCC and MNBC. No significant difference is detected between MNCC and NCC; in this case they are both assessed using u_{65} and then they are both assessed using u_{80}.

Label-wise Analysis. Label-wise results for some data sets are shown in Figure 4. The histograms show, separately for each label, the MNBC accuracy on the instances on which the MNCC model is determinate (light bars) and indeterminate (dark bars). The black squares denote the determinacy level. One can notice that the accuracy of MNBC is lower on the instances imprecisely classified by MNCC. Moreover, the drop is usually stronger when the determinacy is high, i.e. when MNCC is indeterminate only on few difficult instances. This suggests a possible application of MNCC as a preprocessing tool to detect for each instance which are the hard-to-classify labels (e.g., to be annotated at a later time by human experts).

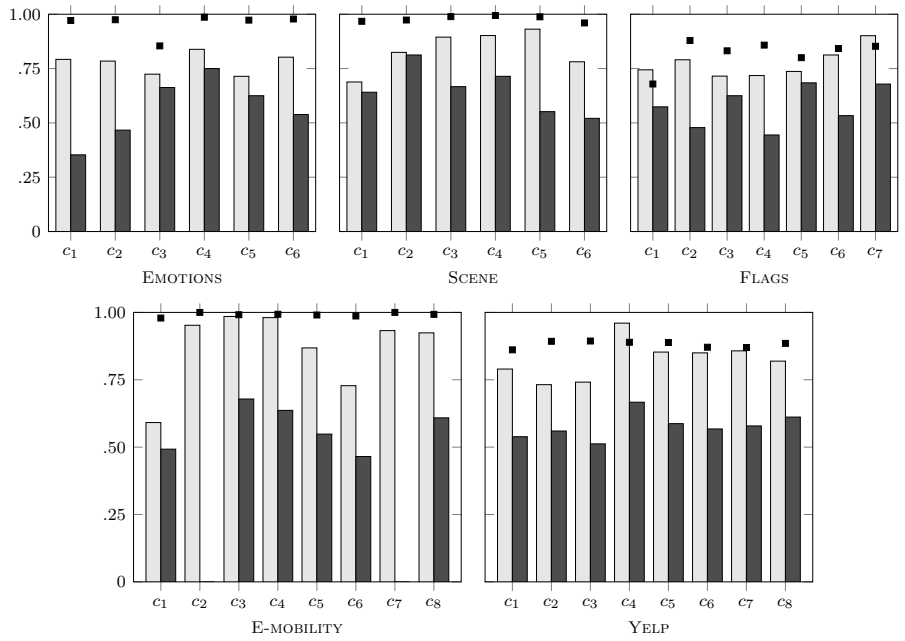

Figure 4: Label-wise results for five benchmark data sets

Joint Determinacy. The joint determinacy tends to decrease with the number of labels. See for instance Figure 5 where the results for some benchmark datasets are reported (numbers above the points are datasets identifier as in Table 2). This can be explained as follows. The number of possible sequences increases

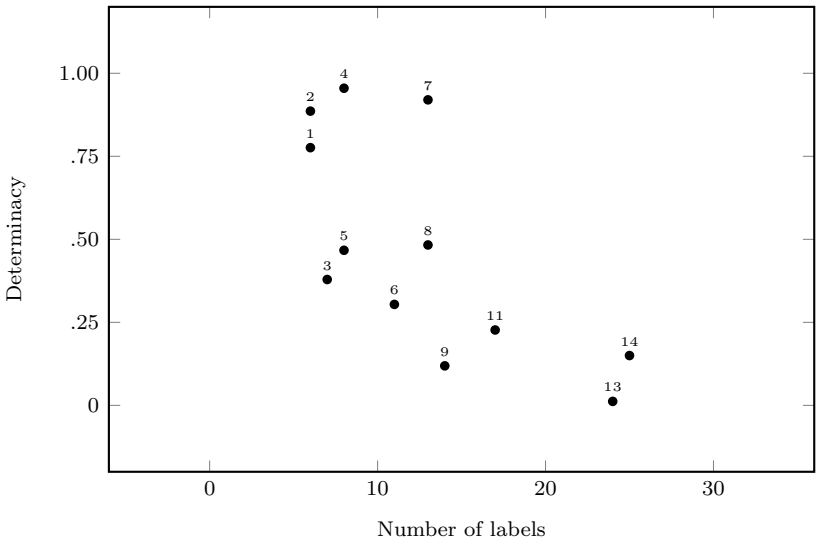

Figure 5: MNCC joint determinacy as a function of the number of labels

exponentially with the number of labels. As the number of labels increases it is thus easier to find at least one maximal sequence in which the label is relevant and another which in which it is irrelevant. With 10 labels, there are 1024 possible sequences. In 512 the label is relevant; in the other 512 it is irrelevant. To classify a label as indeterminate it is enough to find a maximal sequence among the 512 in which the label is relevant and another maximal sequence among the 512 in which the label is irrelevant.

Software. A Matlab software implementation of the MNCC, together with some Java preprocessing tools and the data sets used for the experiments, is freely available at `http://ipg.idsia.ch/software`.

9. Conclusions

We have generalised the naive credal classifier to cope with multilabel data. The corresponding polynomial-time algorithm, called MNCC, is able to decide for each Boolean state of each label, whether or not there is at least an optimal sequence with the label in that particular state. MNCC outperforms its Bayesian counterpart, thus the robustness we introduce in the prediction does not compromise the informativeness of the output.

There is still a lot of future work to be done. As we already mentioned, specific techniques for the treatment of rare class labels and feature selection should be developed. Apart from that, we believe that the connected topology we are currently using to describe the class-to-class relations, might penalise the performance of the classifier when coping with independence relations among

some of the labels. The recently proposed ETAN model [21], to be extended to the credal case, could be a better option. Alternatively, different criteria to identify the superclass as well as ensemble methods could be considered [13].

It might be also interesting to compare the inferences yielded by local and the global specification of the IDM (e.g., by exploiting some of the results in [19]), consider optimality criteria others than maximality (e.g., E-admissibility), and take decision on the basis of marginal (instead of joint) inferences. A comparison with other methods possibly yielding multiple sequences (e.g., [37, 49]) could be also considered.

Acknowledgments

We thank Claudio Bonesana for support during the preparation of the data sets. We thank Jasper De Bock and Cassio de Campos for stimulating discussions about MAP tasks in credal networks. We also thank Denis Mauá for his suggestions about possible justification of models with the replicated features.

References

[1] A. Antonucci and G. Corani. The multilabel naive credal classifier. In T. Augustin, S. Doria, E. Miranda, and E. Quaeghebeur, editors, *Proceedings of the Ninth International Symposium on Imprecise Probability: Theories and Applications*, pages 27–36. SIPTA, 2015.

[2] A. Antonucci, G. Corani, D.D. Mauá, and S. Gabaglio. An ensemble of Bayesian networks for multilabel classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 1220–1225. AAAI Press, 2013.

[3] A. Antonucci and M. Zaffalon. Decision-theoretic specification of credal networks: a unified language for uncertain modeling with sets of Bayesian networks. *International Journal of Approximate Reasoning*, 49(2):345–361, 2008.

[4] J. Arias, J. Gámez, T.D. Nielsen, and J.M. Puerta. A pairwise class interaction framework for multilabel classification. In L.C. van der Gaag and A. Feelders, editors, *Proceedings of the Seventh European Workshop on Probabilistic Graphical Models*, volume 8754 of *Lecture Notes in Artificial Intelligence*, pages 17–32. Springer, 2014.

[5] J.M. Bernard. An introduction to the imprecise Dirichlet model for multinomial data. *International Journal of Approximate Reasoning*, 39(2):123–150, 2005.

[6] W. Bi and J.T. Kwok. Multi-label classification on tree- and DAG-structured hierarchies. In L. Getoor and T. Scheffer, editors, *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, pages 17–24, 2011.

[7] C. Bielza, G. Li, and P. Larrañaga. Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, 2011.

[8] M.R. Boutell, J. Luo, X. Shen, and C.M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[9] F. Briggs, Y. Huang, R. Raich, K. Eftaxias, Z. Lei, W. Cukierski, S.F. Hadley, A. Hadley, M. Betts, X.Z. Fern, J. Irvine, L. Neal, A. Thomas, G. Fodor, G. Tsoumakas, H.W. Ng, T.N.T. Nguyen, H. Huttunen, P. Ruusuvuori, T. Manninen, A. Diment, T. Virtanen, J. Marzat, J. Defretin, D. Callender, C. Hurlburt, K. Larrey, and M. Milakov. The ninth annual MLSP competition: new methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *Proceedings of the 2013 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–8, 2013.

[10] F. Cellina, A. Förster, D. Rivola, L. Pampuri, R. Rudel, and A.E. Rizzoli. Using smartphones to profile mobility patterns in a living lab for the transition to E-mobility. In J. Hřebíček, G. Schimak, M. Kubásek, and A.E. Rizzoli, editors, *Environmental Software Systems. Fostering Information Sharing (Proceedings of the Tenth IFIP WG 5.11 International Symposium)*, volume 413 of *IFIP Advances in Information and Communication Technology*, pages 154–163. Springer, 2013.

[11] W. Cheng, M. Rademaker, B. De Baets, and E. Hüllermeier. Predicting partial orders: ranking with abstention. In J.L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases (Part I)*, pages 215–230. Springer, 2010.

[12] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: A real-world web image database from national university of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 48:1–48:9. ACM, 2009.

[13] G. Corani and A. Antonucci. Credal ensembles of classifiers. *Computational Statistics and Data Analysis*, 71:818–831, 2014.

[14] G. Corani, A. Antonucci, D.D. Mauá, and S. Gabaglio. Trading off speed and accuracy in multilabel classification. In L.C. van der Gaag and A. Feelders, editors, *Proceedings of the Seventh European Workshop on Probabilistic Graphical Models*, volume 8754 of *Lecture Notes in Artificial Intelligence*, pages 145–159. Springer, 2014.

[15] G. Corani and A. Benavoli. Restricting the IDM for classification. In E. Hüllermeier, R. Kruse, and F. Hoffmann, editors, *Information Processing and Management of Uncertainty in Knowledge-based Systems. Theory*

and Methods, volume 80 of *Communications in Computer and Information Science*, pages 328–337. Springer, 2010.

[16] G. Corani and C.P. de Campos. A tree augmented classifier based on extreme imprecise Dirichlet model. *International Journal of Approximate Reasoning*, 51(9):1053–1068, 2010.

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

[18] F.G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.

[19] J. De Bock, C.P. de Campos, and A. Antonucci. Global sensitivity analysis for MAP inference in graphical models. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and Weinberger K.Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2690–2698. Curran Associates, Inc., 2014.

[20] J. De Bock and G. de Cooman. An efficient algorithm for estimating state sequences in imprecise hidden Markov models. *Journal of Artificial Intelligence Research*, 50:189–233, 2014.

[21] C.P. de Campos, G. Corani, M. Scanagatta, M. Cuccu, and M. Zaffalon. Learning extended tree augmented naive structures. *International Journal of Approximate Reasoning*, 68:153–163, 2016.

[22] K.J. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. Regret analysis for performance metrics in multi-label classification: the case of Hamming and subset zero-one loss. In J.L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, volume 6321 of *Lecture Notes in Computer Science*, pages 280–295. Springer, 2010.

[23] K.J. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier. An exact algorithm for F-measure maximization. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1404–1412. Curran Associates, Inc., 2011.

[24] K.J. Dembczyński, W. Waegeman, and E. Hüllermeier. An analysis of chaining in multi-label classification. In L. De Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. Lucas, editors, *Frontiers in Artificial Intelligence and Applications (Proceedings of the Twentieth European Conference on Artificial Intelligence)*, volume 242, pages 294–299. IOS Press, 2012.

[25] S. Destercke. Multilabel predictions with sets of probabilities: the Hamming and ranking loss cases. *Pattern Recognition*, 48(11):3757–3765, 2015.

[26] S. Diplaris, G. Tsoumakas, P. Mitkas, and I. Vlahavas. Protein classification with multiple algorithms. In *Proceeding of the Tenth Panhellenic Conference on Informatics*, pages 448–456, 2005.

[27] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 681–687. MIT Press, 2002.

[28] J.C. Ferreira da Rocha and F.G. Cozman. Inference with separately specified sets of probabilities in credal networks. In A. Darwiche and N. Friedman, editors, *Proceedings of the Eighteenth conference on Uncertainty in Artificial Intelligence*, pages 430–437. Morgan Kaufmann, 2002.

[29] J. Fürnkranz, E. Hüllermeier, E.L. Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008.

[30] E.C. Gonçalves, A. Plastino, and A.A. Freitas. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Proceedings of the IEEE Twenty-Fifth International Conference on Tools with Artificial Intelligence*, pages 469–476. IEEE Computer Society, 2013.

[31] Y. Guo and S. Gu. Multi-label classification using conditional dependency networks. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 2, pages 1300–1305. AAAI Press, 2011.

[32] D. Heider, R. Senge, W. Cheng, and E. Hüllermeier. Multilabel classification for exploiting cross-resistance information in HIV-1 drug resistance prediction. *Bioinformatics*, 29(16):1946–1952, 2013.

[33] B. Klimt and Y. Yang. The Enron corpus: A new dataset for email classification research. In J. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Proceedings of the Fifteenth European Conference on Machine Learning*, pages 217–226. Springer, 2004.

[34] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[35] G. Lastra, O. Luaces, and A. Bahamonde. Interval prediction for graded multi-label classification. *Pattern Recognition Letters*, 49:171–176, 2014.

[36] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch. A shared task involving multi-label classification of clinical free text. In *Biological, Translational, and Clinical Language Processing (Proceedings of the Workshop on BioNLP 2007)*, pages 97–104. Association for Computational Linguistics, 2007.

[37] I. Pillai, G. Fumera, and F. Roli. Multi-label classification with a reject option. *Pattern Recognition*, 46(8):2256–2266, 2013.

[38] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[39] H. Sajnani, V. Saini, K. Kumar, E. Gabrielova, P. Choudary, and C. Lopes. Multilabel calssification of reviews in Yelp data. Technical report, University of California Irvine, 2014.

[40] K. Sechidis, G. Tsoumakas, and I. Vlahavas. On the stratification of multi-label data. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases (Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases)*, volume 6913 of *Lecture Notes in Computer Science*, pages 145–158. Springer, 2011.

[41] C.G.M. Snoek, M. Worring, J.C. van Gemert, J. Geusebroek, and A.W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the Fourteenth ACM International Conference on Multimedia*, pages 421–430, 2006.

[42] M.C.M. Troffaes. Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, 45(1):17–29, 2007.

[43] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multi-label classification of music into emotions. In *Proceedings of the Ninth International Conference on Music Information Retrieval*, pages 325–330, 2008.

[44] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic-description using the CAL500 data set. In *Proceedings of the Thirtieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 439–446, 2007.

[45] L.C. van der Gaag and P.R. de Waal. Multi-dimensional Bayesian network classifiers. In M. Studený and J. Vomlel, editors, *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pages 107–114. Action M, 2006.

[46] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, 1991.

[47] P. Walley. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society: Series B*, 58:3–34, 1996.

[48] I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[49] Z. Younes, F. Abdallah, and T. Denoeux. Fuzzy multi-label learning under veristic variables. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1696–1703. IEEE, 2010.

[50] M. Zaffalon. Statistical inference of the naive credal classifier. In G. de Cooman, T.L. Fine, and T. Seidenfeld, editors, *Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications*, pages 384–393. Shaker, 2001.

[51] M. Zaffalon, G. Corani, and D.D. Mauá. Evaluating credal classifiers by utility-discounted predictive accuracy. *International Journal of Approximate Reasoning*, 53(8):1282–1301, 2012.

[52] M. Zaffalon and E. Fagiuoli. Tree-based credal networks for classification. *Reliable Computing*, 9(6):487–509, 2003.

Appendix A. Proofs

Proof of Theorem 1. *Let us consider the objective function in Eq.* (6) *by distinguishing whether or not the two sequences* $\boldsymbol{c}'$ *and* $\boldsymbol{c}''$ *share the same state for the first label, i.e.,*

$$\frac{P_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'', \boldsymbol{f})} = \begin{cases} G_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{c}'', \boldsymbol{f}), & \text{if } c'_1 = c''_1\,, \\ H_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{c}'', \boldsymbol{f}), & \text{if } c'_1 = \neg c''_1\,. \end{cases} \tag{A.1}$$

Following Eq. (4)*, the first function rewrites as:*

$$G_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{c}'', \boldsymbol{f}) = \prod_{i=2}^{q} \delta(c'_i, \neg c''_i) \left[\frac{n(c'_1, c'_i) + st(c'_1, c'_i)}{n(c''_1, c''_i) + st(c''_1, c''_i)} \prod_{k=1}^{m} \frac{\frac{n(c'_i, f_k) + st(c'_i, f_k)}{n(c'_i) + st(c'_i)}}{\frac{n(c''_i, f_k) + st(c''_i, f_k)}{n(c''_i) + st(c''_i)}} \right],$$

where the delta function emphasises the fact that the contribution of the terms with $c'_i = c''_i$ *is one (remember that* $c'_1 = c''_1$*). A first optimization with respect to the constraints can be achieved as in Section 4 by setting* $t(c'_i, f_k) \to 0$ *and* $t(c''_i, f_k) \to t(c''_i)$ *(remember that* $c'_i = \neg c''_i$*). Similarly, we set* $t(c'_1, c'_i) \to 0$ *and* $t(c''_1, c''_i) \to t(c''_1)$*. After these intermediate optimization, the objective function rewrites as:*

$$\prod_{i=2}^{q} \delta(c'_i, \neg c''_i) \left[\frac{n(c'_1, c'_i)}{n(c''_1, c''_i) + st(c''_1)} \prod_{k=1}^{m} \frac{\frac{n(c'_i, f_k)}{n(c'_i) + st(c'_i)}}{\frac{n(c''_i, f_k) + st(c''_i)}{n(c''_i) + st(c''_i)}} \right],$$

The optimization w.r.t. $t(c''_1)$ *is achieved in the limit* $t(c''_1) \to 1$*. Even the remaining optimization tasks can be achieved one independently of the others. The result is the left-hand side of Eq.* (9)*, where, to obtain the expression in Eq.* (11)*, we set* $t_i := t(c''_i)$*, and hence* $t(c'_i) = 1 - t_i$ *(remember that, for these terms,* $c'_i = \neg c''_i$*).*

We similarly proceed for $H_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{c}'', \boldsymbol{f})$. Following Eq. (A.1) *and Eq.* (4)*, the function rewrites as:*

$$H_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{c}'', \boldsymbol{f}) = \left[\frac{n(c_1'') + st(c_1'')}{n(c_1') + st(c_1')}\right]^{q+m-2} \prod_{k=1}^{m} \frac{n(c_1', f_k) + st(c_1', f_k)}{n(c_1'', f_k) + st(c_1'', f_k)} \times$$
$$\times \prod_{i=2}^{q} \frac{n(c_1', c_i') + st(c_1', c_i')}{n(c_1'', c_i'') + st(c_1'', c_i'')} \left[\prod_{j=2}^{q} \delta(c_j', \neg c_j'') \prod_{k=1}^{m} \frac{\frac{n(c_j', f_k) + st(c_j', f_k)}{n(c_j') + st(c_j')}}{\frac{n(c_j'', f_k) + st(c_j'', f_k)}{n(c_j'') + st(c_j'')}}\right].$$

As in the previous case, we perform some optimization, rename the remaining variables, and independently optimise w.r.t. t_i ($i > 1$). Afterwards, the optimization with respect to t_1 gives the expression in the left-hand side of Eq. (10)*. Finally, we prove the convexity of the objective functions. The derivative of the logarithm of the objective function in the right-hand side of Eq.* (11) *divided by the positive constant s is equal to:*

$$\frac{m}{n(c_i') + s(1 - t_i)} - \sum_{k=1}^{m} \frac{1}{n(c_i'', f_k) + st_i} + \frac{m}{n(c_i'') + st_i}\,.$$

The second derivative, again divided by s, is:

$$\frac{m}{[n(c_i') + s(1 - t_i)]^2} + \sum_{k=1}^{m} \frac{1}{[n(c_i'', f_k) + st_i]^2} - \frac{m}{[n(c_i'') + st_i]^2}\,,$$

and its nonnegativity easily follows from $n(c_i'') \geq n(c_i'', f_k)$. Similarly, the second derivative of the logarithm of the objective function in Eq. (10) *is:*

$$-\frac{q + m - 2}{[n(c_1'') + st_1]^2} + \frac{q + m - 2}{[n(c_1') + s(1 - t_1)]^2}$$
$$+ \sum_{k=1}^{m} \frac{1}{[n(c_1'', f_k) + st_1]^2} + \sum_{i=2}^{q} \frac{1}{[n(c_1'', c_i'') + st_1]^2}\,.$$

As in the previous case, the nonnegativity follows from $n(c_i'') \geq n(c_i'', f_k)$. □

To prove Theorem 2, two simple preparatory results are needed.

Proposition 1. *Given two arrays $\vec{a}$ and $\vec{b}$ with the same length n, the following inequality holds:*

$$\min_i \max\{a_i, b_i\} \geq \max\{\min_i a_i, \min_i b_i\}\,,$$

where a_i and b_i are the i-th elements of $\vec{a}$ and $\vec{b}$, and the minima are intended w.r.t. $i = 1, \ldots, n$.

Proof. We prove the result by contradiction. Thus, we assume that:

$$\min_i \max\{a_i, b_i\} < \max\{\min_i a_i, \min_i b_i\}\,. \tag{A.2}$$

Let i^* denote the arg min of the left-hand side. If, without any lack of generality, we assume $\min_i a_i \geq \min_i b_i$, Eq. (A.2) rewrites as:

$$\max\{a_{i^*}, b_{i^*}\} < \min_i a_i \,.$$

If $a_{i^*} > b_{i^*}$, we obtain the contradiction $a_{i^*} < \min_i a_i$. Otherwise, we have:

$$a_{i^*} \leq b_{i^*} < \min_i a_i \,,$$

which is also a contradiction. □

Proposition 2. *Consider the inequality*

$$\frac{a}{n+t} > \frac{b}{m+t} \tag{A.3}$$

with $a, b, n, m > 0$ and $0 \leq t \leq 1$. If the inequality is true for $t = 0$ and $t = 1$, then it is true for each t.

Proof. To prove the result ad absurdum, assume that there is a t^* such that

$$\frac{a}{n+t} - \frac{b}{m+t} = \frac{am + at - bn - bt}{(n+t)(m+t)} < 0 \,.$$

As the denominator of the second term is always positive, this means that the numerator should be negative. But if we write the inequality in Eq. (A.3) for $t = 0$ and $t = 1$, i.e.,

$$\begin{aligned} am - bn &> 0 \,, \\ am - bn + (a - b) &> 0 \,, \end{aligned}$$

and we multiply the first inequality by $(1 - t)$, the second by t, and we sum them, we obtain a contradiction. □

Proof of Theorem 2. *Let us write the optimum on the left-hand side of Eq.* (8) *in a more explicit form:*

$$\min_{c_1''} \min_{c_2'', \ldots, c_q'' : c_l'' = \tilde{c}_l} \max_{c_1'} \max_{c_2', \ldots, c_q'} \inf_{\boldsymbol{t}} \frac{P_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'', \boldsymbol{f})} \,. \tag{A.6}$$

We swap the maximization over c_1' with the minimization over $c_2'', \ldots, c_q''$. This produces a different optimization task, i.e.,

$$\min_{c_1''} \max_{c_1'} \min_{c_2'', \ldots, c_q'' : c_l'' = \tilde{c}_l} \max_{c_2', \ldots, c_q'} \inf_{\boldsymbol{t}} \frac{P_{\boldsymbol{t}}(\boldsymbol{c}', \boldsymbol{f})}{P_{\boldsymbol{t}}(\boldsymbol{c}'', \boldsymbol{f})} \,, \tag{A.7}$$

which, because of Proposition 1, gives an upper bound for Eq. (A.6). *We prove the theorem by showing that Algorithm 1 returns that bound. To do that, let us*

initially assume $l > 1$. We split the optimisation with respect to c'_1 and c''_1 from the others by setting:

$$\Phi(c'_1, c''_1, \boldsymbol{f}) := \min_{c''_2,\ldots,c''_q : c''_l = \tilde{c}_l} \max_{c'_2,\ldots,c'_q} \inf_{\boldsymbol{t}} \frac{P_{\boldsymbol{t}}(c'_1, c'_2, \ldots, c'_q, \boldsymbol{f})}{P_{\boldsymbol{t}}(c''_1, c''_2, \ldots, c''_q, \boldsymbol{f})}\,, \tag{A.8}$$

The right-hand side of Eq. (A.7) *can be therefore computed by evaluating the four possible values of $\Phi(c'_1, c''_1, \boldsymbol{f})$. This is the evaluation in line 27 of the algorithm.*

First consider the case where the two sequences share the same state for the first label, i.e., $\Phi(c'_1 = c_1, c''_1 = c_1, \boldsymbol{f})$. By exploiting the relation in Eq. (9) *of Theorem 1, we obtain:*

$$\min_{c''_2,\ldots,c''_q : c''_l = \tilde{c}_l} \max_{c'_2,\ldots,c'_q} \prod_{i=2}^{q} \delta(c'_i, \neg c''_i) \frac{n(c_1, c'_i) \cdot g_i(c'_i, c''_i, \boldsymbol{f})}{n(c_1, c''_i) + s}\,. \tag{A.9}$$

As the values of the first label in the two sequences are given, the objective function in Eq. (A.9) *factorises over the other labels. Accordingly we can perform each optimization separately, and rewrite Eq.* (A.9) *as follows:*

$$\prod_{\substack{i=2,\ldots,q \\ i \neq l}} \min\left\{\max\left\{1, \frac{n(c_1,1) g_i(1,0,\boldsymbol{f})}{n(c_1,0)+s}\right\}, \max\left\{\frac{n(c_1,0) g_i(0,1,\boldsymbol{f})}{n(c_1,1)+s}, 1\right\}\right\} \times$$
$$\times \max\left\{1, \frac{n(c_1, \neg\tilde{c}_l) \cdot g_l(\neg\tilde{c}_l, \tilde{c}_l, \boldsymbol{f})}{n(c_1, \tilde{c}_l) + s}\right\}.$$

This is the expression computed in lines 5 and 6 of Algorithm 1.

If the two sequences have opposite values in the first label, Eq. (A.8) *can be evaluated as in Eq.* (10) *in Theorem 1. Accordingly, $\Phi(c'_1 = c_1, c''_1 = \neg c_1, \boldsymbol{f})$ becomes:*

$$\min_{c''_2,\ldots,c''_q : c''_l = \tilde{c}_l} \max_{c'_2,\ldots,c'_q} \inf_{0<t_1<1} \left[h(c_1, \neg c_1, t_1, \boldsymbol{f}) \prod_{i=2}^{q} \frac{n(c_1, c'_i)\tilde{g}_i(c'_i, c''_i, \boldsymbol{f})}{n(\neg c_1, c''_i) + s t_1} \right]. \tag{A.10}$$

If $\tilde{t}_1(c'_2, \ldots, c'_q, c''_2, \ldots, c''_q)$ denotes the value of $t_1 \in [0,1]$ for which the infimum in Eq. (A.10) *is achieved, with some abuse of notation we can confuse the limits with the actual values and rewrite Eq.* (A.10) *as:*

$$\min_{c''_2,\ldots,c''_q : c''_l = c_l} \max_{c'_2,\ldots,c'_q} h(c_1, \neg c_1, \tilde{t}_1, \boldsymbol{f}) \prod_{i=2}^{q} \frac{n(c_1, c'_i)\tilde{g}_i(c'_i, c''_i, \boldsymbol{f})}{n(\neg c_1, c''_i) + s\tilde{t}_1}\,.$$

Unlike Eq. (A.9)*, the above expression does not factorise with respect to the different labels. The reason is the fact that the value of $\tilde{t}_1$ might depend on the actual values of the labels. Yet, for each $t_1 \in]0,1[$, consider the task:*

$$\min_{c''_i} \max_{c'_i} \frac{n(c_1, c'_i)\tilde{g}_i(c'_i, c''_i, \boldsymbol{f})}{n(\neg c_1, c''_i) + s t_1}\,, \tag{A.11}$$

which, for $i \neq l$, can be made more explicit as follows:

$$\min \left\{ \frac{\max\{n(c_1,0), n(c_1,1) g_i(1,0,\boldsymbol{f})\}}{n(\neg c_1, 0) + s t_1}, \right. \\ \left. \frac{\max\{n(c_1,0) g_i(0,1,\boldsymbol{f}), n(c_1,1)\}}{n(\neg c_1, 1) + s t_1} \right\} . \tag{A.12}$$

Once the two maxima in the above expression are identified, whether or not the minimum is the first or the second term might depend on the particular value of t_1. According to Proposition 2, if the minimum if the same for both $t_1 = 0$ and $t_1 = 1$, we have that the values of c_i' and c_i'' leading to the optimum are the same for each t_1. If this is not the case, we can add a further outer approximation by considering the task for $s = 0$. The value of c_i' and c_i'' leading to the above considered solution of Eq. (A.12) *are those returned by line 11 of Algorithm 1. For $i = l$ the task becomes simpler, as the value of the second label is fixed, the task is only a maximization among two options, which are independent of t_1. This is what is done in line 8 of Algorithm 1. If the values of c_i' and c_i'' leading to the optimum in Eq.* (A.11) *are the same for each t_1, we can safely put these values into the objective function. The task becomes therefore a simple dominance test as in Eq.* (10)*. This is what we do in line 14. The case $l = 1$ is simpler. We leave to the reader the discussion of this case, which corresponds to lines 19-25 in the algorithm.* □