Credal Ensembles of Classifiers

G. Corani[a,*], A. Antonucci[a]

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

Abstract

It is studied how to aggregate the probabilistic predictions generated by different SPODE (Super-Parent-One-Dependence Estimators) classifiers. It is shown that aggregating such predictions via compression-based weights achieves a slight but consistent improvement of performance over previously existing aggregation methods, including Bayesian Model Averaging and simple average (the approach adopted by the AODE algorithm). Then, attention is given to the problem of choosing the prior probability distribution over the models; this is an important issue in any Bayesian ensemble of models. To robustly deal with the choice of the prior, the single prior over the models is substituted by a set of priors over the models (*credal set*), thus obtaining a credal ensemble of Bayesian classifiers. The credal ensemble recognizes the *prior-dependent* instances, namely the instances whose most probable class varies when different prior over the models are considered. When faced with prior-dependent instances, the credal ensemble remains reliable by returning a set of classes rather than a single class. Two credal ensembles of SPODEs are developed; the first generalizes the Bayesian Model Averaging and the second the compression-based aggregation. Extensive experiments show that the novel ensembles compare favorably to traditional methods for aggregating SPODEs and also to previous credal classifiers.

Keywords: classification, Bayesian model averaging, compression-based averaging, AODE, credal classification, imprecise probability, credal ensemble

1. Introduction

Bayesian model averaging (BMA) (Hoeting et al., 1999) is a sound approach to address the uncertainty which characterizes the identification of the supposedly best model; given a set of alternative models, BMA weights the inferences produced by the models using as weights the models' posterior probabilities. BMA assumes one of the models in the ensemble to be the true one. Under this assumption, BMA is expected to achieve better predictive accuracy than the choice of any single model (Hoeting et al., 1999). However, such assumption is mostly violated; as a consequence, BMA generally does *not* achieve high predictive performance in experiments. The problem is that

*Corresponding author: giorgio@idsia.ch

BMA gets excessively concentrated around the single most probable model (Domingos, 2000; Minka, 2002): especially on large data sets, "*averaging using the posterior probabilities to weight the models is almost the same as selecting the MAP model*" (Boullé, 2007, page 1672), where the MAP model is the most probable model a posteriori. To address this problem, the *compression-based* approach (Boullé, 2007) applies a logarithmic smoothing to the models posterior probabilities, thus computing weights that are more evenly-distributed. The compression-based weights can be justified from an information-theoretic viewpoint and have been used to combine naive Bayes classifiers characterized by different feature sets, obtaining excellent results in prediction contexts (Boullé, 2007).

The Averaged One-Dependence Estimators (AODE) (Webb et al., 2005) is an ensemble of SPODE (SuperParent-One-Dependence Estimator) classifiers. Each SPODE adopts a certain feature as a *super-parent*, namely it assumes all features to depend on a common feature (the super-parent) in addition to the class. The AODE ensemble simply averages the posterior probabilities computed by the different SPODEs; remarkably, AODE is more effective than the majority of rival schemes for aggregating SPODEs, including BMA (Yang et al., 2007).

As a preliminary step we develop BMA-AODE, namely BMA over SPODEs, with some computational differences with respect to the previous frameworks of Yang et al. (2007) and Cerquides and de Mántaras (2005); we confirm however that BMA over SPODEs is outperformed by AODE. Then we develop the novel COMP-AODE classifier, which weights the SPODEs using the compression-based coefficients and yields a slight but consistent improvement in the classification performance over AODE. Considering the high performance of AODE, this result is noteworthy.

An important issue in any Bayesian ensemble of models is however the choice of the prior over the models. Most commonly it is adopted a uniform prior or a prior which favors simpler models over complex ones (Boullé, 2007). Although such choices are reasonable, the specification of any single prior implies some arbitrariness and entails the risk of drawing prior-dependent conclusions, especially on small data sets. In fact, the specification of the prior over the models is a serious issue for Bayesian ensembles.

To overcome the problem, we take inspiration from the field of imprecise probability (IP) (Walley, 1991). IP approaches can be used to generalize Bayesian models, describing prior uncertainty by a set of prior distributions instead of a single prior; see Walley (1996) for a deep discussion of the reasons for preferring IP approaches to the specification of a single prior. In particular, classifiers based on a set of priors are called *credal classifiers*; they automatically detect the prior-dependent instances, namely the instances whose most probable class varies under different priors. On the prior-dependent instances, traditional classifiers are unreliable (Corani and Zaffalon, 2008b); on the same instances credal classifier remain instead reliable by returning a set of classes. A survey of credal classifiers can be found in (Corani et al., 2012).

In this paper, we extend the idea of *credal model averaging* (CMA) (Corani and Zaffalon, 2008a). Credal model averaging generalizes Bayesian model averaging; it substitutes the single prior over the models by a *set* of priors (credal set). CMA is therefore a credal classifier which can be seen as a credal ensemble of Bayesian classifiers. In (Corani and Zaffalon, 2008a), the application of CMA was limited to the case of naive Bayes.

In this paper we develop BMA-AODE* and COMP-AODE*, namely the credal ensembles which respectively generalize BMA-AODE and COMP-AODE, substituting the uniform prior over the models by a credal set. By extensive experiments we show that both credal ensembles compare favorably to both their single-prior counterparts and to previously existing credal classifiers, including the CMA of (Corani and Zaffalon, 2008a).

2. Methods

We consider a classification problem with k *features*; we denote by C the *class* variable (taking values in $\mathcal{C}$) and by $\mathbf{A} := (A_1, \ldots, A_k)$ the set of features, taking values respectively in $\mathcal{A}_1, \ldots, \mathcal{A}_k$. For a generic variable A, we denote as $P(A)$ the probability mass function over its values and as $P(a)$ the probability that $A = a$. We assume the data to be complete and the training data $\mathcal{D}$ to contain n instances. We estimate the parameters of the SPODEs adopting the usual Bayesian approach for generative models (Heckerman, 1995), namely by setting Dirichlet priors and then taking expectation from the parameter posterior distribution.

Under 0-1 loss a traditional probabilistic classifier returns, for a test instance whose class in unknown, say $\tilde{\mathbf{a}} := \{\tilde{a_1}, \ldots, \tilde{a_k}\}$, the most probable class c^*:

$$c^* := \arg\max_{c \in \mathcal{C}} P(c|\tilde{\mathbf{a}}).$$

Credal classifiers change this paradigm, by occasionally returning more classes; this happens in particular when the most probable class is *prior-dependent*. We discuss this point more in detail later, when presenting credal classifiers.

2.1. From Naive Bayes to AODE

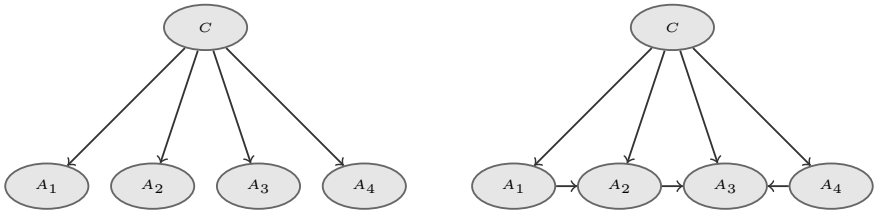
The Naive Bayes classifier assumes the stochastic independence of the features given the class; it therefore factorizes the joint probability as follows:

$$P(c, \mathbf{a}) := P(c) \cdot \prod_{j=1}^{k} P(a_j|c), \tag{1}$$

corresponding to the topology of Fig.1(a). Despite the biased estimate of probabilities due to the above (so-called *naive*) assumption, naive Bayes performs well under 0-1 loss (Domingos and Pazzani, 1997); it thus constitutes a reasonable choice if the goal is simple classification, without the need for accurate probability estimates (Friedman, 1997).

The naive independence assumption is relaxed for instance by the tree-augmented naive classifier (TAN), which allows the subgraph involving only the features to be a tree, allowing thus each feature to depend on the class *and* on another feature; an example is shown in Fig.1(b). Generally, TAN outperforms naive Bayes in classification (Friedman et al., 1997).

The AODE classifier (Webb et al., 2005) is an ensemble of k SPODE (SuperParent One Dependence Estimator) classifiers; each SPODE is characterized by a certain



(a) Naive Bayes. (b) A possible TAN structure.

Figure 1: Naive Bayes vs TAN.

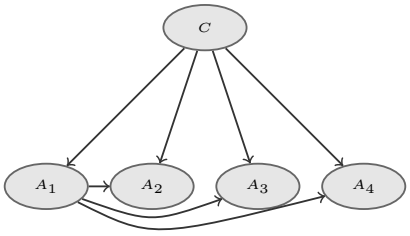


Figure 2: SPODE with super-parent A_1.

super-parent feature, so that the remaining features are children of both the class and the super-parent, as shown in in Fig.2. In fact, each single SPODE is a TAN.

We denote the set of SPODEs as $\mathcal{S} := \{s_1, \ldots, s_k\}$, where s_j indicates the SPODE with super-parent A_j. SPODE s_j factorizes the joint probability as:

$$P(c, \mathbf{a}|s_j) = P(c) \cdot P(a_j|c) \cdot \prod_{l=1, l\neq j}^{k} P(a_l|a_j, c).$$

In order to classify the test instance $\tilde{\mathbf{a}}$, AODE averages the posterior probability $P(c|\tilde{\mathbf{a}}, s_j)$ computed by each single SPODE:

$$P(c|\mathbf{a}) \propto P(c, \mathbf{a}) := \frac{1}{k} \sum_{j=1}^{k} P(c, \mathbf{a}|s_j).$$

In this paper we focus on more sophisticated approaches for aggregating the predictions of the SPODEs.

2.2. Bayesian Model Averaging (BMA) with SPODEs

By ensembling the SPODEs via BMA we assume one of the SPODEs to be the true model. We thus introduce a variable S over $\mathcal{S}$, where $P(S = s_j)$ denotes the *prior* probability of SPODE s_j to be the true model. Each SPODE has the same number of variables, the same number of arcs and the same in-degree (the maximum number of parents per node). Thus, we adopt a *uniform* prior, assigning prior probability $1/k$ to each SPODE. In fact, the uniform prior over the models is frequently adopted when

implementing BMA. To classify the test instance $\tilde{\mathbf{a}}$, BMA computes the following posterior mass function:

$$P(c|\tilde{\mathbf{a}}) := \sum_{j=1}^{k} P(c|\tilde{\mathbf{a}}, s_j) \cdot P(s_j|\mathcal{D}) \propto \sum_{j=1}^{k} P(c|\tilde{\mathbf{a}}, s_j) \cdot P(\mathcal{D}|s_j) \cdot P(s_j),$$

where $P(\mathcal{D}|s_j) = \int_{\boldsymbol{\theta}_j} P(\mathcal{D}|s_j, \boldsymbol{\theta}_j) P(\boldsymbol{\theta}_j) d\boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_i$ denote respectively the *marginal likelihood* and the parameters of s_j. Under certain assumptions, the marginal likelihood can be analytically computed (Heckerman, 1995); this is the approach adopted by both (Cerquides and de Mántaras, 2005) and (Yang et al., 2007) to implement BMA over SPODEs. Such papers report that BMA over SPODEs is outperformed by AODE. This can be due, besides the already discussed excessive concentration of BMA on the MAP model, to the adoption of the marginal likelihood to compute BMA. The marginal likelihood measures how good the model is at representing the *joint* distribution; instead, a classifier has to estimate the posterior probability of the classes *conditionally* on the observed features. Therefore, a high marginal likelihood does not necessarily imply a good classification performance (Cowell, 2001; Kontkanen et al., 1999). Following Boullé (2007), we thus substitute the marginal likelihood with the *conditional* likelihood:

$$L_j := \prod_{i=1}^{n} P(c^{(i)}|\mathbf{a}^{(i)}, s_j) = \prod_{i=1}^{n} \int_{\boldsymbol{\theta}_j} P(c^{(i)}|\mathbf{a}^{(i)}, s_j, \boldsymbol{\theta}_j) P(\boldsymbol{\theta}_j|\mathcal{D}) d\boldsymbol{\theta}_j. \tag{2}$$

We call BMA-AODE the classifier which estimates the posterior probabilities of the class, given the test instance $\tilde{\mathbf{a}}$, as follows:

$$P(c|\tilde{\mathbf{a}}) \propto \sum_{j=1}^{k} P(c|\tilde{\mathbf{a}}, s_j) \cdot L_j \cdot P(s_j). \tag{3}$$

Especially on large data sets, the difference between the likelihoods of the different SPODEs might be of several order of magnitudes. We remove from the ensemble the SPODEs whose conditional likelihood is smaller than $L_{\max}/10^4$, where $L_{\max}$ is the maximum conditional likelihood among all SPODEs. Discarding models with very low posterior probability is in fact common when dealing with BMA and can be seen as a *belief revision* (Dubois and Prade, 1997). Given the joint beliefs $P(X, Y)$, the *revision* $P'(X, Y)$ induced by a marginal $P'(Y)$ is defined by $P'(x, y) := P(x|y) \cdot P'(y)$. In other words, if $P'(y)$ is known to be a better model than $P(y)$ for the marginal beliefs about y, this information can be used in the above described way to redefine the joint. Accordingly, in BMA-AODE, the marginal beliefs about S have been replaced by a better candidate, inducing a revision in the corresponding joint model.

2.2.1. Exponentiation of the Log-Likelihoods

Regardless whether the marginal likelihood or the conditional likelihood is considered, it is common to compute the *log-likelihood* rather than the likelihood, in order to avoid numerical problems due to the multiplication of many probabilities. However, the log-likelihoods can become very negative on large data sets; in this case, their

exponentiation can incur into numerical problems. This issue has forced for instance the usage of high numerical precision, causing a slowdown of the computation: "*BMA often lead to arithmetic overflow when calculating very large exponentials or factorials. One solution is to use the Java class BigDecimal which unfortunately can be very slow.*" (Yang et al., 2007, pag. 1660).

We instead adopt the procedure of Algorithm 1, communicated to us by Dr. Dash who published several works on BMA (Dash and Cooper, 2004). The procedure robustly exponentiates the log-likelihoods using standard numerical precision.

Algorithm 1 Robust exponentiation of log-likelihoods.

Require: Array `log_liks` of log-likelihoods, assumed of length `k`.

```
minVal=min(log_liks)

for i = 1:k do
  shifted_logliks(i)=logliks(i)-minVal;
  tmp_liks(i)=exp(shifted_logliks(i));
end for

total=sum(tmp_liks)

for i = 1:k do
  liks(i)=tmp_liks(i)/total;
end for

return liks {Array of likelihoods exponentiated and normalized.}
```

2.3. BMA-AODE: Extending BMA-AODE to Sets of Probabilities*

By BMA-AODE* we extend BMA-AODE to *imprecise probabilities* (Walley, 1991), substituting the single prior mass function $P(S)$ over the models by a set of priors (credal set). The credal set $\mathcal{P}(S)$, defined over S, lets the prior probability of each SPODE vary within a range rather than being a fixed number. BMA-AODE* is a credal ensemble of Bayesian classifiers, since the parameters of each SPODE are learned in the standard Bayesian way. In principle we could let the prior probability of each SPODE vary exactly between zero and one (*vacuous* model). Yet, this would generate vacuous posterior inferences and prevent learning from data (Piatti et al., 2009). To obtain non-vacuous posterior inferences, we introduce a lower bound ϵ for the prior probability of each model. The credal set is defined by the following constraints:

$$\mathcal{P}(S) := \left\{ P(S) \;\middle|\; \begin{array}{ll} P(s_j) \geq \epsilon & \forall j = 1, \ldots, k \\ \sum_{j=1}^{k} P(s_j) = 1 & \end{array} \right\}. \tag{4}$$

The prior probability of each SPODE varies thus between ϵ and $1 - (k-1)\epsilon$.

The credal set in (4) represents *ignorance* about the prior probability of each SPODE being the true model. Since $\mathcal{P}(S)$ is a set a prior mass functions, BMA-AODE* can be regarded as a set of BMA-AODE classifiers, each corresponding to a different prior. If the most probable class of an instance varies under different priors, the classification is *prior-dependent*. When dealing with prior-dependent instances, credal classifiers (Corani et al., 2012; Corani and Zaffalon, 2008b) become *indeterminate*, by returning a set of classes instead of a single class.

Before discussing how this set of classes is identified, we need introducing the concept of *credal dominance* (or, for short, *dominance*): class c' *dominates* class c'' if c' is more probable than c'' under each prior of the credal set. If no class dominates c', then c' is non-dominated. Credal classifiers return in particular all the *non-dominated* classes, which are identified by performing different pairwise dominance tests among classes. This criterion is called *maximality* (Walley, 1991, Section 3.9.2) and is described by Algorithm 2. If all classes are non-dominated, maximality requires performing $|\mathcal{C}| \cdot (|\mathcal{C}| - 1)$ tests; this is the worst case in terms of computational complexity. However, once a dominated classes is identified, it can be ignored in the following tests, thus reducing the computational burden. While there exist different criteria for taking decisions under imprecise probabilities (Troffaes, 2007), maximality is the criterion commonly used by credal classifiers.

Algorithm 2 Identification of the non-dominated classes $\mathcal{ND}$ through maximality

$\mathcal{ND} := \mathcal{C}$

for $c' \in \mathcal{ND}$ **do**
  for $c'' \in \mathcal{ND}$ $(c'' \neq c')$ **do**
    check whether c' dominates c''
    if c' dominates c'' **then**
      $\mathcal{ND} \leftarrow \mathcal{ND} \setminus c''$
    end if
  end for
end for

return $\mathcal{ND}$

Non-dominated classes are incomparable, which means that there is no available information to rank them. Credal classifiers can be thus seen as dropping the dominated classes and expressing indecision about the non-dominated ones.

Following (3), within BMA-AODE*, c' dominates c'' if the solution of the following optimization problem is greater than zero:

$$\begin{aligned} \text{minimize:} \quad & \sum_{j=1}^{k} P(c'|\tilde{\mathbf{a}}, s_j) \cdot L_j \cdot P(s_j) - \sum_{j=1}^{k} P(c''|\tilde{\mathbf{a}}, s_j) \cdot L_j \cdot P(s_j) \\ \text{with respect to:} \quad & P(s_1), \ldots, P(s_k) \end{aligned} \tag{5}$$

$$\text{subject to:} \qquad P(s_j) \geq \epsilon \quad \forall j = 1, \ldots, k$$
$$\sum_{j=1}^{k} P(s_j) = 1.$$

The constraints of problem (5) are constituted by the definition of credal set; the problem is a linear programming task and as such it can be solved in polynomial time (Karmakar, 1984). As already discussed for BMA-AODE, we include in the computation only the SPODEs whose conditional likelihood is at least $L_{\max}/10^4$. This can be regarded as a belief revision process, involving the credal set. The marginal credal set $\mathcal{P}'(Y)$ induces the following revision of the joint credal set $\mathcal{P}(X, Y)$:

$$\mathcal{P}'(X,Y) := \left\{ P'(X,Y) \;\middle|\; \begin{array}{l} P'(x,y) := P(x|y) \cdot P'(y) \\ P'(Y) \in \mathcal{P}'(Y) \end{array} \right\}.$$

The uniform prior belongs to the credal set of BMA-AODE*; therefore, the set of non-dominated classes identified by BMA-AODE* includes by design the most probable class returned by BMA-AODE; if in particular BMA-AODE* returns a single non-dominated class, this coincides with the class returned by BMA-AODE.

2.4. Compression-Based Averaging

Compression-based averaging has been introduced in (Boullé, 2007) to mitigate the excessive concentration of BMA around the MAP model; it replaces the posterior probabilities $P(s_j|\mathcal{D})$ of the models by smoother *compression weights*, which we denote as $P'(s_j|\mathcal{D})$ for model s_j. Note that adopting the compression coefficients in place of the posterior probabilities can be seen as a belief revision; for this reason, we adopt again the notation P'.

To present the method, we need some further notation. In particular, we denote by LL_j the *log* of the conditional likelihood of model s_j. We moreover introduce the *null classifier*, which classifies instances by simply using the marginal probabilities of the classes, without conditioning on the features. The null classifier will be used for the computation of the compression coefficients. We denote the null classifier as s_0; therefore we associate a further state s_0 to S, whose domain thus becomes $\{s_0, s_1, \ldots, s_k\}$. We denote as LL_0 the conditional log-likelihood of the null classifier and as $H(C) := -\sum_{c \in \mathcal{C}} P(c) \log P(c)$ the sample estimate of the entropy of the class variable. Assuming $P(c)$ to be estimated by maximum likelihood (unlike the parameters of the SPODEs, which are instead estimated in a Bayesian way), then $LL_0 = -nH(C)$ (Boullé, 2007).

Since we are dealing with a traditional single-prior classifier, we set a single prior mass function over the models, assigning uniform prior probability to the various SPODEs and prior probability ϵ to the null model; assigning a prior probability to the null model is necessary, since its posterior probability appears in the compression coefficients. Thus, we define the prior over variable S as follows:

$$P(s_j) = \left\{ \begin{array}{ll} \epsilon & j = 0, \\ \frac{1-\epsilon}{k} & j = 1, \ldots, k. \end{array} \right. \tag{6}$$

The compression coefficients are computed in two steps: computation of the *raw* compression coefficients and normalization. The *raw* compression coefficient associated to SPODE s_j is:

$$\pi_j := 1 - \frac{\log P(s_j|\mathcal{D})}{\log P(s_0|\mathcal{D})} = 1 - \frac{LL_j + \log P(s_j)}{LL_0 + \log P(s_0)} = 1 - \frac{LL_j + \log \frac{1-\epsilon}{k}}{-nH(C) + \log \epsilon}. \quad (7)$$

A negative π_j means that s_j is a worse predictor than the null model; a positive π_j means that s_j is a better predictor than the null model, which is the general case in practical situations. The upper limit of π_j is one: in this case s_j is a perfect predictor, with likelihood 1, and thus log-likelihood 0. Following (Boullé, 2007), we keep in the ensemble only the *feasible* models, namely those with $\pi_j > 0$, and we discard the models with $\pi_j \leq 0$; therefore, the null model is not part of the resulting ensemble. The procedure corresponds to a belief revision induced by the removal from the ensemble of the models whose posterior probability falls below a certain threshold. An information-theoretic interpretation of the compression weights can be given recalling that the principle of minimum description length (MDL) (Grünwald, 2005) prescribes to minimize the overall description length of model and data given the model. The description length correspond to the logarithm of the posterior probability, namely the logarithm of the prior (interpreted as the code length of the model) plus the logarithm of the likelihood (interpreted as the code length of the data given the model). Thus, $LL_j + \log P(s_j)$ "*represents the quantity of information required to encode the model plus the class values given the model. The code length of the null model can be interpreted as the quantity of information necessary to describe the classes, when no explanatory data is used to induce the model. Each model can potentially exploit the explanatory data to better compress the class conditional information. The ratio of the code length of a model to that of the null model stands for a relative gain in compression efficiency.*" (Boullé, 2007, pag. 1663).

With no loss of generality, assume the features to be ordered such that $A_1, A_2, \ldots, A_{\tilde{k}}$ yield a feasible model when used as super-parent; thus, SPODEs $s_1, s_2, \ldots, s_{\tilde{k}}$ are feasible, while SPODEs s_j with $j > \tilde{k}$ are removed from the ensemble. The *normalized* compression coefficients $P'(s_j|\mathcal{D})$ are obtained by normalizing the raw compression coefficients of the feasible SPODEs:

$$P'(s_j|\mathcal{D}) = \begin{cases} \frac{\pi_j}{\sum_{l=1}^{\tilde{k}} \pi_l} & \text{if } j = 1, \ldots, \tilde{k}, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The posterior probabilities of the classes are estimated as:

$$P(c|\tilde{\mathbf{a}}) := \sum_{j=1}^{k} P(c|\tilde{\mathbf{a}}, s_j) \cdot P'(s_j|\mathcal{D}). \quad (9)$$

We call this classifier COMP-AODE, where COMP stands for compression-based.

2.5. COMP-AODE: Extending COMP-AODE to Sets of Probabilities*

We extend COMP-AODE to imprecise probabilities by substituting the single prior distribution $P(S)$ over the models by the credal set $\mathcal{P}_c(S)$, the subscript '$_c$' denoting

compression. The credal set of COMP-AODE* differs from that of BMA-AODE* in that it assigns prior probability ϵ to the null model:

$$\mathcal{P}_c(S) := \left\{ P(S) \;\middle|\; \begin{array}{l} P(s_0) = \epsilon, \\ P(s_j) \geq \epsilon \quad \forall j = 1, \ldots, k, \\ \sum_{j=0}^{k} P(s_j) = 1 \end{array} \right\}. \tag{10}$$

The raw compression weight of SPODE s_j varies within the interval:

$$\pi_j \in \left[1 - \frac{LL_j + \log \epsilon}{-nH(C) + \log \epsilon},\; 1 - \frac{LL_j + \log (1 - k\epsilon)}{-nH(C) + \log \epsilon} \right]. \tag{11}$$

Since the prior used by COMP-AODE (6) belongs to the credal set of COMP-AODE*, the point estimate (7) of the compression coefficient adopted by COMP-AODE lies within the interval (11). The weights π_j cannot vary independently from each other; they are instead linked by the normalization constraint in (10).

COMP-AODE* regards SPODE s_j as non-feasible if the *upper* coefficient of compression is non-positive: this approach thus preserves all the models which are feasible, in the sense of Section 2.4, for at least a prior in the set $\mathcal{P}_c(S)$. COMP-AODE* is thus more conservative than COMP-AODE, namely it discards a lower number of models. However, generally neither COMP-AODE* nor COMP-AODE remove any SPODE from the ensemble. Since the prior adopted by COMP-AODE is contained in the credal set of COMP-AODE*, the most probable class identified by COMP-AODE is part of the non-dominated classes identified by COMP-AODE*; exception to this statement are possible only if the set of feasible SPODEs differs between COMP-AODE* and COMP-AODE. However, this never happened in our experiments.

Like BMA-AODE*, COMP-AODE* identifies the non-dominated classes through maximality (Algorithm 2). In the following, we explain how to compute the test of dominance among two classes.

Testing dominance

Without loss of generality, we assume the features to have been re-ordered, so that the first $\tilde{k}$ features yield a model with positive *upper* coefficient of compression when used as super-parent. Thus, SPODEs $\{s_1, \ldots, s_{\tilde{k}}\}$ are the feasible ones. In this case the dominance test corresponds to evaluate whether or not the solution of the following optimization problem is greater than zero.

$$\begin{array}{rcc} \text{minimize:} & \sum_{j=1}^{\tilde{k}} P(c'|\tilde{\mathbf{a}}, s_j) \cdot \pi_j - \sum_{j=1}^{\tilde{k}} P(c''|\tilde{\mathbf{a}}, s_j) \cdot \pi_j & \\ \text{with respect to:} & P(s_0), P(s_1), \ldots, P(s_k) & (12) \\ \text{subject to:} & P(s_0) = \epsilon & \\ & P(s_j) \geq \epsilon \quad \forall j = 1, \ldots, k & \\ & \sum_{j=1}^{k} P(s_j) = 1, & \end{array}$$

where the normalization term $\sum_{j=1}^{\tilde{k}} \pi_j$ has been already simplified and the constraints represent the credal set.

To express the dependence on the optimization variables of the objective function, recall that $P(s_0) = \epsilon$ and express π_j through (7). The objective function rewrites as:

$$\sum_{j=1}^{\tilde{k}} P(c'|\mathbf{a}, s_j) \cdot \left(1 - \frac{\log P(s_j) + LL_j}{\log \epsilon + LL_0}\right) - \sum_{j=1}^{\tilde{k}} P(c''|\mathbf{a}, s_j) \cdot \left(1 - \frac{\log P(s_j) + LL_j}{\log \epsilon + LL_0}\right),$$

which, removing the constant terms, becomes:

$$\sum_{j=1}^{\tilde{k}} \left(P(c''|\mathbf{a}, s_j) - P(c'|\mathbf{a}, s_j)\right) \cdot \log P(s_j), \tag{13}$$

The task is therefore a linearly constrained optimization of a non-linear objective function. However, the objective function is a sum of terms including only single optimization variables and, by separing the negative from the positive terms, can be rewritten as a difference of two convex functions. Thus, the problem reduces to a convex optimization, which can be solved with polynomial complexity.

2.6. Computational Complexity of the Classifiers

We now analyze the computational complexity of the various classifiers. We distinguish between *learning* and *classification* complexity, the latter referring to the classification of a single instance. Both the *space* and the *time* required for computations are evaluated. The orders of magnitude of these descriptors are reported as a function of the dataset size n, the number of attributes/SPODEs k, the number of classes $l := |\mathcal{C}|$, and average number of states for the attributes $v := k^{-1} \sum_{i=1}^{k} |\mathcal{A}_i|$. A summary of this analysis is in Table 1 and the discussion below.

Algorithm	Space	Time	
	learning/classification	learning	classification
AODE	$\mathcal{O}(lk^2v^2)$	$\mathcal{O}(nk^2)$	$\mathcal{O}(lk^2)$
BMA[COMP]-AODE	$\mathcal{O}(lk^2v^2)$	$\mathcal{O}(n(l+k)k)$	$\mathcal{O}(lk^2)$
BMA[COMP]-AODE*	$\mathcal{O}(lk^2v^2)$	$\mathcal{O}(n(l+k)k)$	$\mathcal{O}(l^2 \cdot poly(k))$

Table 1: Complexity of classifiers.

Let us first evaluate the AODE. For a single SPODE s_j, the tables $P(C)$, $P(A_j|C)$ and $P(A_i|C, A_j)$, with $i = 1, \ldots, k$ and $i \neq j$ should be stored, this implying space complexity $\mathcal{O}(lkv^2)$ for learning each SPODE and $\mathcal{O}(lk^2v^2)$ for the AODE ensemble. These tables should be available during learning and classification for both classifiers; thus, space requirements of these two stages are the same.

Time complexity to scan the dataset and learn the probabilities is $\mathcal{O}(nk)$ for each SPODE, and hence $\mathcal{O}(nk^2)$ for the AODE. The time required to compute the posterior probabilities is $\mathcal{O}(lk)$ for each SPODE, and hence $\mathcal{O}(lk^2)$ for AODE.

Learning BMA-AODE or COMP-AODE takes the same space as AODE, but higher computational time, due to the evaluation of the conditional likelihood as in (2). The additional computational time is $\mathcal{O}(nlk)$, thus requiring $\mathcal{O}(n(l+k)k)$ time overall. For classification, time and space complexity during learning and classification are just the same.

The credal classifiers BMA-AODE* and COMP-AODE* require the same space complexity and the same time complexity in learning of their non-credal counterparts. However, credal classifiers have higher time complexity in classification. The pairwise dominance tests in Algorithm 2 requires the solution of a number of optimization problems for each test instance which is quadratic in the number of classes. The complexity of the linear programming problem for BMA-AODE* is polynomial in the number of variables (Borgwardt, 1987); the (convex) optimization problem required COMP-AODE* has polynomial complexity too, as already discussed.

3. Experiments

We run experiments on 40 data sets, whose characteristics are given in the Appendix. On each data set we perform 10 runs of 5-fold cross-validation. In order to have complete data, we replace missing values with the median and the mode for respectively numerical and categorical features. We discretize numerical features by the entropy-based method of Fayyad and Irani (1993). For the implementation of all credal sets, we set $\epsilon = 0.01$. For pairwise comparison of classifiers over the collection of data sets we use the Wilcoxon signed-rank test, as recommended by Demšar (2006) .

3.1. Determinate Classifiers

We call *determinate* the classifiers which always return a single class, namely AODE, BMA-AODE and COMP-AODE. For determinate classifiers we measure two indicators: the accuracy, namely the percentage of correct classifications, and the Brier loss

$$\frac{1}{n_{te}} \sum_{i}^{n_{te}} \left(1 - P(c^{(i)}|\mathbf{a}^{(i)})\right)^2,$$

where n_{te} denotes the number of instances in the test set, while $P(c^{(i)}|\mathbf{a}^{(i)})$ is the probability estimated by the classifier for the true class of the i-th instance.

A preliminary finding is that adopting conditional likelihood instead of marginal likelihood is an effective refinement, which reduces of 6% on average the Brier loss of BMA-AODE. Despite this refinement, however, BMA-AODE is outperformed (p<0.01) by AODE regarding both accuracy and Brier loss. We present in Figure 3(a) the scatter plot of accuracies and in Figure 4(a) the *relative* Brier losses, namely the Brier loss of BMA-AODE divided, data set by data set, by the Brier loss of AODE. On average, BMA-AODE has 3% higher Brier loss than AODE. BMA-AODE computed with marginal likelihood was already found to be outperformed by AODE (Yang et al., 2007; Cerquides and de Mántaras, 2005).

As for the comparison of COMP-AODE with AODE: there is no significant difference on accuracy, as it can be inferred from Figure 3(b), but COMP-AODE outperforms AODE on the Brier loss (p-value $< .01$). Figure 4(b) shows the *relative* Brier

losses, namely the Brier loss of COMP-AODE divided, data set by data set, by the Brier loss of AODE. Averaging over data sets, COMP-AODE reduces the Brier loss of about 3% compared to AODE. We see this result as noteworthy, since AODE is a high performance classifier. These results broaden the scope of the experiments of (Boullé, 2007), in which the compression approach was applied to an ensemble of naive Bayes classifiers.

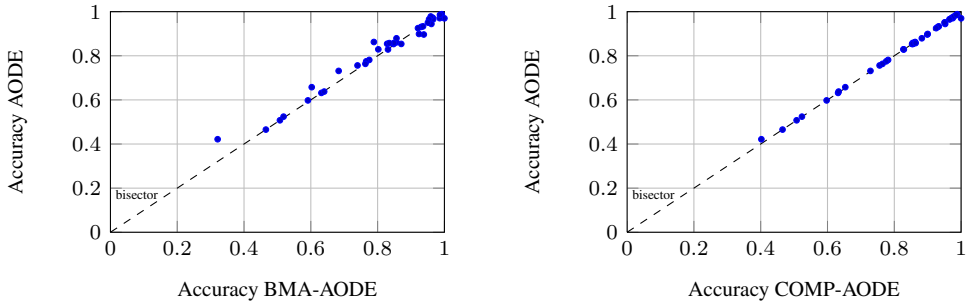


Figure 3: Scatter plots of accuracies; the solid line shows the bisector.

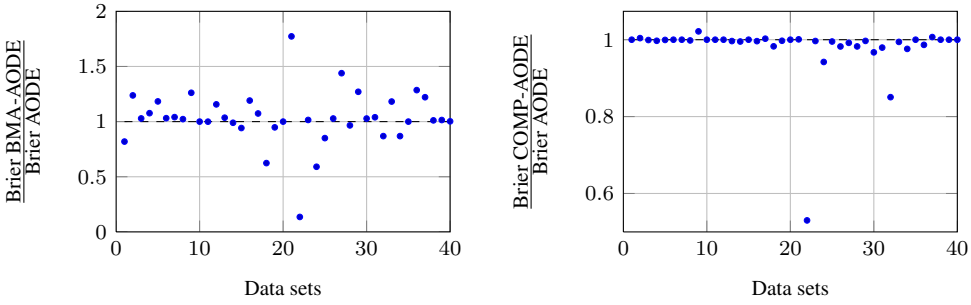


Figure 4: Relative Brier losses; points lying *below* the horizontal line represent performance better than AODE, and vice versa. Note the different y-scales of the two graphs.

3.2. Credal Classifiers

A credal classifier returns a set of classes if the instance is *prior-dependent* and a single class if the instance is instead *safe*, i.e. *non* prior-dependent. Note however that an instance can be judged as prior-dependent by a certain credal classifier and as safe by a different credal classifier. To characterize the performance of a credal classifier, the following four indicators are considered (Corani and Zaffalon, 2008b):

- *determinacy*: % of instances recognized as safe, namely classified with a single class;

- *single-accuracy*: the accuracy achieved over the instances recognized as safe;
- *set-accuracy*: the accuracy achieved, by returning a set of classes, over the prior-dependent instances;
- *indeterminate output size*: the average number of classes returned on the prior-dependent instances.

Averaging over data sets, BMA-AODE* has 94% determinacy; it is completely determinate on 7 data sets. The determinacy fluctuates among data sets, being however correlated with the sample size ($\rho = 0.3$). The choice of the prior is less important on large data sets: bigger data sets tend to contain a lower percentage of prior-dependent instances, thus increasing determinacy. BMA-AODE* performs well when indeterminate: averaging over all data sets, it achieves 90% set-accuracy by returning 2.3 classes (the average number of classes in the collection of data sets is 3.6). It is worth analyzing the performance of BMA-AODE on the prior-dependent instances. In Figure 5(a) we compare, data set by data set, the accuracy achieved by BMA-AODE on the instances judged respectively as safe and as prior-dependent by BMA-AODE*; the plot shows a sharp drop of accuracy on the prior-dependent instances, which is statistically significant (p-value $< .01$). As a rough indication, averaging over data sets, the accuracy of BMA-AODE is 83% on the safe instances but only 52% on the instances recognized as prior-dependent by BMA-AODE*. Thus, on the prior-dependent instances, BMA-AODE provides fragile classifications; on the same instances, BMA-AODE* returns a small-sized but highly accurate set of classes.

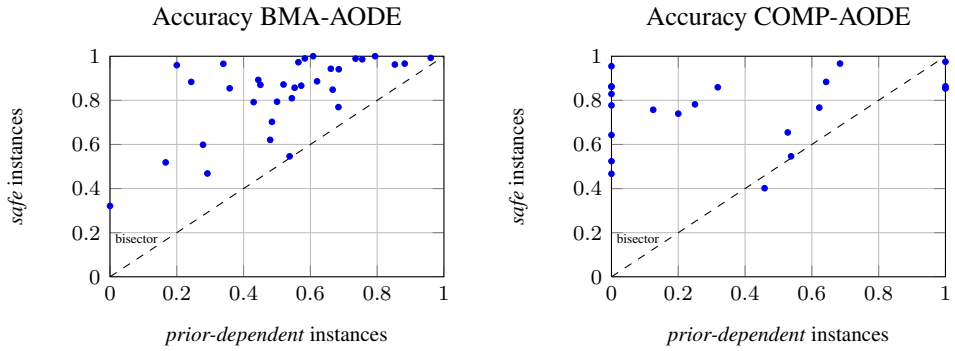


Figure 5: Accuracy of the determinate classifiers on the instances recognized as safe and as prior-dependent by their credal counterparts. The accuracies of BMA-AODE [COMP-AODE] is thus separately measured on the instances judged safe and prior-dependent by BMA-AODE* [COMP-AODE*]. The solid line shows the bisector.

Let us now analyze the performance of COMP-AODE*; it has higher determinacy than BMA-AODE*; averaging over data sets, its determinacy is 99%, with only minor fluctuations across data sets; the classifier is moreover completely determinate on 18 data sets. The determinacy of COMP-AODE* is very high and stable across data sets.

Therefore, under the compression-based approach only a small fraction of the instances is prior-dependent; this robustness to the choice of the prior is likely to contribute to the good performance of compression-based ensemble of classifiers and constitutes a desirable but previously unknown property of the compression-based approach. Numerical inspection shows that the logarithmic smoothing of the models' posterior probabilities makes indeed the compression weights only little sensitive to the choice of the prior. COMP-AODE* performs well when indeterminate: averaging over all data sets, it achieves 95% set-accuracy by returning 2 classes (note that the indeterminate output size cannot be less than two).

Again, it is worth checking the behavior of the corresponding determinate classifier, namely COMP-AODE, on the instances that are prior-dependent for the COMP-AODE*. In Figure 5(b) we compare, data set by data set, the accuracy achieved by COMP-AODE on the instances judged respectively safe and prior-dependent by COMP-AODE*; there is a large drop of accuracy on the prior-dependent instances, and the drop is significant (p-value $< .01$). Averaging over data sets, the accuracy of COMP-AODE drops from 82% on the safe instances to only 47% on the instances judged as prior-dependent by COMP-AODE*. Thus even COMP-AODE, despite its robustness and its high-performance, undergoes a severe loss of accuracy on the instances recognized as prior-dependent by COMP-AODE*. On the very same instances, COMP-AODE* returns a small sized but highly reliable set of classes, thus enhancing the overall classification reliability. These results convincingly show the importance of detecting the prior-dependent classifications.

3.3. Utility-based Measures

We have seen so far that the credal ensembles extend in a sensible way their determinate counterparts, being able to recognize prior-dependent instances and to robustly deal with them. Yet, it is not obvious how to compare credal and determinate classifiers by means of a synthetic indicator. In our view, the most principled answer to this question is that of Zaffalon et al. (2012), which allows comparing the 0-1 loss of a traditional classifier with a utility score defined for credal classifiers.

The starting point is the *discounted accuracy*, which rewards a prediction containing m classes with $1/m$ if it contains the true class, and with 0 otherwise. The discounted accuracy can be compared to the accuracy achieved by a determinate classifier. It has been shown (Zaffalon et al., 2012) that, within a betting framework based on fairly general assumptions, discounted-accuracy is the only score which satisfies some fundamental properties for assessing both determinate and indeterminate classifications. Yet Zaffalon et al. (2012) also shows some severe shortcomings of discounted-accuracy: consider two medical doctors, doctors *random* and doctor *vacuous*, who should diagnose whether a patient is *healthy* or *diseased*. Doctor *random* issues uniformly random diagnosis; doctor *vacuous* instead always returns both categories, thus admitting its ignorance. Let us assume that the hospital profits a quantity of money proportional to the discounted-accuracy achieved by its doctors at each visit. Both doctors have the same *expected* discounted-accuracy for each visit, namely $1/2$. For the hospital, both doctors provide the same *expected* profit on each visit, but with a substantial difference: the profit of doctor vacuous is *deterministic*, while the

profit of doctor random is affected by considerable variance. Any risk-averse hospital manager should thus prefer doctor vacuous over doctor random, since it yields the same expected profit with less variance. In fact, under risk-aversion, the expected utility increases with expectation of the rewards and decreases with their variance (Levy and Markowitz, 1979). To model this fact, it is necessary applying a utility function to the discounted-accuracy score assigned on each instance. Zaffalon et al. (2012) design the utility function as follows: the utility of a correct and determinate classification (discounted-accuracy 1) is 1; the utility of a wrong classification (discounted-accuracy 0) is 0. Therefore, the utility of a traditional determinate classifier is its accuracy. The utility of an accurate but indeterminate classification consisting of two classes (discounted-accuracy 0.5) is assumed to lie between 0.65 and 0.8. Two quadratic utility functions are then derived corresponding to these boundary values, and passing respectively through $\{u(0) = 0, u(0.5) = 0.65, u(1) = 1\}$ and $\{u(0) = 0, u(0.5) = 0.8, u(1) = 1\}$, denoted as u_{65} and u_{80} respectively; the mathematical expression of these utility functions are as follows: $u_{65}(x) = -1.2x^2 + 2.2x$, $u_{80}(x) = -0.6x^2 + 1.6x$, where x is the value of discounted accuracy. Since $u(1) = 1$, utility and accuracy coincide for determinate classifiers; therefore, utility of credal classifiers and accuracy of determinate classifiers can be directly compared. In Del Coz and Bahamonde (2009) classifiers which return indeterminate classifications are scored through the F_1-metric, originally designed for Information Retrieval tasks. The F_1 metric, when applied to indeterminate classifications, returns a score which is always comprised between u_{65} and u_{80}, further confirming the reasonableness of both utility functions. More details on the links between F_1, u_{65} and u_{80} are given in Zaffalon et al. (2012). While in real case studies the utility function should be elicited by discussing with the decision maker, u_{65} and u_{80} can be suitably used for data mining experiments like those shown in the following.

We now analyze the utilities generated by the various classifiers, comparing each credal classifier with its determinate counterpart; recall that for a traditional classifier, utility and accuracy are the same. The utility of BMA-AODE* is significantly higher (p-value $< .01$) than that of BMA-AODE under both u_{65} and u_{80}. This confirms that extending the model to imprecise probability is a sensible approach. In the first row of Figure 6 we show the *relative* utility, namely the utility of BMA-AODE* divided, data set by data set, by the utility (i.e., accuracy) of BMA-AODE; the two plots refer respectively to u_{65} and u_{80}. Averaging over data sets, the improvement of utility is about 1% and 2% under u_{65} and u_{80}; although the improvement might look small, we recall that it is obtained by modifying the classifications of the prior-dependent instances only, 6% of the total on average. If we focus on the prior-dependent instances only, the increase of utility generally varies between +10% and +40% depending on the data set and on the utility function. Clearly, the improvement is even larger under u_{80} which assigns higher utility than u_{65} to the indeterminate but accurate classifications.

The analysis is similar when comparing COMP-AODE* with COMP-AODE. In the second row of Figure 6 we show the *relative* utility, namely the utility of COMP-AODE* divided, data set by data set, by the utility (i.e., accuracy) of COMP-AODE. The increase of utility is in this case generally under 1%, as a consequence of the higher determinacy of COMP-AODE (99% on average), which allows less room for improving utility through indeterminate classifications. In fact, the robustness of COMP-

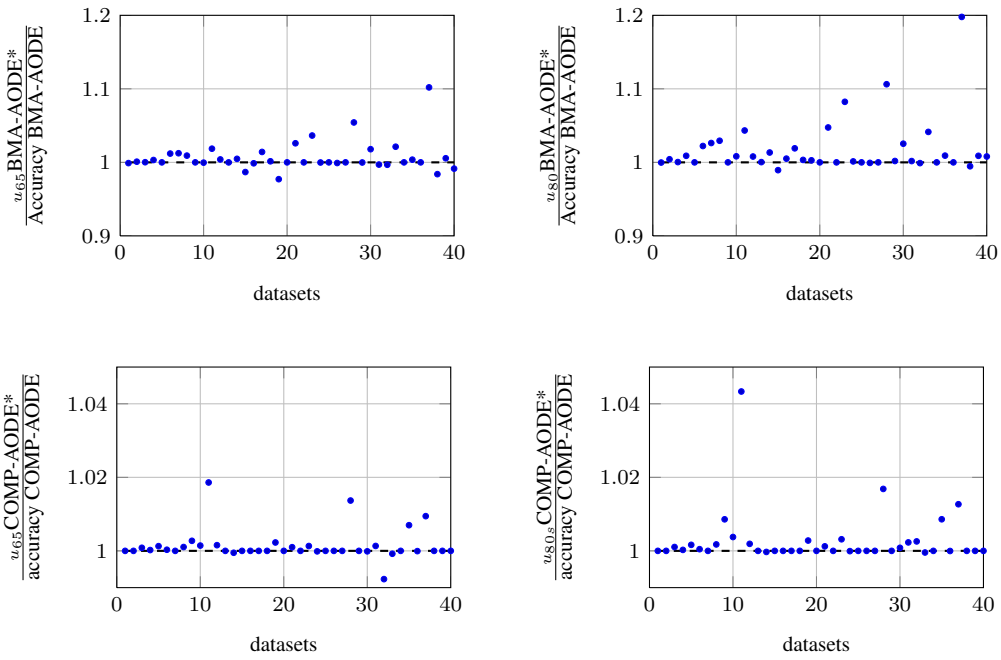


Figure 6: Relative utilities of credal classifiers compared to their precise counterparts.

AODE to the choice of the prior reduces the portion of instances where it is necessary making the classification indeterminate. Focusing however on the (rare) indeterminate instances, the increase of utility deriving to the extension to imprecise probability lies between 39% and 60%, depending on the data set and on the utility function. Eventually, COMP-AODE* has significantly (p-value $< .01$) higher utility than COMP-AODE under *both* u_{65} and u_{80}; also in this case the credal ensemble outperforms its traditional counterpart.

The utilities of COMP-AODE* and BMA-AODE* are also compared; under u_{65} COMP-AODE* yields significantly (p-value $< .05$) higher utility than BMA-AODE*, while under u_{80} the difference among the two classifiers is not significant, although the utility generated by COMP-AODE* is generally slightly higher. The point is that BMA-AODE* is more often indeterminate than COMP-AODE*; under u_{80} the indeterminate but accurate classifications are rewarded more than under u_{65}, thus allowing BMA-AODE* to almost close the gap with COMP-AODE*. We conclude however that COMP-AODE* should be generally preferred over BMA-AODE*.

Eventually we point out that COMP-AODE* generates significantly (p-value $<$ .01) higher utility than AODE, under *both* u_{65} and u_{80}. The extension to imprecise probability has thus concretely improved the overall performance of the compression-based ensemble: recall that the determinate COMP-AODE yields better probability estimates but not better accuracy than AODE.

3.4. Comparison with Previous Credal Classifiers

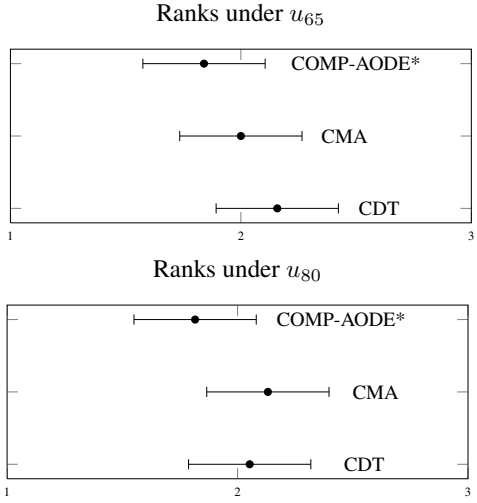


Figure 7: Comparison between credal classifiers by means of the Friedman test: the boldfaced points show the average ranks; a lower rank implies better performance. The bars display the critical distance, computed with 95% confidence: the performance of two classifiers are significantly different if their bars do not overlap.

In this section we compare COMP-AODE* with previous credal classifiers. A well-known credal classifier is the *naive credal classifier* (NCC) (Corani and Zaffalon, 2008b), which is an extension of naive Bayes to imprecise probability. The comparison of NCC and COMP-AODE* on the 40 data sets of Section 3 shows that COMP-AODE yields significantly (p <0.01) higher utility than NCC under *both* u_{65} and u_{80}.

However, over time algorithms more sophisticated than NCC have been developed, such as:

- *credal model averaging* (CMA) (Corani and Zaffalon, 2008a), namely a generalization of BMA (in the same spirit of BMA-AODE) for naive Bayes classifier;
- *credal decision tree* (CDT) (Abellán and Moral, 2005), namely an extension of classification trees to imprecise probability.

We then compare CDT, CMA and COMP-AODE* via the Friedman test; this is the approach recommended by (Demšar, 2006) for comparing multiple classifiers on multiple data sets. First, the procedure ranks on each data set the classifiers according to the utility they generate; then, it tests the null hypothesis of all classifiers having the same average rank across the data sets. If the null hypothesis is rejected, a post-hoc test is adopted to identify the significant differences among classifiers. Adopting a 95% confidence, no significant difference is detected among classifiers; the result is the same under both utilities. However, under both utilities COMP-AODE* has the best average rank, as shown in Figure 3.4. Lowering the confidence to 90%, two significant differences are found: a) COMP-AODE* produces significantly higher utility than CMA

under u_{65} and b) COMP-AODE* produces significantly higher utility than CDT under u_{80}. These results, though not completely conclusive, suggest that COMP-AODE* compares favorably to previous credal classifiers.

3.5. Some Comments on Credal Classification versus Reject Option

Determinate classifiers can be equipped with a *reject option* (Herbei and Wegkamp, 2006), thus refusing to classify an instance if the posterior probability of the most probable class is less than a threshold. For the sake of simplicity we consider a case with two classes only; to formally introduce the reject option, it is necessary setting a cost d ($0 < d < 1/2$), which is incurred into when rejecting an instance. A cost 0, 1, d is therefore incurred into when respectively correctly classifying, wrongly classifying and rejecting an instance. Under 0-1 loss, the *expected* cost for classifying an instance corresponds to the probability of misclassification; it is thus $1 - p^*$, where p^* denotes the posterior probability of the most probable class. The optimal behavior is thus to reject the classification whenever the expected classification cost is higher than the rejection cost, namely when $(1 - p^*) > d$; this is equivalent to rejecting the instance whenever $p^* < 1 - d$, where $(1 - d)$ constitutes the *rejection threshold*.

The behavior induced by the reject option is quite different from that of a credal classifier, as we show in the following example. On an a very large data set the posterior probability of the classes is little sensitive on the choice of the prior, because of the wide amount of data available for learning; in this condition, instance are rarely prior-dependent and therefore a credal classifier will mostly return a single class. On the other hand, the determinate classifier with reject option (RO in the following) rejects all the instances for which $p^* < 1 - d$; if d is small, there can be even a high number of rejected instances. The difference between these behaviors is due to the credal classifier being unaware of the cost d associated with rejecting an instance, which is instead driving the behavior of RO. To rigorously compare RO against a credal classifier, it is thus necessary making the credal classifier aware of the cost d. Recalling that the credal classifier already returns both classes on the instances which are prior dependent, this will change the behavior of the credal classifier only on the instances which are *not* prior-dependent. In particular, the credal classifier should reject all the instances for which $\underline{p}^* < 1 - d$, where $\underline{p}^*$ is the *lower* probability of the most probable class; the instances rejected by means of this criterion will be thus a superset of those rejected by RO. Therefore, the credal classifier will reject the instances which are prior-dependent *and* those for which $\underline{p}^* < 1 - d$. Eventually, the cost generated by the credal classifier should be compared with those generated by the RO. In the case with more than 2 classes the analysis might become slightly more complicated than what discussed here; however, we leave the analysis of credal classifiers with reject option as a topic for future research. Note also that this kind of experiment will require the computation of upper and lower posterior probability of the classes, which is not always trivial with credal classifiers.

4. Conclusions

This main contribution of this paper regards two novel credal ensembles of SPODE classifiers. Both ensembles compare favorably to more traditional ensemble, showing

the effectiveness of the IP approach to deal with the problem of specifying the prior over the models. In particular, the credal ensembles automatically identify the prior-dependent instances and cope reliably with them by returning a small-sized but highly accurate set of classes. On the same instances, the traditional ensembles of classifiers undergo instead a severe drop of accuracy. In particular, the credal ensemble based on compression weights achieves very good performance; our findings thus broaden the application scope of compression weights, originally introduced by Boullé (2007).

As a future work, it could be interesting developing a credal generalization of the MAPLMG algorithm (Cerquides and de Mántaras, 2005) which is so far the highest-performing approach (Yang et al., 2007) for aggregating SPODEs. Other interesting developments might include applying the ideas of credal ensemble to domains other than classification, such as regression. A credal ensemble of different regressor models would then return predictions which are interval rather than points; the interval would highlight the sensitivity of the prediction on the prior which is set over the competing regression models.

In terms of computational speed, it would be possible making BMA-AODE* faster by solving its credal-dominance test without running the linear programming procedure, identifying instead the solution on the basis of the derivatives of the objective functions; a similar approach has been for instance followed by (Corani and Zaffalon, 2008a).

Acknowledgements

The research in this paper has been partially supported by the Swiss NSF grants no. 200020-132252 and by the Hasler foundation grant n. 10030. We thank the anonymous reviewers for their valuable insights.

References

Abellán, J., Moral, S., 2005. Upper entropy of credal sets. Applications to credal classification. International Journal of Approximate Reasoning 39, 235–255.

Borgwardt, K., 1987. The simplex algorithm: a probabilistic analysis. Springer.

Boullé, M., 2007. Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research 8, 1659–1685.

Cerquides, J., de Mántaras, R., 2005. Robust Bayesian linear classifier ensembles, in: Proc. of the 16th European Conference on Machine Learning (ECML-PKDD 2005), pp. 72–83.

Corani, G., Antonucci, A., Zaffalon, M., 2012. Bayesian networks with imprecise probabilities: theory and application to classification, in: Holmes, D.E., Jain, L.C., Kacprzyk, J., Jain, L.C. (Eds.), Data Mining: Foundations and Intelligent Paradigms. volume 23, pp. 49–93.

Corani, G., Zaffalon, M., 2008a. Credal model averaging: an extension of Bayesian model averaging to imprecise probabilities, in: Proc. of the European Conference on Machine Learning (ECML-PKDD 2008), pp. 257–271.

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

Cowell, R., 2001. On searching for optimal classifiers among Bayesian networks, in: Proc. of the 8th International Conference on Artificial Intelligence and Statistics (AISTATS 2001), pp. 175–180.

Del Coz, J., Bahamonde, A., 2009. Learning nondeterministic classifiers. Journal of Machine Learning Research 10, 2273–2293.

Dash, D., Cooper, G., 2004. Model Averaging for Prediction with Discrete Bayesian Networks. Journal of Machine Learning Research 5, 1177–1203.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Domingos, P., 2000. Bayesian averaging of classifiers and the overfitting problem, in: Proc. of the 17th International Conference on Machine Learning (ICML 2000), pp. 223–230.

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

Dubois, D., Prade, H., 1997. A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. International Journal of Approximate Reasoning 17, 295–324.

Fayyad, U.M., Irani, K.B., 1993. Multi-interval discretization of continuous-valued attributes for classification learning, in: Proc. of the 13th international joint conference on artificial intelligence (IJCAI-93), pp. 1022–1027.

Friedman, J., 1997. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery 1, 55–77.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian networks classifiers. Machine Learning 29, 131–163.

Grünwald, P., 2005. A tutorial introduction to the minimum description length principle, in: Grünwald, P., Myung, I.J., Pitt, M. (Eds.), Advances in Minimum Description Length: Theory and Applications, MIT Press. pp. 3–81.

Heckerman, D., 1995. A tutorial on learning with Bayesian networks, in: Jordan, M. (Ed.), Learning in Graphical Models, MIT Press.

Herbei, R., Wegkamp, M., 2006. Classification with reject option. Canadian Journal of Statistics 34, 709–721.

Hoeting, J., Madigan, D., Raftery, A., Volinsky, C., 1999. Bayesian Model Averaging: a Tutorial. Statistical Science 14, 382–417.

Karmakar, N., 1984. A new polynomial-time algorithm for linear programming. Combinatorica 4, 373–395.

Kontkanen, P., Myllymaki, P., Silander, T., Tirri, H., 1999. On supervised selection of Bayesian networks, in: Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (AISTATS 1999), pp. 334–342.

Levy, H., Markowitz, H., 1979. Approximating expected utility by a function of mean and variance. The American Economic Review 69, 308–317.

Minka, T., 2002. Bayesian model averaging is not model combination. Technical Report. MIT Media Lab.

Piatti, A., Zaffalon, M., Trojani, F., Hutter, M., 2009. Limits of learning about a categorical latent variable under prior near-ignorance. International Journal of Approximate Reasoning 50, 597–611.

Troffaes, M., 2007. Decision making under uncertainty using imprecise probabilities. International Journal of Approximate Reasoning 45, 17–29.

Walley, P., 1991. Statistical reasoning with imprecise probabilities. Chapman and Hall, New York.

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

Webb, G., Boughton, J., Wang, Z., 2005. Not so naive Bayes: Aggregating one-dependence estimators. Machine Learning 58, 5–24.

Yang, Y., Webb, G., Cerquides, J., Korb, K., Boughton, J., Ting, K., 2007. To select or to weigh: A comparative study of linear combination schemes for superparent-one-dependence estimators. IEEE Trans. on Knowledge and Data Engineering 19, 1652–1665.

Zaffalon, M., Corani, G., Maua, D., 2012. Evaluating credal classifiers by utility-discounted predictive accuracy. International Journal of Approximate Reasoning 53 (8),1282–1301.

Appendix A. Data sets list

Table A.2: List of the 40 data sets used for experiments.

dataset	n	k	classes	dataset	n	k	classes
labor	57	11	2	ecoli	336	6	8
white_clover	63	6	4	liver_disorders	345	1	2
postoperative	90	8	3	ionosphere	351	33	2
zoo	101	16	7	monks3	554	6	2
lymph	148	18	4	monks1	556	6	2
iris	150	4	3	monks2	601	6	2
tae	151	2	3	credit_a	690	15	2
grub_damage	155	6	4	breast_w	699	9	2
hepatitis	155	16	2	diabetes	768	6	2
hayes_roth	160	3	3	anneal	898	31	6
wine	178	13	3	credit_g	1000	15	2
sonar	208	21	2	cmc	1473	9	3
glass	214	7	7	yeast	1484	7	10
heart_h	294	9	2	segment	2310	18	7
heart_c	303	11	2	kr_vs_kp	3196	36	2
haberman	306	2	2	hypothyroid	3772	25	4
solarflare_C	323	10	3	waveform	5000	19	3
solarflare_M	323	10	4	page_blocks	5473	10	5
solarflare_X	323	10	2	pendigits	10992	16	10
ecoli	336	6	8	nursery	12960	8	5