Rough Set-based Feature Selection for Weakly Labeled Data

Andrea Campagner[a], Davide Ciucci[a], Eyke Hüllermeier[b]

[a]Department of Informatics, Systems and Communication
University of Milano–Bicocca, viale Sarca 336 – 20126 Milano, Italy
[b] Institute of Informatics, University of Munich (LMU), Germany

Abstract

Supervised learning is an important branch of machine learning (ML), which requires a complete annotation (labeling) of the involved training data. This assumption is relaxed in the settings of *weakly* supervised learning, where labels are allowed to be imprecise or partial. In this article, we study the setting of superset learning, in which instances are assumed to be labeled with a set of *possible* annotations containing the correct one. We tackle the problem of learning from such data in the context of *rough set theory* (RST). More specifically, we consider the problem of RST-based feature reduction as a suitable means for *data disambiguation*, i.e., for the purpose of figuring out the most plausible precise instantiation of the imprecise training data. To this end, we define appropriate generalizations of decision tables and reducts, using tools from generalized information theory and belief function theory. Moreover, we analyze the computational complexity and theoretical properties of the associated computational problems. Finally, we present results of a series of experiments, in which we analyze the proposed concepts empirically and compare our methods with a state-of-the-art dimensionality reduction algorithm, reporting a statistically significant improvement in predictive accuracy.

Keywords: Superset Learning, Rough Sets, Feature Selection, Evidence Theory, Entropy.

1. Introduction

Weakly supervised learning [69] refers to machine learning tasks in which training instances are not required to be associated with a precise target label. Instead, the annotations can be imprecise or partial. Such tasks could be the consequence of certain data pre-processing operations such as anonymization [15, 49] or censoring [17], could be due to imprecise measurements or expert opinions, or meant to limit data annotation costs [45]. Some examples of weakly supervised learning tasks include semi-supervised learning, but also more general tasks like learning from soft labels [8, 12, 13, 48], (in which partial labels are represented through belief functions) which, in turn, encompasses both learning from fuzzy labels [14, 28] (in which partial labels are represented through possibility distributions) and superset learning [29, 40, 44]. In this latter setting, which will be the focus of this article, each instance x is annotated with a set S of candidate labels that are deemed (equally) *possible*. In other words, we know that the label of x is an element of S, but nothing more. For example, an image could be tagged with $\{\text{horse}, \text{pony}, \text{zebra}\}$, suggesting that the animal shown on the picture is one of these three, though it is not exactly known which of them.

In the recent years, the superset learning task has been widely investigated both under the classification perspective [19, 30, 64, 66] and from a theoretical standpoint [39]. The latter result is particularly relevant, as it shows that, as in the standard PAC learning model, superset learnability is characterized by combinatorial dimensions (e.g., Vapnik-Chervonenkis or Natarajan dimension) which, in general, depend on the dimensionality (i.e., the number of features) of the learning problem. Thus, the availability of effective *feature selection* [24] or dimensionality reduction algorithms would be of critical importance in order to control model capacity and, hence, ensure proper model generalization. Nevertheless, this task has not received much attention so far [61].

In this article, which is an extension of our previous article [6], we study the application of *rough set theory* in the setting of superset learning. In particular, adhering to the generalized risk minimization principle [28], we consider the problem of feature reduction as a mean for *data disambiguation*, i.e., for the purpose of figuring out the most plausible precise instantiation of the imprecise training data. Compared to our previous work, we provide a finer characterization of the theoretical properties and relations among the proposed definitions of reduct through Theorems 3.4, 3.5, 3.7 that were previously left as open problems. In Section 4, which has been newly added, we also discuss two computational experiments by which we study the empirical performance of the proposed reduct definitions, also in comparison with the state-of-the-art method for dimensionality reduction in superset learning.

2. Background

In this section, we recall basic notions of rough set theory (RST) and belief function theory, which will be used in the main part of the article.

2.1. Rough Set Theory

Rough set theory has been proposed by Pawlak [46] as a framework for representing and managing uncertain data, and has since been widely applied for various problems in the ML domain (see [4] for a recent overview and survey). We briefly recall the main notions of RST, especially regarding its applications to feature reduction.

A decision table (DT) is a triple $DT = \langle U, Att, t \rangle$ such that U is a universe of objects and Att is a set of *attributes* employed to represent objects in U. Formally, each attribute $a \in Att$ is a function $a : U \to V_a$, where V_a is the domain of values of a. Moreover, $t \notin Att$ is a distinguished *decision* attribute, which represents the target decision (also labeling or annotation) associated with each object in the universe. We say that DT is *inconsistent* if the following holds: $\exists x_1, x_2 \in U, \forall a \in Att, a(x_1) = a(x_2)$ and $t(x_1) \neq t(x_2)$.

Given $B \subseteq Att$, we can define the *indiscernibility relation* with respect to B as xI_Bx' iff $\forall a \in B$, $a(x') = a(x)$. Clearly, it is an equivalence relation that partitions the universe U in equivalence classes, also called *granules of information*, $[x]_B$. Then, the *indiscernibility partition* is denoted as $\pi_B = \{[x]_B \mid x \in U\}$.

We say that $B \subseteq Att$ is a *decision reduct* for DT if $\pi_B \leq \pi_t$ (where the order $\leq$ is the refinement order for partitions, that is, π_t is a coarsening of π_B) and there is no $C \subsetneq B$ such that $\pi_C \leq \pi_t$. Then, evidently, a reduct of a decision table DT represents a set of non-redundant and necessary features to represent the information in DT. We say that a reduct R is *minimal* if it is among the smallest (with respect to cardinality) reducts.

Given $B \subseteq Att$ and a set $S \subseteq U$, a *rough approximation* of S (with respect to B) is defined as the pair $B(S) = \langle l_B(S), u_B(S)\rangle$, where $l_B(S) = \bigcup\{[x]_B \mid [x]_B \subseteq S\}$ is the *lower approximation* of S, and $u_B(S) = \bigcup\{[x]_B \mid [x]_B \cap S \neq \emptyset\}$ is the corresponding *upper approximation*.

Finally, given $B \subseteq Att$, the *generalized decision* with respect to B for an object $x \in U$ is defined as $\delta_B(x) = \{t(x') \mid x' \in [x]_B\}$. Notably, if DT is consistent and B is a reduct, then $\delta_B(x) = \{t(x)\}$ for all $x \in U$.

We notice that in the RST literature, there exist several definitions of reduct that, while equivalent on consistent DTs, are generally non-equivalent for inconsistent ones. We refer the reader to [55] for an overview of such a list and a study of their dependencies, while here we report two specific definitions that are useful for the following:

Definition 2.1. *$B \subset Att$ is a δ-reduct if $\forall x \in U, \;\; \delta_B(x) = \delta_{Att}(x)$.*

Definition 2.2. *$B \subset Att$ is μ-reduct if $\forall x \in U, \forall v \in V_t, \;\; Pr(v|[x]_B) = Pr(v|[x]_{Att})$, where*

$$Pr(v|[x]_B) = \frac{|\{x' \in [x]_B : t(x') = v\}|}{|[x]_B|}.$$

Further, we recall the following result:

Theorem 2.1. *Let DT be a decision table. Then, every μ-reduct of DT is also a δ-reduct of DT, but not vice versa.*

We further notice that, given a decision table, the problem of finding the minimal reduct is in general NP-hard (by reduction to the *Shortest Implicant* problem [53, 59]).

2.2. Belief Function Theory

Belief Function Theory (BFT), also known as Dempster-Shafer theory (DST) or Evidence theory (ET), has originally been introduced by Dempster in [10] and subsequently formalized by Shafer in [50] as a generalization of probability theory (although this interpretation has been disputed [47]). The starting point is a *frame of discernment* X, which represents all possible states of a system under study, together with a *basic belief assignment* (bba) $m : 2^X \to [0,1]$, such that $m(\emptyset) = 0$ and $\sum_{A \in 2^X} m(A) = 1$. From this bba, a pair of functions, called respectively *belief* and *plausibility*, can be defined as follows:

$$Bel_m(A) = \sum_{B : B \subseteq A} m(B) \tag{1}$$

$$Pl_m(A) = \sum_{B : B \cap A \neq \emptyset} m(B) \tag{2}$$

As can be seen from these definitions, there is a clear correspondence between belief functions (resp., plausibility functions) and lower approximations (resp., upper approximations) in RST: this connection has been first established in [63], in which the authors showed that every belief function can be derived from a corresponding (generalized) decision table. More recently, the connection between BFT and RST have been investigated from both the theoretical point of view, for example in [68], where the authors provide a characterization of belief functions in terms of lower and upper approximation operators, and in [65], where the author discuss a novel approach to decision-theoretic rough sets based on BFT; and also from the application point of view: in [67] the authors propose an algorithm for feature reduction based on BFT in the setting of Pythagorean fuzzy rough approximation spaces; while in [7] the authors propose an algorithm to induce weighted decision rules based on RST and BFT.

Starting from a bba, a probability distribution, called *pignistic probability*, can be obtained [57]:

$$P^m_{Bet}(x) = \sum_{A : x \in A} \frac{m(A)}{|A|} \tag{3}$$

Finally, we recall that appropriate generalizations of information-theoretic concepts [51], specifically the concept of *entropy* (which was also proposed to generalize the definition of reducts in RST [54]), have been defined for evidence theory. These include measures of non-specificity [1, 16], measures of conflict or dissonance [26, 35, 56, 62], and measures of total uncertainty [2, 25, 34]: see [34] for a comprehensive review on generalizations of entropy for evidence theory. Most relevantly for the purposes of this article, we recall the definition of *aggregate uncertainty* [25]:

$$AU(m) = \max_{p \in \mathcal{P}(m)} H_p(X)\,, \tag{4}$$

where $\mathcal{P}(m)$ is the set of probability distributions p such that $Bel_m \leq p \leq Pl_m$, and $H_p(X) = -\sum_{x \in X} p(x) log_2 p(x)$ the Shannon entropy of p. While this measure is not compatible with Dempster combination rule (see [34]; note, however, that we do not rely on Dempster combination rule in this paper), it complies with the generalized risk minimization approach [28] to superset learning and, more in particular, with the pessimistic loss approach to generalized risk minimization [22, 23, 31]. Another relevant approach is the *normalized pignistic entropy* (see [36] for the non-normalized definition)

$$H_{Bet}(m) = \frac{H(P^m_{Bet})}{H(\hat{p}_m)} \,, \tag{5}$$

where $\hat{p}_m$ is the probability distribution that is uniform on the support of $P^m_{Bet}(x)$, i.e., on the set of elements $\{x \,|\, P^m_{Bet}(x) > 0\}$. Similarly to the AU, also the pignistic entropy is not compatible with Dempster combination rule, but has the advantage of being efficiently computable.

2.3. Superset Learning

As already mentioned in the introduction, *superset learning* is a specific type of *weakly supervised learning* and, more precisely, a specific type of the *learning from soft labels* [8, 11, 13, 48] task. While in learning from soft labels the partial labels are represented through general belief functions [11], in the case of superset learning each instance (or object) $x \in U$, where U is a data set (e.g., the training data in a machine learning setting), is annotated with a collection of labels $S \subseteq \mathcal{Y}$ (i.e., in BFT terminology, the partial labels are represented by belief functions with a single focal set). The common interpretation of S is in terms of a set of candidates of an underlying ground-truth: There is a true label y, which is not precisely known, but which is known to be an element of S. In other words, S is a superset of y, hence the name "superset learning".

As an illustration, consider the famous Iris data, where the objects are iris plants characterized by four attributes $a_1, \ldots, a_4$ (sepal length, sepal width, petal length, petal width). Moreover, each plant belongs to either of the three categories Setosa, Versicolor, Virginica. Thus, a labeled instance in a data set might be given by $(6.1, 2.9, 4.7, 1.4, \text{Versicolor})$. Now, imagine that a botanist who is responsible for the categorization is not entirely certain about the type of a plant with features $x = (6.1, 2.9, 4.7, 1.4)$, but can at least exclude Setosa as an option. She could then label the instance with $S = \{\text{Versicolor}, \text{Virginica}\}$.

In spite of the ambiguous, set-valued training data, the goal that is commonly considered in superset learning is to induce a unique model, i.e., a map $h : \mathcal{X} \rightarrow \mathcal{Y}$ that generalizes beyond the training data and can be used to make predictions $h(x) \in \mathcal{Y}$ for any new query instance $x \in \mathcal{X}$. In one way or the other, this requires the "disambiguation" of the training data. To this end, various methods and algorithmic approaches have been proposed in the literature, for example based on maximum likelihood estimation [8, 13, 33, 40, 48], generalizations of empirical risk minimization [28, 30, 31], convex optimization [9, 18], and instance-based approaches [11, 29, 66]. In [39], superset learning has been studied from a theoretical perspective in the framework of PAC learning.

Superset learning has mostly been studied for classification problems so far, while other (related) machine learning tasks have been considered much less. This also includes feature selection, despite its important influence on model complexity, generalization performance, and transparency of learning algorithms. Indeed, while many works have studied feature selection and dimensionality reduction in the setting of semi-supervised learning [3, 52], which is actually a special case of superset learning, to the knowledge of the authors, the only work focusing on the more general setting of superset learning is the DELIN algorithm proposed in [61]. Compared to the method put forward in this article, we note two main differences. First, being based on Linear Discriminant Analysis (LDA), DELIN relies on specific assumptions regarding the statistical distribution of the data, whereas our method (based on Rough Set Theory) is completely non-parametric. Second, DELIN is a dimensionality reduction algorithm, which means that it constructs a new set of attributes that is not (in general) a subset of the original one. By contrast, our approach is a feature selection algorithm, which selects a subset of the original set of attributes. In Section 4, we will provide an experimental comparison of the two methods.

As for the notation and connection to RST, it should be clear that the attribute y and its domain $\mathcal{Y}$ in superset learning play the role, respectively, of the decision attribute t and its domain V_t in RST. As an aside, let us note that the information provided in superset learning may also be interpreted in a different way, which provides an alternative motivation for the superset extension of decision tables in general and the search for reducts of such tables in particular. As explained above, the superset extension is mostly motivated by the assumption of imprecise labeling: The value of the decision attribute is not known precisely but only characterized in terms of a set of possible candidates. As will be seen further below, finding a reduct is then supposed to help disambiguate the data, i.e., figuring out the most plausible among the candidates. Instead of this "don't know" interpretation, a superset S can also be given a "don't care" interpretation: In a certain context characterized by x, all decisions in S are sufficiently good, or "satisficing" in the sense of March and Simon [42]. A reduct can then be considered as a maximally simple (least cognitively demanding) yet satisficing decision rule.

3. Superset Decision Tables and Reducts

In this section, we extend some key concepts of rough set theory to the setting of superset learning.

3.1. Superset Decision Tables

In superset learning, an object $x \in U$ is not necessarily assigned a single annotation $t(x) \in V_t$, but instead a set S of candidate annotations, one of which is assumed to be the true annotation associated with x. To model this idea in terms of RST, we generalize the definition of a decision table as follows.

Definition 3.1. *A* superset decision table *(SDT) is a tuple $SDT = \langle U, Att, t, d\rangle$, where $\langle U, Att, t\rangle$ is a decision table, i.e.:*

- *U is a universe of objects of interest;*

- *Att is a set of attributes (or features);*
- *t is the (real) decision attribute (whose value, in general, is not known);*
- *$d \notin Att$ is a candidate decision attribute, that is, a set-valued map $d : U \to \mathcal{P}(V_t)$ such that the* superset property *holds: $t(x) \in d(x)$ for all $x \in U$.*

The intuitive meaning of the set-valued information d is that, if $|d(x)| > 1$ for some $x \in U$, then the real decision associated with x (i.e., $t(x)$) is not known precisely, but is known to be in $d(x)$. Notice that Definition 3.1 is a proper generalization of decision tables: if $|d(x)| = 1$ for all $x \in U$, then we have a standard decision table.

Remark 3.1. *In Definition 3.1, a set-valued decision attribute is modelled as a function $d : U \to \mathcal{P}(V_t)$. While this mapping is formally well-defined for a concrete decision table, let us mention that, strictly speaking, there is no functional dependency between x and $d(x)$. In fact, $d(x)$ is not considered as a property of x, but rather represents* information *about a property of x, namely the underlying decision attribute $t(x)$. As such, it reflects the epistemic state of the decision maker.*

A SDT can be associated with a collection of compatible (standard) decision tables, which we call instantiations of the SDT.

Definition 3.2. *An* instantiation *of a SDT $\langle U, Att, t, d\rangle$ is a standard DT $I = \langle U, Att, t_I\rangle$ such that $t_I(x) \in d(x)$ for all $x \in U$. The set of instantiations of SDT is denoted $\mathcal{I}(SDT)$.*

The notion of inconsistency of a SDT has to reflect this richness. The following definition reflects the idea that no instantiations are consistent.

Definition 3.3. *For $B \subset Att$, the SDT is B-inconsistent if*

$$\exists x_1, x_2 \in U, \forall a \in B, a(x_1) = a(x_2) \text{ and } d(x_1) \cap d(x_2) = \emptyset\,. \tag{6}$$

We call such a pair x_1, x_2 inconsistent. If condition (6) is not satisfied, the SDT is B-consistent.

Thus, inconsistency implies the existence of (at least) two indiscernible objects with non-overlapping superset decisions. We say that an instantiation I is *consistent with a SDT S* (short, is consistent) if the following holds for all x_1, x_2: if x_1, x_2 are consistent in S, then they are also consistent in I.

3.2. Superset Reducts

Learning from superset data is closely connected to the idea of *data disambiguation* in the sense of figuring out the most plausible instantiation of the set-valued training data [27, 31]. But what makes one instantiation more plausible than another one? One approach originally proposed in [29] refers to the principle of simplicity in the spirit of *Occam's razor* (which can be given a theoretical justification in terms of *Kolmogorov complexity* [38]): An instantiation that can be explained by a simple model is more plausible than an instantiation that requires a complex model. In the context of RST-based data analysis, a natural measure of model complexity is the size of the reduct. This leads us to the following definition.

Definition 3.4. *A set of attributes $R \subseteq Att$ is a (consistent)* superset reduct *if there exists a (consistent) instantiation $I = \langle U, Att, t_I \rangle$ such that R is a reduct for I and there is no other (consistent) instantiation $I' = \langle U, Att, t_{I'} \rangle$ with reduct $R' \subset R$. We denote with R_{super} (resp., R^c_{super}) the set of superset reducts (resp., consistent superset reducts). A* minimum description length (MDL) instantiation *is one of the (consistent) instantiations of SDT that admits a reduct of minimum size compared to all the reducts of all possibile (consistent) instantiations. We will call the corresponding reducts* MDL reducts.

First of all, in order to clarify these concepts, we show a brief example.

Example 3.1. *Consider the superset decision table*

$$SDT = \Big\langle U = \{x_1, ..., x_6\}, A = \{a_1, a_2, a_3, a_4\}, d \Big\rangle$$

given in Table 1.

Table 1: An example of superset decision table

	a_1	a_2	a_3	a_4	d
x_1	0	0	0	0	0
x_2	0	0	0	1	$\{0,1\}$
x_3	0	1	1	0	0
x_4	0	1	1	1	$\{0,1\}$
x_5	0	1	0	1	1
x_6	0	1	0	0	$\{0,1\}$

It is easy to observe that the SDT admits 8 possible instantiations:

- I_1 *s.t.* $t_{I_1}(x_1) = t_{I_1}(x_2) = t_{I_1}(x_3) = t_{I_1}(x_4) = t_{I_1}(x_6) = 0$ *and* $t_{I_1}(x_5) = 1$*;*
- I_2 *s.t.* $t_{I_2}(x_1) = t_{I_2}(x_2) = t_{I_2}(x_3) = t_{I_2}(x_4) = 0$ *and* $t_{I_2}(x_5) = t_{I_2}(x_6) = 1$*;*
- I_3 *s.t.* $t_{I_3}(x_1) = t_{I_3}(x_2) = t_{I_3}(x_3) = t_{I_3}(x_6) = 0$ *and* $t_{I_3}(x_5) = t_{I_3}(x_4) = 1$*;*
- I_4 *s.t.* $t_{I_4}(x_1) = t_{I_4}(x_3) = t_{I_4}(x_4) = t_{I_4}(x_6) = 0$ *and* $t_{I_4}(x_5) = t_{I_4}(x_2) = 1$*;*
- I_5 *s.t.* $t_{I_5}(x_1) = t_{I_5}(x_2) = t_{I_5}(x_3) = 0$ *and* $t_{I_5}(x_4) = t_{I_5}(x_5) = t_{I_5}(x_6) = 1$*;*
- I_6 *s.t.* $t_{I_6}(x_1) = t_{I_6}(x_3) = t_{I_6}(x_4) = 0$ *and* $t_{I_6}(x_2) = t_{I_6}(x_5) = t_{I_6}(x_6) = 1$*;*

- *I_7 s.t. $t_{I_7}(x_1) = t_{I_7}(x_3) = t_{I_7}(x_6) = 0$ and $t_{I_7}(x_2) = t_{I_7}(x_4) = t_{I_7}(x_5) = 1$;*
- *I_8 s.t. $t_{I_8}(x_1) = t_{I_8}(x_3) = 0$ and $t_{I_8}(x_2) = t_{I_8}(x_4) = t_{I_8}(x_5) = t_{I_8}(x_6) = 1$;*

All of the instantiations are Att-consistent, since no two $x, x' \in U$ are associated with the same representation. It is easy to observe that the single shortest reduct among all instantiations is $R = \{a_4\}$, with corresponding instantiation I_7: thus I_7 is a MDL instantiation and $\{a_4\}$ is the unique MDL reduct (and thus also a superset reduct). The SDT also admits another superset reduct, namely $\{a_2, a_3\}$ (with corresponding instantiation I_2).

Then, we briefly comment on the fact that the definition of MDL reduct generalizes the standard definition of (minimal) reduct. Indeed, in a classical decision table, there is only one possible instantiation, hence the MDL reduct is exactly (one of) the minimal reducts of the decision table. Further, if we denote by R_{MDL} the set of MDL reducts, and by R^c_{MDL} the set of consistent MDL reducts (i.e., the MDL reducts corresponding only to consistent instantiations), then we can prove the following result:

Theorem 3.1. *$R_{MDL} \subseteq R_{super}$ and $R^c_{MDL} \subseteq R^c_{super}$. Furthermore, if $R \in R^c_{MDL}$ (resp., R^c_{super}), then $\exists R' \in R_{MDL}$ (resp., R_{super}) s.t. $R' \subseteq R$.*

Proof. If R is a consistent MDL reduct, then by definition it is also a consistent superset reduct, thus $R^c_{MDL} \subsetneq R^c_{super}$. The same holds for R_{MDL}, R_{super}.

As regard the second pair of statement, it is obviously the case that if we consider also inconsistent instantiations then the set of superset super-reducts (denoted with SR_{super}) contains the set of superset super-reducts that we would obtain were we to consider only consistent instantiations (denoted SR^c_{super}): this implies that if $R \in SR^c_{super}$ then $R \in SR_{super}$ and the result easily follows. □

An algorithmic solution to the problem of finding the MDL reduct for an SDT can be given as a brute-force algorithm, which computes the reducts of all the possible instantiations, see Algorithm 1. It is easy to see that the worst case runtime complexity of this algorithm is exponential in the size of the input. Unfortunately, it is unlikely that an asymptotically more efficient algorithm exists. Indeed, if we consider the problem of finding *any* MDL reduct, then the number of instantiations of S is, in the general case, exponential in the number of objects, and for each such instantiation one should find the shortest reduct for the corresponding decision table, which is known to be NP-hard. Interestingly, we can prove that the following decision problem (i.e., does there exists a superset reduct of size $\leq k$?) related to finding MDL-Reducts is in NP^{NP} (i.e., the class of problems that can be checked in polynomial time with access to an oracle for SAT).

Theorem 3.2. *Let MDL-Reduct be the problem of deciding if, given an SDT S and $k \in \mathbb{N}^+$, the MDL reducts of S are of size $\leq k$. Then, MDL-Reduct is in NP^{NP}.*

Algorithm 1 The brute-force algorithm for finding MDL reducts of a superset decision table S.

```
procedure BRUTE-FORCE-MDL-REDUCT($S$: superset decision table)
    $reds \leftarrow \emptyset$
    $l \leftarrow \infty$
    $ists \leftarrow enumerate\text{-}instantiations(S)$
    for all $i \in ists$ do
        $tmp\text{-}reds \leftarrow find\text{-}shortest\text{-}reducts(i)$
        $len \leftarrow |red|$ where $red \in tmp\text{-}reds$
        if $len < l$ then
            $reds \leftarrow tmp\text{-}reds$
            $l \leftarrow len$
        else if $len = l$ then
            $reds \leftarrow reds \cup tmp\text{-}reds$
        end if
    end for
    return $reds$                                    ▷ The MDL reducts for S
end procedure
```

Proof. We need to show that there is an algorithm for verifying instances of $MDL\text{-}Reduct$ whose runtime is polynomial given access to an oracle for an NP-complete problem. Indeed, a certificate can be given by an instantiation I (whose size is clearly polynomial in the size of the input SDT) together with a minimal reduct r for I s.t. $|r| \leq k$. Verifying whether r is a minimal reduct for I can then be done in polynomial time with an oracle for NP, from which the result follows. □

From the above proof we can observe that the pair (I, r), used as a certificate, only requires that r is a reduct of I, which means that in general it is a superset super-reduct of S and not necessarily also a superset reduct.

While heuristics could be applied to speed up the computation of reducts [58] (specifically, to reduce the complexity of the $find\text{-}shortest\text{-}reducts$ step in Algorithm 1) the approach described in Algorithm 1 still requires enumerating all the possible instantiations. Therefore, in the following section, we propose two alternative definitions of reduct in order to reduce the computational costs.

3.3. Entropy Reducts

We begin with a definition based on the notion of entropy [54], which simplifies the complexity of finding a reduct for an SDT. Indeed, while finding Superset and MDL reducts requires to enumerate all possible instantiations of a given SDT (which, in general, are exponentially many in the size of the SDT), the two alternative notions of entropy-based reducts that we propose in this Section do not require such an enumeration.

Given a decision d, we can associate with it a pair of belief and plausibility functions. Let $v \in V_t$ and $[x]_B$ for $B \subseteq Att$ an equivalence class, i.e. $[x]_B =$

$\{x' \in U : \forall a \in B,\ \ a(x') = a(x)\}$. Then:

$$Bel_S(v|[x]_B) = \frac{|\{x' \in [x]_B : d(x') = \{v\}\}|}{|[x]_B|}$$

$$Pl_S(v|[x]_B) = \frac{|\{x' \in [x]_B : v \in d(x')\}|}{|[x]_B|}$$

For each $W \subseteq V_t$, the corresponding basic belief assignment is defined as

$$m(W|[x]_B) = \frac{|\{x' \in [x]_B : d(x') = W\}|}{|[x]_B|} \,. \tag{7}$$

Given this setting, we now consider two different entropies. The first one is the pignistic entropy $H_{Bet}(m)$ as defined in (5). As regards the second definition, we will not directly employ the AU measure (see equation (4)). This measure, in fact, corresponds to a quantification of the degree of conflict in the bba m, which is not appropriate in our context, as it would imply finding an instantiation which is maximally inconsistent. We thus consider a modification of the AU measure called *Optimistic Aggregate Uncertainty* (OAU), which is consistent with the optimistic approach to generalized risk minimization [28, 30, 31]. This measure, which has already been studied in the context of evidence theory [1], superset decision tree learning [29] and soft clustering [5], is defined as follows:

$$OAU(m) = \min_{p \in \mathcal{P}(m)} H_p(X) \,, \tag{8}$$

where m is a bba, and H is the Shannon entropy (see Section 2).

We now show how these two entropies can be defined for a given SDT. Let $SDT = \langle U, Att, t, d \rangle$ be an SDT, $B \subseteq Att$ be a set of attributes and denote by $IND_B = \{[x]_B : x \in U\}$ the collection of equivalence classes (granules) determined by B. Let $d_{[x]_B}$ be the restriction of d on the equivalence class $[x]_B$, that is $d_{[x]_B} = \{d(x') : x' \in [x]_B\}$. The H_{Bet} and OAU entropy of d, conditional on B, are defined as

$$\begin{aligned} H_{Bet}(d|B) &= \sum_{[x]_B \in IND_B} \frac{|[x]_B|}{|U|} H_{Bet}(d_{[x]_B}) \\ &= \sum_{[x]_B \in IND_B} \frac{|[x]_B|}{|U|} \frac{H(P^m_{Bet}(d_{[x]_B}))}{H(\hat{p}_m(d_{[x]_B}))} \\ &= \sum_{[x]_B \in IND_B} \frac{|[x]_B|}{|U|} \frac{\sum_{v \in d_{[x]_B}} P^{m(\cdot|[x]_B)}_{Bet}(v) * log(P^{m(\cdot|[x]_B)}_{Bet}(v))}{\sum_{v \in d_{[x]_B}} \frac{1}{|d_{[x]_B}|} * log(\frac{1}{|d_{[x]_B}|})} \end{aligned} \tag{9}$$

$$\begin{aligned} OAU(d|B) &= \sum_{[x]_B \in IND_B} \frac{|[x]_B|}{|U|} OAU(d_{[x]_B}) \\ &= \sum_{[x]_B \in IND_B} \frac{|[x]_B|}{|U|} min_{I \in \mathcal{I}(SDT)} \sum_{v \in \delta^I_B(x)} Pr(v|[x]^I_B) * log(\frac{1}{Pr(v|[x]^I_B)}) \end{aligned} \tag{10}$$

where $m(\cdot|[x]_B)$ is the bba determined by the granule $[x]_B$ (see Eq. 7), $P_{Bet}^{m(\cdot|[x]_B)}$ is the pignistic probability distribution (see Section 2), $[x]_B^I$ is the granule of x determined by $B \subset Att$ in the instantiation $I \in \mathcal{I}(SDT)$, δ_B^I is the generalized decision w.r.t B for the instantiation $I \in \mathcal{I}(SDT)$ (see Section 2), and $Pr(v|[x]_B^I)$ is the probability of class label v in the granule of x determined by $B \subset Att$ in instantiation I (see the definition of μ-reduct in Section 2).

Definition 3.5. *We say that $B \subseteq Att$ is*

- *an OAU super-reduct (resp., H_{Bet} super-reduct) if $OAU(d \mid B) \leq OAU(d \mid Att)$ (resp., $H_{Bet}(d \mid B) \leq H_{Bet}(d \mid Att)$);*
- *an OAU reduct (resp., H_{Bet} reduct) if no proper subset of B is also a super-reduct.*

As a further heuristic, we introduce appoximate reducts as follows.

Definition 3.6. *We say that $B \subseteq Att$ is*

- *an OAU ϵ-approximate super-reduct (resp., H_{Bet} ϵ-approximate super-reduct), with $\epsilon \in [0,1)$, if $OAU(d \mid B) \leq OAU(d \mid Att) - log_2(1-\epsilon)$ (resp., $H_{Bet}(d \mid B) \leq H_{Bet}(d \mid Att) - log_2(1-\epsilon)$);*
- *an OAU ϵ-approximate reduct (resp., H_{Bet} ϵ-approximate reduct) if no proper subset of B is also an ϵ-approximate super-reduct.*

It is easy to observe that both OAU and H_{Bet} naturally define two families of instantiations of the underlying SDT. Indeed, let B be an OAU reduct and let $[x]_B$ be one of the granules with respect to an OAU reduct. Then, a *OAU instantiation* is any instantiation $I_{OAU} \in \mathcal{I}(SDT)$ s.t.:

$$\begin{aligned} dec_{OAU}([x]_B) = \operatorname*{arg\,max}_{v \in V_t} \Big\{ & Pr(v|[x]_B^I) \ : \\ & I \in \{ \operatorname*{arg\,min}_{J \in \mathcal{I}(SDT)} \sum_{v \in \delta_B^J(x)} Pr(v|[x]_B^J) * log(\frac{1}{Pr(v|[x]_B^J)} \} \Big\} . \end{aligned} \tag{11}$$

That is, an OAU reduct determines an instantiation in which each object is assigned to the most probable among the classes, under the probability distribution which corresponds to the minimum value of entropy.
Similarly, a H_{Bet} *instantiation* with respect to $[x]_B$ is given by

$$dec_{H_{Bet}}([x]_B) = \operatorname*{arg\,max}_{v \in V_t} P_{Bet}^{m(\cdot|[x]_B)}(v) \tag{12}$$

We note that, in general, neither $dec_{OAU}([x]_B)$ nor $dec_{H_{Bet}}([x]_B)$ are unique: for the case of $dec_{OAU(B)}([x]_B)$ there may exist two instantiations $I, I' \in \mathcal{I}(SDT)$ with corresponding probability distributions p, p' (over the labels $v \in V_t$) s.t.

both $p, p' \in \arg\min_{p \in P_m} H_p(X)$; while for the case of $dec_{H_{Bet}}([x]_B)$ there may be two classes $v', v'' \in V_t$ s.t.

$$P_{Bet}^{m(\cdot|[x]_B)}(v') = P_{Bet}^{m(\cdot|[x]_B)}(v'') = max_{v \in V_t} P_{Bet}^{m(\cdot|[x]_B)}(v).$$

The following example shows, for a simple SDT, the OAU reducts, MDL reducts, and H_{Bet} reducts and their relationships.

Example 3.2. *Consider the superset decision table*

$$SDT = \left\langle U = \{x_1, ..., x_6\}, A = \{a_1, a_2, a_3, a_4\}, d \right\rangle$$

given in Table 1 and described in Example 3.1. We have that, for $B = \{a_2, a_3\}$:

$$OAU(d \,|\, A) = OAU(d \,|\, B) = 0.$$

Thus, B is an OAU reduct of SDT, as $OAU(d \,|\, a_2) = OAU(d \,|\, a_3) > 0$. It can easily be seen that B admits only a single OAU instantiation, which is given by $\{a_2, a_3\}$ is $dec_{a_2,a_3}(\{x_1, x_2\}) = dec_{a_2,a_3}(\{x_3, x_4\}) = 0$, $dec_{a_2,a_3}(\{x_5, x_6\}) = 1$. Indeed, every other possible assignment of class labels to the equivalence classes determined by B would result in a greater entropy.

Note that $\{a_4\}$ is also an OAU reduct and also in this case there exists a single corresponding OAU instantiation: this is given by $\{a_4\}$ is $dec_{a_4}(\{x_1, x_3, x_6\}) = 0$, $dec_{a_4}(\{x_2, x_4, x_5\}) = 1$.

On the other hand, $H_{Bet}(d \,|\, A) = \frac{1}{2}$, while $H_{Bet}(d \,|\, \{a_2, a_3\}) = 0.81$. Therefore, $\{a_2, a_3\}$ is not an H_{Bet} reduct. Notice that, in this case, there are no H_{Bet} reducts (excluding A). However, it can easily be seen that $\{a_2, a_3\}$ is an H_{Bet} approximate reduct when $\epsilon \geq 0.20$. We note that there exists 8 different H_{Bet} instantiations corresponding to the H_{Bet} reduct A: in all these instantiations we have that $dec_A(x_1) = dec_A(x_3) = 0$ and $dec_A(x_5) = 1$, while we have a different instantiation for each of the possible class assignments for the remaining objects.

As shown in Example 3.1, the unique MDL reduct is $\{a_4\}$, with corresponding MDL instantiation $dec_{MDL}(\{x_1, x_3, x_6\}) = 0$, $dec_{MDL}(\{x_2, x_4, x_5\}) = 1$. Thus, in this case, the MDL reduct is equivalent to one of the OAU reducts.

We note that also the other possible superset reduct (i.e. $\{a_2, a_3\}$, as shown in Example 3.1) is an OAU reduct: as we'll show in the next Section, this is a general property of OAU reducts.

Before studying the formal properties of the proposed entropy reducts, we observe that the computation of H_{Bet} and OAU entropies do not require one to enumerate all instantiations of a SDT, and can be performed in polynomial time. This is clearly immediate for the computation of H_{Bet}:

Proposition 3.1. *H_{Bet} can be computed in polynomial time, without enumerating the instantiations $I \in \mathcal{I}(SDT)$.*

Proof. In Equation 9 we only perform $|IND_B||d_{[x]_B}| \in O\left(|U||V_t|\right)$ operations, and there is clearly no dependency on $|\mathcal{I}(SDT)|$. □

The analogous result for the computation of the OAU entropy is less immediate (indeed, in Eq. 10 we need to solve a minimization problem over $\mathcal{I}(SDT)$), and rests on a previous characterization of this uncertainty measure [5, 29]:

Proposition 3.2. *OAU can be computed in polynomial time, without enumerating the instantiations $I \in \mathcal{I}(SDT)$.*

Proof. The OAU entropy can be computed efficiently through the P-LLE (Polynomial Lower Logical Entropy) algorithm proposed in [5]: the time complexity of this procedure is $\Omega\left(|V_t|\right)$ and $O(|U||V_t| * log|V_t|)$, and has no dependency on $|\mathcal{I}(SDT)|$. □

These properties imply that the computation of H_{Bet} and OAU reduct does not require one to enumerate the instantiations $I \in \mathcal{I}(SDT)$, and instead the required computations can be performed by directly relying on the statistics in the original SDT: this property will be useful for designing efficient (heuristic) procedures for searching reducts, as we show in Section 3.4, and is similarly useful for computing OAU and H_{Bet} instantiations. Indeed, it can easily be seen that, as a consequence of Propositions 3.1 and 3.2, the OAU (resp. H_{Bet}) instantiations, corresponding to a given OAU (resp. H_{Bet}) reduct, can be computed in polynomial time.

3.4. Properties of Reducts

In this section, we study the properties of, and relationships among, the different definitions of reducts on superset decision tables. In Example 3.2, it is shown that the MDL reduct is one of the OAU reducts. Indeed, we can prove that this holds in general.

Theorem 3.3. *Let R be an MDL reduct whose MDL instantiation is consistent (that is, $R \in R^c_{MDL}$). Then R is also an OAU reduct.*

Proof. As the instantiation corresponding to R is consistent, $OAU(d \,|\, R) = 0$. Thus, R is an OAU reduct. □

Corollary 3.1. *Finding the minimal OAU reduct for a consistent SDT is NP-hard.*

Proof. As any MDL reduct of a consistent SDT is also an OAU reduct and MDL reducts are by definition minimal, the complexity of finding a minimal OAU reduct is equivalent to that of finding MDL reducts. □

More in general, if we consider a consistent SDT, we can prove that the collection of OAU reducts and (consistent) superset reducts are equivalent, that is, the following result holds.

Theorem 3.4. *Let S be a consistent SDT, then $R^c_{super} = R_{OAU}$, that is each OAU reduct is a consistent superset reduct (and viceversa). Furthermore, for each $r \in R_{OAU}$ there exists $r' \in R_{super}$ (i.e. a superset reduct) s.t. $r' \subseteq r$, that is each OAU reduct is a superset super-reduct.*

Proof. Let $r \in R^c_{super}$, then its instantiation is consistent and hence $OAU(d|r) = 0$, thus $r \in R_{OAU}$. Conversely, let $r \in R_{OAU}$ and notice that every OAU instantiation (i.e., an instantiation s.t. $\forall [x]_r, d([x]_r) = dec_{OAU(r)}([x]_r))$ is necessarily consistent (as $OAU(d|r) = 0$). Hence, r is a reduct of a consistent instantiation, thus $r \in R^c_{super}$.

For the last part of the theorem, it suffices to notice that no inconsistent instantiation can be an OAU instantiation, and that each consistent superset reduct is also a (not necessarily consistent) superset super-reduct (by Theorem 3.1). The result follows. □

In inconsistent SDTs, only the last part of the previous theorem holds, as shown by the following theorem.

Theorem 3.5. *Let S be an inconsistent SDT. Then, for each $r \in R_{OAU}$, there exists $r' \in R_{super}$ s.t. $r' \subseteq r$.*

Proof. We can notice that each $r \in R_{OAU}$ corresponds to a OAU instantiation, whose Shannon entropy (by definition of the OAU measure) is minimal with respect to all possible instantiations. Thus, R_{OAU} is the collection of superset super-reducts whose corresponding instantiations have minimal entropy. Further, note that there may be $r' \in R_{super}$ s.t. $r' \subseteq r$. □

On the other hand, as shown in Example 3.2, the relationship between MDL reducts (or OAU reducts) and H_{Bet} reducts is more complex as, in general, an OAU reduct is not necessarily a H_{Bet} reduct. In particular, one could be interested in knowing whether an H_{Bet} (smaller than the whole set of attributes *Att*) exists and whether there exists a H_{Bet} reduct which is able to disambiguate objects that are not disambiguated when taking in consideration the full set of attributes *Att*. The following two results provide a characterization in the binary (i.e., $V_t = \{0, 1\}$), consistent case.

Theorem 3.6. *Let $B \subseteq Att$ be a set of attributes, $[x_1]_{Att}, [x_2]_{Att}$ be two distinct equivalence classes (i.e., $[x_1]_{Att} \cap [x_2]_{Att} = \emptyset$) that are merged by B (i.e., $[x_1]_B = [x_1]_{Att} \cup [x_2]_{Att}$), that are consistent and such that $|[x_1]_{Att}| = n_1 + m_1$, $|[x_2]_{Att}| = n_2 + m_2$, where the n_1 (resp., n_2) objects are such that $|d(x)| = 1$ and the m_1 (resp., m_2) objects are such that $|d(x)| = 2$. Then, $H_{Bet}(d \,|\, B) \geq H_{Bet}(d \,|\, Att)$, with equality holding iff one of the following two holds:*

1. *$m_1 = m_2 = 0$ and $n_1, n_2 > 0$;*
2. *$m_1, m_2 > 0$ and $n_1 \geq 0$, $n_2 = \frac{m_2 n_1}{m_1}$ (and, symmetrically when changing n_1, n_2).*

Proof. A sufficient and necessary condition for $H_{Bet}(d \,|\, B) \geq H_{Bet}(d \,|\, Att)$ is:

$$\frac{n_1 + \frac{m_1+m_2}{2} + n_2}{n_1 + m_1 + n_2 + m_2} \geq \max\left\{\frac{n_1 + \frac{m_1}{2}}{n_1 + m_1}, \frac{\frac{m_2}{2} + n_2}{n_2 + m_2}\right\} \tag{13}$$

under the constraints $n_1, n_2, m_1, m_2 \geq 0$, as the satisfaction of this inequality implies that the probability is more peaked on a single alternative. The integer solutions for this inequality provide the statement of the theorem. Further, one can see that the strict inequality is not achievable. □

Corollary 3.2. *On a binary consistent SDT, a subset $B \subseteq Att$ is a H_{Bet} reduct iff, whenever it merges a pair of equivalence classes, the conditions expressed in Theorem 3.6 are satisfied.*

Notably, these two results also provide an answer to the second question, that is, whether an H_{Bet} reduct can disambiguate instances that are not disambiguated when considering the whole attribute set *Att*. Indeed, Theorem 3.6 provides sufficient conditions for this property and shows that, in the binary case, disambiguation is possible only when at least one of the equivalence classes (w.r.t. *Att*), that are merged by the reduct, is already disambiguated.

As we described in the statement of Theorem 3.6, our result applies only to the binary case: indeed, the general n-ary case is much more complex and, in such cases, disambiguation could happen also in more general situations. This is shown by the following example.

Example 3.3. *Let $SDT = \langle U = \{x_1, ..., x_{10}\}, Att = \{a_1, a_2\}, d\rangle$ such that $\forall i \leq 5$, $d(x_i) = \{0, 1\}$ and $\forall i > 5, d(x_i) = \{1, 2\}$. Then, assuming the equivalence classes determined by Att are $\{x_1, ..., x_5\}, \{x_6, ..., x_{10}\}$, it holds that $H_{Bet}(d \,|\, Att) = 1$. If we further assume that a_1 determines a single equivalence class U, then it is easy to observe that $H_{Bet}(d \,|\, a_1) < 0.95 < H_{Bet}(d \,|\, Att)$ and hence a_1 is a H_{Bet} reduct.*

Note that the conditions expressed in Theorem 3.6 are satisfied for the set of all attributes Att, but Att is not a H_{Bet} reduct: indeed, if we consider the equivalence classes determined by Att, then $n_1 = n_2 = 0$ while $m_1 = m_2 = 5$ and therefore condition 2 in Theorem 3.6 holds. However, as previously shown, Att is not a H_{Bet} reduct.
Furthermore, note that Att is not able to disambiguate, since

$$dec_{H_{Bet}(Att)}([x_1]_{Att}) = \{0, 1\}\,,$$
$$dec_{H_{Bet}(Att)}([x_6]_{Att}) = \{1, 2\}\,.$$

On the other hand, $dec_{H_{Bet(a_1)}}(x_i) = 1$ for all $x_i \in U$. Notice that, in this case, $\{a_1\}$ would also be an OAU reduct (and hence a MDL reduct, as it is minimal).

On the other hand, regarding the relationships between H_{Bet} reducts and the other families of reducts, it is easy to show that, even on consistent SDTs, the conditions for existence of H_{Bet} reducts (smaller than the whole set of attributes *Att*) are quite restrictive. Indeed, the following result holds.

Theorem 3.7. *Let S be an SDT and r be an H_{Bet} reduct. Then, there exists $r' \subseteq r$ s.t. r' is an OAU reduct. That is, the collection of H_{Bet} reducts is a sub-collection of the OAU super-reducts.*

Proof. First, let us assume that S is consistent, and let $r \in R_{H_{Bet}}$. Then, since S is consistent, each $[x]_r$ is also consistent and therefore, by definition,

$OAU(d|r) = 0$ and r is an OAU super-reduct (but not necessarily also a OAU reduct). Consequently, the result holds for consistent SDTs.

For the inconsistent case, let r be an H_{Bet} reduct, and $\{[x]_r^i\}_i$ be the collection of the equivalence classes w.r.t. r. By definition of H_{Bet} reducts, we have $\sum_i Pr([x]_r^i) \cdot H_{Bet}(d|[x]_r^i) \leq H_{Bet}(d|Att)$. Therefore, for the (weighted) majority of equivalence classes the probability distributions $P_{Bet}(d|[x]_r^i)$ are more peaked (equivalently, less uniform) and, hence, there exists an instantiation I s.t. the probability distributions $P_I(d^I \,|\, x_r^i)$ are also more peaked. Hence, $OAU(d \,|\, r) \leq OAU(d \,|\, Att)$ holds. Notice, however, that this only guarantees that r is a OAU super-reduct, thus the result. □

As we did not find an appropriate generalization of Theorem 3.6 for the general multi-class case, we leave this as an open problem: such a result would be useful to provide general existence conditions for H_{Bet} reducts. Moreover, we also leave as open problem that of finding conditions required for an H_{Bet} to also be an OAU (or MDL) reduct.

Concerning the computational complexity of finding OAU or H_{Bet} reducts, since as we shown in the previous Section,both OAU and H_{Bet} can be computed in polynomial time, the following result holds as a simple consequence of the general hardness result for finding reducts in standard decision tables.

Theorem 3.8. *Finding all OAU (resp. H_{Bet}) reduct is NP-hard.*

Finally, we notice that, while the complexity of finding OAU (resp. H_{Bet}) reducts is still NP-hard, even in the approximate case, these definitions are more amenable to optimization through heuristics, as they employ a quantitative measure of quality for each attribute. Indeed, a simple greedy procedure can be implemented, as shown in Algorithm 2, which obviously has polynomial time complexity, and is guaranteed to find an OAU (resp., H_{Bet}) reduct (albeit not necessarily a minimal one).

Proposition 3.3. *Algorithm 2 returns an OAU (resp. H_{Bet}) reduct in polynomial time. In particular:*

- *The complexity of finding a OAU reduct is $O\left(|Att|^2|U||V_t| * log|V_t|\right)$;*
- *The complexity of finding a H_{Bet} reduct is $O\left(|Att|^2|U||V_t|\right)$*

Proof. That the algorithm returns an OAU (resp. H_{Bet}) reduct is obvious, thus we only need to prove that its complexity is polynomial in the size of the SDT.

Indeed, Algorithm 2 requires a polynomial number of evaluations of the OAU (resp. H_{Bet}) entropy: in particular, the number of such evaluations is $O\left(|Att|^2\right)$. As shown in Propositions 3.1 and 3.2, both OAU and H_{Bet} can be computed in polynomial time, thus the result follows. □

Thus, Algorithm 2 has a linear dependence in the number of objects, a linear (or log-linear, depending on whether H_{Bet} or OAU reducts are searched for) dependence in the number of possible class labels, and a quadratic dependence in the number of conditional attributes: we note that since usually

$|V_t| \ll min\{|U|, |Att|\}$, one can assume w.l.o.g. that the complexity of searching reducts is dominated by the leading term among $\{|U|, |Att|^2\}$, and it is thus more or less independent of the number of possible class labels.

Algorithm 2 A heuristic greedy algorithm for finding approximate entropy reducts of a superset decision table S.

```
procedure HEURISTIC-ENTROPY-REDUCT($S$: superset decision table, $\epsilon$: approximation level, $E \in \{OAU, H_{Bet}\}$)
    $red \leftarrow Att$
    $Ent \leftarrow E(d \,|\, red)$
    $check \leftarrow True$
    while check do
        Find $a \in red$ s.t. $\begin{cases} E(d \,|\, red \setminus \{a\}) \leq E(d \,|\, Att) - log_2(1-\epsilon) \\ E(d \,|\, red \setminus \{a\}) \text{ is minimal} \end{cases}$
        if $a$ exists then
            $red \leftarrow red \setminus \{a\}$
        else
            $check \leftarrow False$
        end if
    end while
    return $red$
end procedure
```

4. Experiments

In this section, we present a series of experimental studies meant to evaluate the different definitions of reduct in superset learning as put forward in this paper, as well as the performance of the proposed algorithms in light of the state-of-the-art in superset dimensionality reduction (DELIN algorithm, see Section 2). More specifically, our experiments are aimed at studying the following aspects:

- Reduct approximation: The ability of the different types of reducts to recover the true reducts (i.e., the reducts w.r.t. the true, but generally unknown, decision attribute t) when varying both the number of objects associated with a set-valued decision and the size of the set-valued decision.
- Predictive Performance: The quality of the selected feature subsets from a machine learning point of view. We measured the latter in terms of the predictive accuracy of a model trained on that subset of features, using a suitable algorithm for superset learning.

We conduct experiments with the following datasets from the UCI repository [20]:

- Iris: 150 objects, 3 classes, 4 attributes

- Boston house prices (Boston): 506 objects, 3 classes, 13 attributes
- Wine: 178 objects, 3 classes, 13 attributes
- Breast Cancer: 569 objects, 2 classes, 30 attributes
- Diabetes: 442 objects, 3 classes, 10 attributes
- Adult Census Income: 48842 objects, 2 classes, 14 attributes
- Abalone: 4177 objects, 15 classes, 8 attributes
- Forest Fires: 517 objects, 5 classes, 13 attributes

For the second experiment, we used the PL-KNN [29] classifier, a simple generalization of the k-nearest neighbor algorithm for superset learning. Of course, more sophisticated methods for superset learning might be used as well, and the choice of the learning methods may clearly influence the results. However, as one advantage of a simple nearest neighbor approach, let us mention that its performance critically depends on the underlying feature representation, which is exactly what we seek to capture. Many other algorithms have in-built feature selection or transformation capabilities, which may bias the results.

For each UCI dataset, we created 5 different SDTs, each one generated through random coarsening: For each value $y \in V_t \setminus \{t(x)\}$, a biased coin with success probability γ was flipped to decide whether or not it is added to the true decision $t(x)$ as an additional candidate. Obviously, the parameter γ allows for varying and controlling the degree of *ambiguity* [9, 39]. We considered the following values: 0% (i.e., the case in which $d(x) = t(x)$, which allows us to compute the true reducts for the SDT as a reference comparison), 5%, 10%, and 25%.

To estimate predictive performance, we adopted a 5-fold cross-validation approach: during each iteration, 4 folds were used for training while the remaining fold was used for testing. The training folds were used for feature selection, using the proposed methods and the DELIN algorithm, and for training the PL-KNN algorithm. The test fold was then used to measure the accuracy of the trained PL-KNN models. Specifically, we measured both the average accuracy across the 5 folds and the corresponding 95% confidence intervals.

4.1. Comparison of Reducts for Superset Decision Tables

In the first experiment, each dataset was discretized in a pre-processing step, i.e., numerical attributes were replaced by categorical attributes. In particular, since *Boston*, *Abalone* and *Forest Fires* are originally regression datasets (i.e., the target attribute t is continuous), we also discretized the target attribute. The discretization was performed by applying the k-means algorithm [37] with $k = 5$ (on the values of the respective numerical attribute, i.e., running k-means on a one-dimensional dataset) and $k = 2$ (on the values of the target attribute).

We evaluated five different algorithms: the brute-force enumeration algorithm for computing MDL reducts (see Algorithm 1), the brute-force enumeration algorithms for computing H_{Bet} and OAU reducts, and the greedy algorithms to compute H_{Bet} and OAU reducts (see Algorithm 2). The algorithms were compared with respect to both their running time and their ability to recover the true reducts (that is, the reducts on the SDT with 0% ambiguity degree). A time budget of 10,000 seconds was assigned to each algorithm. The results of the experiments are reported in Tables 2–10. Based on these results, the following observations can be made:

- Computing MDL reducts, at least through the application of the brute-force algorithm (see Algorithm 1), is in general infeasible in terms of computation time. Indeed, among all 8 examined datasets, only on two 5% SDT and only on one 10% SDT, the algorithm finished the computation within the time budget. The two datasets were the smallest in terms of number of objects and attributes. This is hardly surprising, as the time complexity of Algorithm 1 is exponential in both the number of attributes and the number of objects. In the average case, we expect the algorithm to have a time complexity of $O(2^{|Att|} \cdot 2^{\epsilon|U|})$ on an ϵ% SDT. Let us also note that for all three datasets, the MDL reducts coincided with the minimal OAU reducts. This finding is interesting as, in light of Theorems 3.3 and 3.5, we know that the two definitions of reducts are equivalent only for consistent SDT, while all the considered SDTs were actually inconsistent.

- Regarding OAU reducts, it is interesting to observe that on all datasets, in the 5% and 10% SDT, the true reducts (that is, the reducts on the 0% SDT) were among the OAU reducts, and in all cases but three (Wine, Boston, and Forest Fires), the OAU reducts coincided with the true reducts. For the 25% SDT, on all datasets but three (Boston House Prices, Breast Cancer, Adult Census Income), the true reducts were among the OAU reducts, while on the three remaining datasets, the OAU reducts were subsets of the true reducts. Thus, from an empirical point of view, the notion of OAU reduct seems to be effective as a method to discover the true reducts.

- On the other hand, regarding H_{Bet} reducts, in only three 5% SDT (Forest Fires, Abalone, Iris) and in only one 10% SDT (Forest Fires), the H_{Bet} (minimal) reducts were among the true (minimal) reducts. In only one case (the 5% SDT for dataset Iris), the H_{Bet} reducts coincided with the true reducts, while in all other cases the H_{Bet} reducts were either a subfamily of the true reducts or super-reducts (indeed, in most cases the only H_{Bet} reduct was the set *Att* of all attributes). Thus, compared with OAU reducts, the requirement imposed by H_{Bet} entropy seems to be too conservative. This provides a stronger empirical counterpart of Theorems 3.2 and 3.7 and suggests that, in most practical cases, the requirements for the existence of H_{Bet} reducts are strictly stronger than those for OAU reducts.

Table 2: Results for dataset Iris.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	1 reduct (2,3)				
5%	1 reduct (2,3) 60 sec	1 reduct (2,3) 0.24 sec	1 reduct (2,3) 0.17 sec	(2,3) 0.15 sec	(2,3) 0.13 sec
10%	1 reduct (2,3) 5570 sec	1 reduct (2,3) 0.24 sec	1 reduct A 0.17 sec	(2,3) 0.15 sec	A 0.13 sec
25%	-	2 reducts (1,2)(2,3) 0.24 sec	1 reduct A 0.17 sec	(1,2) 0.15 sec	A 0.13 sec

Table 3: Results for dataset Boston house prices.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	2 reducts, 1 minimal (0, 3, 5, 6, 7, 10, 11, 12)				
5%	-	2 reducts, 1 minimal (0,3,5,6,7,10,11,12) 86 sec	1 reduct A 70 sec	(0,1,5,6,7,8,10,11,12) 1.73 sec	A 1.26 sec
10%	-	2 minimal reducts (0,3,5,6,7,10,11,12) (0,5,6,7,8,10,11,12) 86 sec	1 reduct A 70 sec	(0,3,5,6,7,10,11,12) 1.73 sec	A 1.26 sec
25%	-	1 minimal reducts (0,5,6,7,10,11,12) 86 sec	1 reduct A 70 sec	(0,5,6,7,10,11,12) 1.73 sec	A 1.26 sec

- As for the approximate entropy (both OAU and H_{Bet}) computed according to Algorithm 2, the computed reduct was in all cases except two (the Wine dataset for H_{Bet} reducts, Boston House Prices for OAU reducts) a minimal reduct (according to the respective definition of entropy reducts). In particular, the approximate Algorithm for computing OAU reducts was able to recover one of the true minimal reducts in most datasets, at a computational cost which was, on average, at least ten times smaller. Thus, the heuristic greedy algorithm for finding OAU reducts seems to be effective in finding minimal reducts with significantly reduced time complexity.

4.2. Comparison between Rough Set Feature Selection and DELIN

Based on the results of the first experiment, we decided to use the algorithm for computing OAU reducts for the second study, since this algorithm has shown strong performance in discovering the real reducts, as discussed in Section 4.1. Specifically, we evaluated the greedy algorithm for computing OAU reducts (see Algorithm 2), in order to limit the execution time, as the evaluation was

Table 4: Results for dataset Breast Cancer.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	6 reducts (0,3,4,5,6,10,11,12,13,14)(0,3,4,5,7,10,11,12,13,14)(0,3,4,5,8,10,11,12,13,14) (1,3,4,5,6,10,11,12,13,14)(1,3,4,5,7,10,11,12,13,14)(1,3,4,5,8,10,11,12,13,14)				
5%	-	6 reducts As in the 0% SDT 3316 sec	1 reduct A 3186 sec	(0,3,4,5,6,10,11,12,13,14) 15.3 sec	A 13.8 sec
10%	-	6 reducts As in the 0% SDT 3316 sec	1 reduct A 3186 sec	(0,3,4,5,6,10,11,12,13,14) 15.3 sec	A 13.8 sec
25%	-	1 minimal reducts (0,5,6,7,10,11,12) 86 sec	1 reduct A 70 sec	(0,5,6,7,10,11,12) 1.73 sec	A 1.26 sec

Table 5: Results for dataset Diabetes.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	2 reducts (0,1,2,3,4,6,7,8,9)(0,1,2,3,5,6,7,8,9)				
5%	-	2 reducts As in the 0% SDT 147 sec	1 reduct A 147 sec	(0,1,2,3,4,6,7,8,9) 16.2 sec	A 16.7 sec
10%	-	2 reducts As in the 0% SDT 147 sec	1 reduct A 147 sec	(0,1,2,3,4,6,7,8,9) 16.2 sec	A 16.7 sec
25%	-	2 reducts As in the 0% SDT 147 sec	1 reduct A 147 sec	(0,1,2,3,4,6,7,8,9) 16.2 sec	A 16.7 sec

Table 6: Results for dataset Adult Census Income.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	8 reducts, 2 minimal (1,2,4,5,7,8,11,12,13,14)(1,2,4,5,7,10,11,12,13,14)				
5%	-	8 reducts, 2 minimal As in the 0% SDT 2645 sec	4 reducts, 2 minimal (1,2,3,4,5,6,7,8,11,12,13,14) (1,2,3,4,5,6,7,10,11,12,13,14) 2637 sec	$A \setminus \{0, 3, 9, 10\}$ 15 sec	$A \setminus \{0, 9, 10\}$ 11 sec
10%	-	8 reducts, 2 minimal As in the 0% SDT 2645 sec	3 reducts, 2 minimal As in the 5% SDT 2637 sec	$A \setminus \{0, 3, 9, 10\}$ 15 sec	$A \setminus \{0, 9, 10\}$ 11 sec
25%	-	4 reducts, 1 minimal (1,2,4,5,7,11,12,13,14) 2645 sec	2 reducts, 1 minimal $A \setminus \{8, 9\}$ 2637 sec	(1,2,4,5,7,11,12,13,14) 15 sec	$A \setminus \{8, 9\}$ 11 sec

Table 7: Results for dataset Abalone.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	1 reduct (2,3,5,6,7)				
5%	-	1 reduct As in the 0% SDT 475 sec	1 reduct As in the 0% SDT 421 sec	(2,3,5,6,7) 16 sec	(2,3,5,6,7) 16 sec
10%	-	1 reduct As in the 0% SDT 475 sec	1 reduct (0,2,3,5,6,7) 421 sec	(2,3,5,6,7) 16 sec	(0,2,3,5,6,7) 16 sec
25%	-	2 reducts (2,3,5,6)(2,3,5,7) 475 sec	1 reduct A 421 sec	(2,3,5,6) 16 sec	A 11 sec

Table 8: Results for dataset Forest Fires.

	MDL reducts	OAU reducts	H_{Bet} reducts	Greedy OAU reduct	Greedy H_{Bet} reduct
0%	3 reducts, 2 minimal (0,1,3,4,5,6,7,8,10) (0,1,3,5,6,7,8,9,10)				
5%	-	3 reducts, 2 minimal As in the 0% SDT 1258 sec	2 reducts, 1 minimal (0,1,3,5,6,7,8,9,10) 1212 sec	(0,1,3,4,5,6,7,8,10) 14 sec	(0,1,3,5,6,7,8,9,10) 12 sec
10%	-	4 reducts, 3 minimal As in the 0% SDT plus (0,1,2,3,5,6,7,8,10) 1258 sec	1 reduct (0,1,3,5,6,7,8,9,10) 1212 sec	(0,1,2,3,5,6,7,8,10) 14 sec	(0,1,3,5,6,7,8,9,10) 12 sec
25%	-	4 reducts, 3 minimal As in the 10% SDT 1258 sec	1 reduct A 1207 sec	(0,1,2,3,5,6,7,8,10) 14 sec	A 9 sec

implemented using 5-fold cross-validation. For comparison, as already said, we used the DELIN algorithm. For the DELIN algorithm, at each iteration of 5-fold cross-validation procedure, the number of dimensions to be selected was set equal to the size of the minimal reduct found by the greedy OAU algorithm.[1]

For each iteration of the 5-fold cross-validation procedure, the training fold was used to both compute the minimal reducts (respectively, applying dimensionality reduction using the DELIN algorithm) and the reduced training set was then used to train the PL-KNN algorithm and the performance of the two feature selection approaches was compared by assessing the accuracy of the trained models on the reduced test fold. The results were then averaged across the 5 folds. The results are reported in Table 9. In most datasets (6 out of 8), the rough set-based feature selection algorithm performed better (in terms of average predictive accuracy) than the DELIN algorithm. In order to evaluate if the reported differences are statistically significant, we performed a Wilcoxon signed rank test [60] with a confidence level of 95% ($\alpha = 0.05$). The obtained statistic was $z = -3.2797$ (p-value = 0.001), which means the difference between the two algorithms is statistically significant at the selected confidence level. Thus, our results provide evidence in favor of the conjecture that the features selected by the rough set-based approach are more informative than the features constructed using the DELIN algorithm.

That said, these results should of course be taken with some caution. Indeed, one may argue that a direct comparison between the two algorithms is difficult, for example because OAU requires discrete data while DELIN is working on numerical attributes. Moreover, DELIN relies on certain assumptions regarding the distribution of the data, so that its performance will depend on whether or not these assumptions are met. Rough set-based feature selection methods, on the other side, are entirely non-parametric and thus allow more flexibility in modeling the relationship between the target and the features. While this is clearly an advantage, some information might be lost through the discretization of numerical features: future work should be devoted toward generalizing the proposed approach to encompass rough set-techniques that can directly manage continuous features.

In terms of computational complexity and running time, the DELIN algorithm is vastly more efficient than the standard brute-force algorithm to compute OAU reducts. Indeed, the algorithm for finding OAU reducts is combinatorial and, in general, has exponential running time (in the number of features). Compared to this, DELIN is based on LDA and has a running time which is essentially quadratic (more precisely, $O(|U||Att|^2)$) and can easily be implemented using standard linear algebra and optimization software. We note, though, that Algorithm 2, which was the method we adopted in our comparison, has the same computational complexity as DELIN, and in our experiments was shown to still be effective at finding the true reducts. Thus, the heuristic greedy approach to

[1] Moreover, since DELIN requires numerical features, categorical features were first one-hot encoded.

Table 9: Accuracy of the PL-KNN algorithm on reduced datasets, using both the OAU algorithm and the DELIN algorithm. For each dataset and level of ambiguity, the numbers in bold denote the feature selection algorithm resulting in the best performance.

Dataset	5%		10%		25%	
	OAU	DELIN	OAU	DELIN	OAU	DELIN
Iris	$\mathbf{0.91 \pm 0.09}$	$\mathbf{0.91 \pm 0.07}$	$\mathbf{0.90 \pm 0.10}$	0.83 ± 0.13	$\mathbf{0.90 \pm 0.10}$	0.82 ± 0.15
Cancer	0.91 ± 0.03	$\mathbf{0.92 \pm 0.03}$	0.89 ± 0.05	$\mathbf{0.90 \pm 0.03}$	0.89 ± 0.05	$\mathbf{0.90 \pm 0.05}$
Wine	$\mathbf{0.82 \pm 0.12}$	0.78 ± 0.08	$\mathbf{0.81 \pm 0.12}$	0.72 ± 0.17	$\mathbf{0.79 \pm 0.13}$	0.71 ± 0.17
Boston	$\mathbf{0.81 \pm 0.10}$	0.73 ± 0.12	$\mathbf{0.81 \pm 0.10}$	0.71 ± 0.12	$\mathbf{0.79 \pm 0.11}$	0.70 ± 0.12
Diabetes	$\mathbf{0.72 \pm 0.03}$	0.71 ± 0.03	$\mathbf{0.71 \pm 0.03}$	$\mathbf{0.71 \pm 0.05}$	$\mathbf{0.70 \pm 0.04}$	0.69 ± 0.05
Adult	$\mathbf{0.73 \pm 0.04}$	0.72 ± 0.04	$\mathbf{0.73 \pm 0.04}$	0.72 ± 0.04	$\mathbf{0.73 \pm 0.04}$	0.72 ± 0.04
Forest Fires	$\mathbf{0.86 \pm 0.07}$	0.82 ± 0.07	$\mathbf{0.86 \pm 0.07}$	0.82 ± 0.07	$\mathbf{0.83 \pm 0.09}$	0.79 ± 0.10
Abalone	$\mathbf{0.76 \pm 0.07}$	$\mathbf{0.76 \pm 0.07}$	$\mathbf{0.75 \pm 0.07}$	$\mathbf{0.75 \pm 0.07}$	$\mathbf{0.75 \pm 0.09}$	$\mathbf{0.75 \pm 0.09}$

finding OAU reducts could be seen as a useful trade-off on large-scale datasets.

5. Conclusion

Addressing the problem of superset learning in the context of rough set theory, as we did in this paper, appears to be interesting and mutually beneficial for both sides:

- RST provides natural tools for *data disambiguation*, which is at the core of methods for superset learning, most notably the notion of a reduct. Here, the basic idea is that the plausibility of an instantiation of the data is in direct correspondence with the (information-theoretic) complexity it implies for the dependency between input features and target (decision) variable (and a reduct in turn captures just this complexity).
- For RST itself, the setting of superset learning is a quite natural extension of the standard setting of supervised learning, and comes with a number of interesting challenges and non-trivial generalizations of existing concepts.

One such challenge has been tackled in this paper, namely the question how to generalize the notion of a reduct as well as devising algorithms for feature selection on the basis of this notion.

To this end, we first proposed a generalization of decision tables and then studied a purely combinatorial definition of reducts inspired by the Minimum Description Length principle, which we called MDL reducts. Since, as we showed, the computational complexity of finding this type of reducts is NP-hard, we proposed two alternative definitions based on the notion of entropy and harnessing the natural relationship between superset learning and belief function theory. We then provided a characterization for both these notions in terms of their relationship with MDL reducts, their existence conditions and their disambiguation power. Moreover, we developed simple heuristic algorithms for computing approximate entropy reducts.

Table 10: Results for dataset Wine.

	MDL	OAU	H_{Bet}	Greedy OAU	Greedy H_{Bet}
0%	163 reducts, 9 minimal (0,1,4,5,8,9),(0,2,3,5,7,10),(0,2,3,5,10,11) (0,2,3,7,10,11),(0,2,5,8,9,11)(0,3,4,5,7,10) (0,3,4,5,8,9),(0,4,5,6,8,9),(4,5,6,9,10,12)				
5%	174 reducts, 16 minimal (0,1,2,5,6,9)(0,1,4,5,8,9) (0,2,3,5,6,9)(0,2,3,5,6,10) (0,2,3,5,7,10)(0,2,3,5,9,10) (0,2,3,5,10,11)(0,2,3,7,10,11) (0,2,5,6,8,9)(0,2,5,8,9,11) (0,3,4,5,7,10)(0,3,4,5,8,9) (0,4,5,6,8,9)(1,2,6,7,9,12) (4,5,6,9,10,12)(4,5,8,9,10,12) 9281 sec	174 reducts, 16 minimal Same as MDL reducts 98 sec	9 reducts, 6 minimal (0,2,3,6,7,8,9,11) (0,2,3,7,8,9,10,11) (0,2,4,7,8,9,10,11) (0,3,4,6,7,8,9,11) (0,3,5,6,7,8,9,11) (0,3,5,7,8,9,10,11) 70 sec	(0,2,5,6,8,9) 1.73 sec	(0,2,3,6,7,8,9,10,11) 1.26 sec
10%		188 reducts, 16 minimal The minimal reducts are as in the 5% SDT 98 sec	9 reducts, 3 minimal (0,1,2,3,7,8,9,10,11) (0,2,3,4,7,8,9,10,11) (0,2,4,5,7,8,9,10,11) 70 sec	(0,2,5,6,8,9) 1.73 sec	(0,2,3,6,7,8,9,10,11) 1.26 sec
25%	-	191 reducts, 34 minimal The minimal reducts are as in the 5% SDT plus (0,1,2,7,9,10)(0,1,2,7,10,12) (0,1,4,5,9,10)(0,2,3,4,5,10) (0,2,3,4,6,7)(0,2,3,4,7,10) (0,2,3,6,7,10)(0,2,3,7,8,10) (0,2,3,7,9,10)(0,2,3,7,10,12) (0,2,4,7,9,10)(0,2,5,7,8,9) (0,2,7,8,9,10)(0,3,4,5,6,10) (0,3,4,5,9,10)(0,3,4,5,10,11) (0,4,5,7,9,10)(0,4,5,8,9,11) 98 sec	1 reduct A 70 sec	(0,1,2,7,9,10) 1.73 sec	A 1.26 sec

Finally, we conducted experiments on real datasets in order to empirically compare the different definitions of reducts for superset learning and the algorithms for computing them. As a result of these experiments, we conclude that the definition based on OAU entropy seems to be more effective in terms of its ability to recover the true but unknown reducts, compared with the definition based on H_{Bet} entropy. We have also shown that our heuristic algorithm for computing approximate entropy provides an effective approach to finding minimal reducts with limited computational resources. Finally, we compared the proposed feature selection methods with a state-of-the-art dimensionality reduction algorithm for superset learning and showed that the proposed method leads to a significantly higher classification accuracy on a collection of benchmark datasets, thus highlighting its usefulness in applications.

While this paper provides a promising direction for the application of RST-based feature reduction in superset learning, it naturally leaves many questions open. Specifically, we plan to address the following problems in future works:

- In Theorem 3.5, we proved that, in general, OAU reducts are a sub-family of the superset super-reducts. However, our experiments also showed that in most cases (in which the MDL reducts were actually computed within the assigned time budget) the MDL reducts were exactly equivalent to the OAU reducts. Thus, the conditions for such an equivalence between the two definitions should be investigated in more depth;

- In Theorems 3.6 and 3.7, we described two characterizations of H_{Bet} reducts: first, showing sufficient and necessary conditions for their existence on *binary* decision tables; second, showing that, in general, H_{Bet} reducts are OAU super-reducts. Therefore, the generalization of Theorem

3.6 to the multi-class case, together with a characterization of the conditions for the equivalence between H_{Bet} reducts and OAU reducts, should be investigated;

- The proposed RST feature reduction methods require the available data to be discrete: otherwise, data discretization techniques need to be applied which, in turn, could have an impact on the results and performance of the feature selection procedure. While, at least in principle, scaling techniques [21] (such as those applied in Formal Concept Analysis) could be applied to manage continuous features, these would likely have a huge impact on the computational complexity of the proposed methods. Thus, the generalization of the proposed approach to also encompass RST techniques that can directly manage continuous features, such as neighborhood- [43] or fuzzy-rough [32] based approaches, should be investigated;

- We studied the application of RST feature reduction to the superset learning task, however, it would also be interesting to study an extension of the proposed framework to other, even more general settings, such as learning from fuzzy [14, 28] or evidential [8, 11, 13, 48, 41] data.

- In this paper, the superset assumption was motivated by the problem of imprecise labeling. As explained in Section 2.3, this “don’t know” interpretation can be distinguished from a “don’t care” interpretation. Proceeding from the latter, a reduct can be considered as a maximally simple (least cognitively demanding) yet satisfying decision rule. Interestingly, in spite of very different interpretations, the theoretical problems that arise are essentially the same as those studied in this paper. Nevertheless, elaborating on the idea of reduction as a means for finding satisfying decision rules from a more practical point of view is another interesting direction for future work.

References

[1] Joaquin Abellan. Combining nonspecificity measures in dempster–shafer theory of evidence. *International journal of general systems*, 40(6):611–622, 2011.

[2] Joaquin Abellan and Serafin Moral. Completing a total uncertainty measure in the dempster-shafer theory. *International Journal Of General System*, 28(4-5):299–314, 1999.

[3] Pierre C Bellec, Arnak S Dalalyan, Edwin Grappin, Quentin Paris, et al. On the prediction loss of the lasso in the partially labeled setting. *Electronic Journal of Statistics*, 12(2):3443–3472, 2018.

[4] Rafael Bello and Rafael Falcon. Rough sets in machine learning: A review. In *Thriving Rough Sets*, pages 87–118. Springer, 2017.

[5] Andrea Campagner and Davide Ciucci. Orthopartitions and soft clustering: Soft mutual information measures for clustering validation. *Knowledge-Based Systems*, 180:51–61, 2019.

[6] Andrea Campagner, Davide Ciucci, and Eyke Hüllermeier. Feature reduction in superset learning using rough sets and evidence theory. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 471–484. Springer, 2020.

[7] Leilei Chang, Chao Fu, Wei Zhu, and Weiyong Liu. Belief rule mining using the evidential reasoning rule for medical diagnosis. *International Journal of Approximate Reasoning*, 130:273–291, 2021.

[8] Etienne Côme, Latifa Oukhellou, Thierry Denoeux, and Patrice Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern recognition*, 42(3):334–348, 2009.

[9] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

[10] Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 57–72. Springer, 2008.

[11] T Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5):804–813, 1995.

[12] Thierry Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 737–760. Springer, 2008.

[13] Thierry Denoeux. Maximum likelihood estimation from uncertain data in the belief function framework. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):119–130, 2011.

[14] Thierry Denœux and Lalla Meriem Zouhal. Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy sets and systems*, 122(3):409–424, 2001.

[15] Adrian Dobra and Stephen E Fienberg. Bounds for cell entries in contingency tables given marginal totals and decomposable graphs. *Proceedings of the National Academy of Sciences*, 97(22):11885–11892, 2000.

[16] Didier Dubois and Henri Prade. Properties of measures of information in evidence and possibility theories. *Fuzzy sets and systems*, 24(2):161–182, 1987.

[17] Bradley Efron. Censored data and the bootstrap. *Journal of the American Statistical Association*, 76(374):312–319, 1981.

[18] Lei Feng and Bo An. Leveraging latent label distributions for partial label learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2107–2113, 2018.

[19] Lei Feng and Bo An. Partial label learning with self-guided retraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3542–3549, 2019.

[20] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[21] Bernhard Ganter and Rudolf Wille. Conceptual scaling. In *Applications of combinatorics and graph theory to the biological and social sciences*, pages 139–167. Springer, 1989.

[22] Romain Guillaume and Didier Dubois. Robust parameter estimation of density functions under fuzzy interval observations. In *9th International Symposium on Imprecise Probability: Theories and Applications (ISIPTA'15)*, pages 147–156, 2015.

[23] Romain Guillaume and Didier Dubois. A maximum likelihood approach to inference under coarse data based on minimax regret. In *International Conference Series on Soft Methods in Probability and Statistics*, pages 99–106. Springer, 2018.

[24] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[25] David Harmanec and George J Klir. Measuring total uncertainty in dempster-shafer theory: A novel approach. *International Journal of General Systems*, 22(4):405–419, 1994.

[26] Ulrich Hohle. Entropy with respect to plausibility measures. In *Proceedings of 12th IEEE International Symposium on Multiple Valued Logic, Paris, 1982*, 1982.

[27] E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.

[28] Eyke Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.

[29] Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

[30] Eyke Hüllermeier and Weiwei Cheng. Superset learning based on generalized loss minimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 260–275. Springer, 2015.

[31] Eyke Hüllermeier, Sébastien Destercke, and Ines Couso. Learning from imprecise data: Adjustments of optimistic and pessimistic variants. In Nahla Ben Amor, Benjamin Quost, and Martin Theobald, editors, *Scalable Uncertainty Management - 13th International Conference, SUM 2019, Compiègne, France, December 16-18, 2019, Proceedings*, volume 11940 of *Lecture Notes in Computer Science*, pages 266–279. Springer, 2019.

[32] Richard Jensen and Qiang Shen. Fuzzy-rough sets assisted attribute selection. *IEEE Transactions on fuzzy systems*, 15(1):73–89, 2007.

[33] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in neural information processing systems*, pages 921–928, 2003.

[34] Radim Jiroušek and Prakash P Shenoy. A new definition of entropy of belief functions in the dempster–shafer theory. *International Journal of Approximate Reasoning*, 92:49–65, 2018.

[35] Radim Jiroušek and Prakash P Shenoy. On properties of a new decomposable entropy of dempster-shafer belief functions. *International Journal of Approximate Reasoning*, 119:260–279, 2020.

[36] A-L Jousselme, Chunsheng Liu, Dominic Grenier, and Éloi Bossé. Measuring ambiguity in the evidence theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(5):890–903, 2006.

[37] Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

[38] Ming Li, Paul Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*. Springer, 3rd edition, 2008.

[39] Liping Liu and Thomas Dietterich. Learnability of the superset label learning problem. In *International Conference on Machine Learning*, pages 1629–1637, 2014.

[40] Liping Liu and Thomas G Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in neural information processing systems*, pages 548–556, 2012.

[41] Liyao Ma, Sébastien Destercke, and Yong Wang. Online active learning of decision trees with evidential data. *Pattern Recognition*, 52:33–45, 2016.

[42] J.G. March and H.A. Simon. *Organizations*. Wiley, New York, 1958.

[43] Michinori Nakata, Hiroshi Sakai, and Keitarou Hara. Rule induction based on rough sets from information tables having continuous domains. *CAAI Transactions on Intelligence Technology*, 4(4):237–244, 2019.

[44] Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD*, pages 551–559, 2008.

[45] Qiang Ning, Hangfeng He, Chuchu Fan, and Dan Roth. Partial or complete, that's the question. *arXiv preprint arXiv:1906.04937*, 2019.

[46] Zdzisław Pawlak. Rough sets. *International journal of computer & information sciences*, 11(5):341–356, 1982.

[47] Judea Pearl. Reasoning with belief functions: An analysis of compatibility. *International Journal of Approximate Reasoning*, 4(5-6):363–389, 1990.

[48] Benjamin Quost, Thierry Denoeux, and Shoumei Li. Parametric classification with soft labels using the evidential em algorithm: linear discriminant analysis versus logistic regression. *Advances in Data Analysis and Classification*, 11(4):659–690, 2017.

[49] Hiroshi Sakai, Chenxi Liu, Michinori Nakata, and Shusaku Tsumoto. A proposal of a privacy-preserving questionnaire by non-deterministic information and its analysis. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1956–1965. IEEE, 2016.

[50] Glenn Shafer. *A mathematical theory of evidence.* Princeton university press, 1976.

[51] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[52] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64:141–158, 2017.

[53] Andrzej Skowron and Cecylia Rauszer. The discernibility matrices and functions in information systems. In *Intelligent decision support*, pages 331–362. Springer, 1992.

[54] Dominik Slezak. Approximate entropy reducts. *Fundamenta Informaticae*, 53(3-4):365–390, 2002.

[55] Dominik Slezak and Soma Dutta. Dynamic and discernibility characteristics of different attribute reduction criteria. In *Lecture Notes in Computer Science*, volume 11103, pages 628–643, 2018.

[56] Philippe Smets. Information content of an evidence. *International Journal of Man-Machine Studies*, 19(1):33–43, 1983.

[57] Philippe Smets and Robert Kennes. The transferable belief model. *Artificial intelligence*, 66(2):191–234, 1994.

[58] K Thangavel and A Pethalakshmi. Dimensionality reduction based on rough set theory: A review. *Applied Soft Computing*, 9(1):1–12, 2009.

[59] Christopher Umans. On the complexity and inapproximability of shortest implicant problems. In *Automata, Languages and Programming*, pages 687–696, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[60] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[61] Jing-Han Wu and Min-Ling Zhang. Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 416–424, 2019.

[62] Ronald R Yager. Entropy and specificity in a mathematical theory of evidence. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 291–310. Springer, 2008.

[63] YY Yao and PJ Lingras. Interpretations of belief functions in the theory of rough sets. *Information sciences*, 104(1-2):81–106, 1998.

[64] Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian Conference on Machine Learning*, pages 96–111, 2016.

[65] Chunying Zhang, Xiaoze Feng, and Ruiyan Gao. Three-way decision models and its optimization based on dempster–shafer evidence theory and rough sets. *Granular Computing*, pages 1–10, 2019.

[66] Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[67] Shao-Pu Zhang, Pin Sun, Ju-Sheng Mi, and Tao Feng. Belief function of pythagorean fuzzy rough approximation space and its applications. *International Journal of Approximate Reasoning*, 119:58–80, 2020.

[68] Yan-Lan Zhang and Chang-Qing Li. Relationships between relation-based rough sets and belief structures. *International Journal of Approximate Reasoning*, 127:83–98, 2020.

[69] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.