THE THREE-WAY-IN AND THREE-WAY-OUT FRAMEWORK TO TREAT AND EXPLOIT AMBIGUITY IN DATA

ANDREA CAMPAGNER, FEDERICO CABITZA, AND DAVIDE CIUCCI

Abstract. In this paper, we address ambiguity, intended as a characteristic of any data expression for which a unique meaning cannot be associated by the computational agent for either lack of information or multiple interpretations of the same configuration. In particular, we will propose and discuss ways in which a decision-support classifier can accept ambiguous data and make some (informative) value out of them for the decision maker. Towards this goal we propose a set of learning algorithms within what we call the three-way-in and three-way-out approach, that is, respectively, learning from partially labeled data and learning classifiers that can abstain. This approach is based on orthopartitions, as a common representation framework, and on three-way decisions and evidence theory, as tools to enable uncertain and approximate reasoning and inference. For both the above learning settings, we provide experimental results and comparisons with standard Machine Learning techniques, and show the advantages and promising performances of the proposed approaches on a collection of benchmarks, including a real-world medical dataset. a logic of incomplete information, cast in the setting of possibility theory.

Keywords: Three-way decision, uncertainty, ambiguity, partial labels, machine learning.

1. Introduction and Related Works

Recently Machine Learning (ML) has continuously attracted the interest of the research community, both from a mathematical-theoretical point of view and, more predominantly, from an application point of view. This interest has been stimulated by the fact that different research communities (e.g. health-care and medicine, finance and economics, . . .) have acknowledged the ubiquity of *uncertainty*, in different forms, e.g. *vagueness*, *randomness*, *ambiguity*, as an *intrinsic* part of their practice [1, 2], and thus see ML as a potential tool to represent, manage and, in some cases, overcome this uncertainty.

However, uncertainty has been largely ignored in the mainstream ML community, despite the fact that this common condition can undermine the reliability and performance of ML systems when deployed in real-world settings [3].

The goal of this article is to analyze a specific form of uncertainty affecting data, which we denote with the common term of *ambiguity*. With this term we intend a characteristic of any data configuration with which a unique, clear meaning cannot be associated by a human or computational agent for either the coexistence of multiple interpretations of the same configuration, or a lack of (disambiguating) information. In particular, we will consider ways in which a classifier can accept

APPEARED IN INT. J. OF APPROXIMATE REASONING, 119: 292-312 (2020), DOI: 10.1016/J.IJAR.2020.01.010

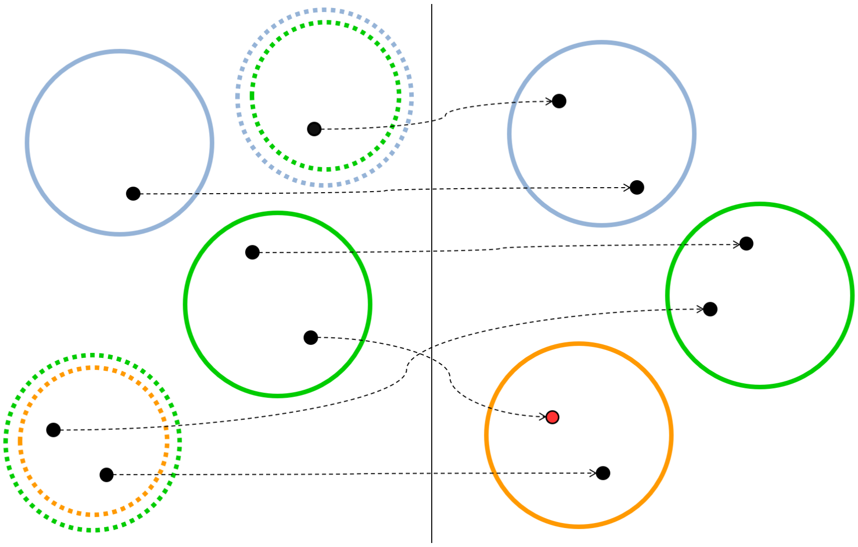

FIGURE 1. An illustration of three-way-in learning: classes are rendered as circles (different colors are associated with different classes). Objects are represented as dots. On the left there is the true labelling (training set); on the right, the classifier's labelling, possibly affected by misclassification errors (red dots). In three-way-in learning, the classes are disjoint, and objects in the input (on the left) can be labelled with uncertainty (and thus represented within circles with a multi-color dashed border). The classifier assigns each object to one and only one class in the output (on the right).

ambiguous data in the above sense, and provide the decision maker with some informative advice in spite of (and sometimes even in virtue of) this ambiguity. Our proposal is motivated by the idea that ambiguity should not be discarded, when it occurs in real-world data that are to be used as input data of model learning, i.e., as training data. Moreover, even classifiers could produce ambiguous output, in terms of non-discriminative predictions, if necessary and for the sake of better decision support.

We will consider classifiers as composed by two components involved in a two-step pipeline:

- The *learner*, which takes a set of labeled data as *input*, i.e. as training set, and according to this data it optimizes some parameters of the second component;
- The *predictor*, which, on the basis of a new instance and the parameters optimized by the learner, produces a new prediction for the instance at hand as *output*.

Under this view, we call *input* the input of the learner, i.e., the labeled training set, and *output* the predictions obtained by the predictor, i.e., the model learnt by the learner.

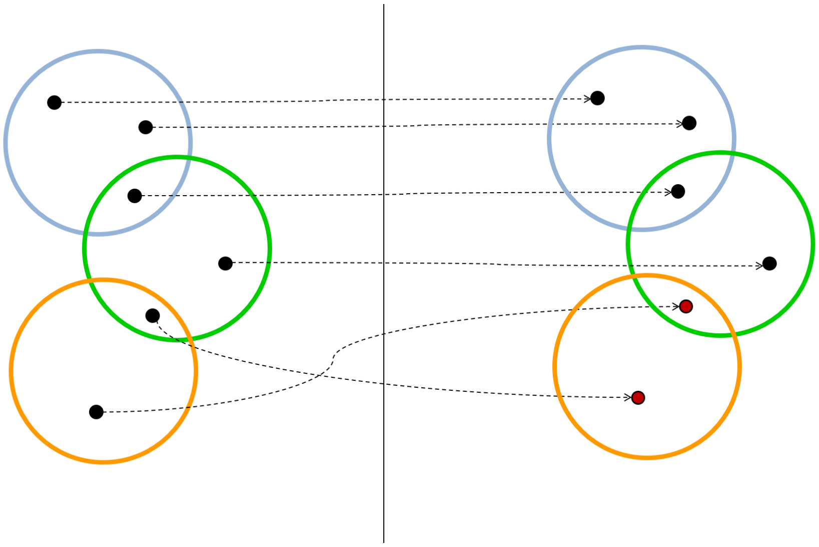

FIGURE 2. An illustration of multi-label classification: classes are rendered as circles (different colors are associated with different classes). Objects are represented as dots. On the left there is the true labelling (training set); on the right, the classifier's labelling, possibly affected by misclassification errors (red dots). In multi-label classification, the classes are not necessarily disjoint, and therefore the objects in the input (on the left) can belong to multiple classes. The classifier can assign objects to multiple classes in the output (on the right).

Furthermore, we distinguish between two types of ambiguity: *r-ambiguity* (for *row* ambiguity); and *c-ambiguity* (for *column* ambiguity). In particular:

- We have r-ambiguity when, given a single instance x in the training set, there is no specific classification that we can assign to x, that is, x has a set-valued labeling. Notice that *r-ambiguity* is essentially different from *multi-label* classification: this latter classification assumes a set-valued but *certain* labeling, that is each input instance x may naturally belong to different categories (this frequently occurs in domains such as text categorization or bio-informatics [4]). R-ambiguity also potentially causes a given instance to have a set-valued labeling, but in this case the set represents a *degree of uncertainty*: only one or, potentially, none[1] of the labels in the set is the correct one, while the others are assigned to instance x only due to our incomplete or uncertain knowledge.
- We have c-ambiguity when multiple instances $x_1, ..., x_n$, that are *sufficiently similar* to each other, have different classifications associated with them and, thus, no clear-cut classification can be assigned to any of them. This concept is similar to, but generalizes, *inconsistency* in Rough Set Theory:

[1] In this case the proper learning setting is a generalization of *superset* learning [5].

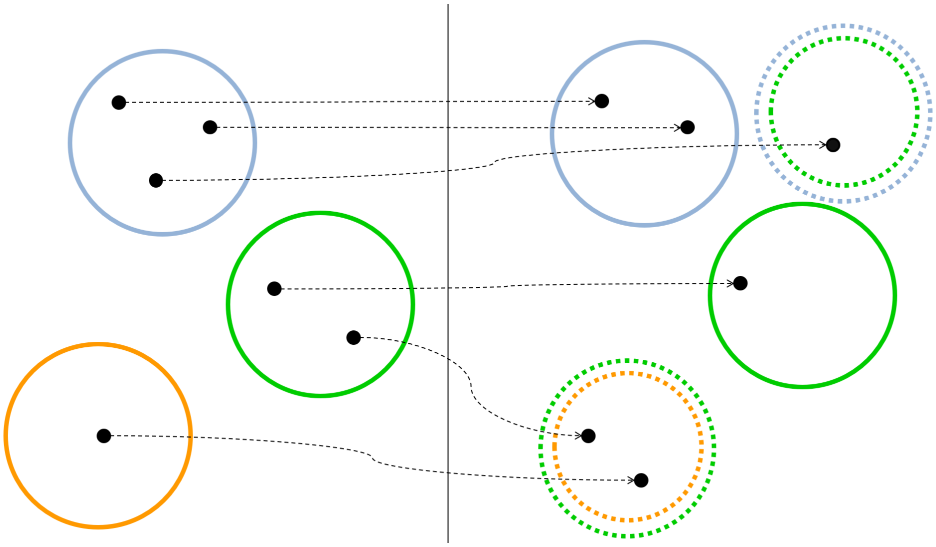

FIGURE 3. An illustration of three-way-out learning: classes are rendered as circles (different colors are associated with different classes). Objects are represented as dots. On the left there is the true labelling (training set); on the right, the classifier's labelling, possibly affected by misclassification errors (red dots). In three-way-out learning, the classes are disjoint, and the objects in the input (on the left) belong to one and only one class. The classifier can abstain on objects to avoid potential misclassifications.

we will allow, in general, the decision attribute to be set-valued, i.e., representing a degree of uncertainty over the real value of the decision. Furthermore, we will not typically assume the correctness of the decision attribute, i.e. different decisions associated with two indistinguishable objects could be due to errors or noise.

Similarly we will distinguish two different learning frameworks on the basis of these two distinct ambiguity types:

- *Three-way In* (TWI) learning, which concerns r-ambiguity, that is, when a set of different possible but mutually contradictory labels are assigned to each training instance. In this case, the learner takes this set-valued label representation and produces a model that can disambiguate its predictions as much as possible;
- *Three-way Out* (TWO) learning, which concerns c-ambiguity: in this case, the classifier must detect the ambiguities in the data representation and yield predictions accordingly. Thus, in this case, the predictor can *abstain* (completely or partially) on certain instances, whenever the available evidence is not sufficient to make a precise decision.

An intuitive representation of the two considered learning settings, compared with standard multi-class and multi-label classification, is depicted in Figures 1, 2, 3 and 4.

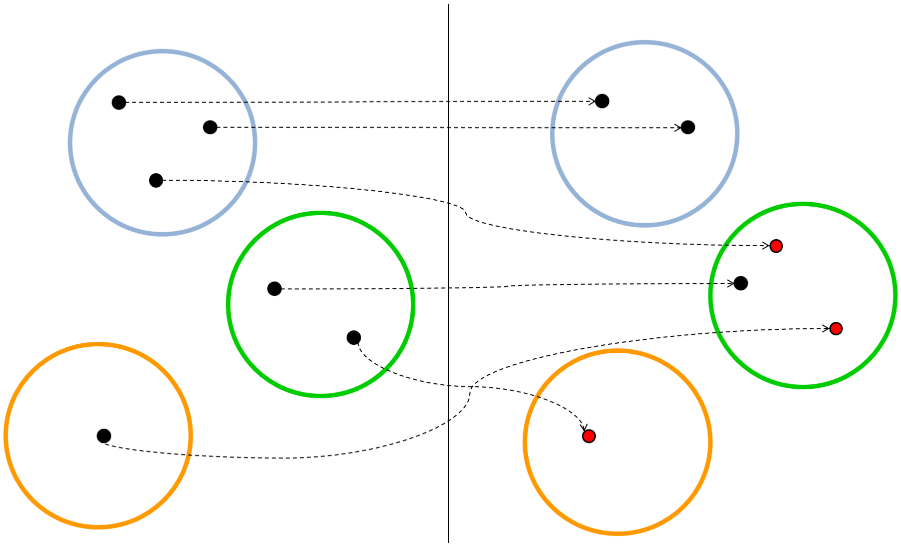

FIGURE 4. An illustration of multi–class classification: classes are rendered as circles (different colors are associated with different classes). Objects are represented as dots. On the left there is the true labelling (training set); on the right, the classifier's labelling, possibly affected by misclassification errors (red dots). In multi–class classification, the classes are disjoint, and the objects in the input (on the left) belong to one and only one class. The classifier assigns objects to one and only one class in the output (on the right).

Some previous research have been conducted in both learning tasks. The concept of the three-way-out learning, indeed, is based on, and adapts, the ideas originated in works about *three-way decisions* [6] and *cautious classification* [7, 8]. In fact, three-way decisions have been originally suggested, taking inspiration from human decision-making, as an approach to treat and manage the uncertainty in data by making use of a *third* category (different from positive and negative), that reflects a lack of knowledge, a (temporary) abstention, or a delayed decision while waiting for more evidence. This concept has been widely and successfully applied in the communities of Data Analysis and Machine Learning [9], and Decision Making [10]. Most relevantly with respect to the present work, we can cite the work on classification with uncertain boundaries [11], which presents a novel algorithm to properly take into account uncertain objects in binary classification by using rough sets and three-way decisions, a focus that we adopt in the discussion of the TWO learning paradigm. Also the work of [12], which proposes a sequential three-way decision based approach to manage multi-class decision problems, is relevant to our proposed techniques. However, we do not employ sequential methods, instead, we directly operate on a multi-class representation employing decision-theoretic techniques. Finally, we also cite the work presented in [13], which proposes a novel technique for ensemble construction based on three-way decisions. As already pointed out, we can see that the concept of r-ambiguity shares similarities with the concept of inconsistency in Rough Set Theory. Indeed, in the study of inconsistent decision tables the notion of *possible rules* [14, 15] represents a form of what we call Three-Way-Out (TWO) learning. Differently from the standard Rough Set techniques tackling this problem, our approach is not based on fundamental Rough Set theoretic notions (such as reducts and approximations), nor we consider rule-based approaches. In fact, our work proposes generalizations of standard Machine Learning algorithms like tree-based, ensemble-based or optimization-based classifiers. The decision-theoretic framework that we adopt, however, shares some similarities with works on Decision-Theoretic Rough Sets [16, 17] and, relevantly for our discussion of ensemble and optimization-based approaches, on *Interval-Valued Decision-Theoretic Rough Sets* [18] .

On the other hand, Three-way-In (TWI) learning originates from the seminal work on learning from ambiguous [19], superset [5] or partial labels [20] that proposed, under the standard optimization-based framework of modern Machine Learning literature, a generalization of the semi-supervised learning setting. Similar related approaches, which offer a different perspective that is more focused on attribute reduction and rule induction, have been investigated in the Rough Set and three-way decision communities by considering approaches applicable to semi-supervised and incomplete decision tables [21, 22]. However, TWI learning consists in a generalization of these approaches by not assuming that the real label is in the superset labeling of a given instance x. Moreover, the algorithms that we propose represent a generalization of ensemble learning and convex optimization approaches to Machine Learning, augmented with the capability to take into account the intrinsic uncertainty and ambiguity of these learning problems.

Our main goal in this article is to provide a unified mathematical framework for these two learning tasks, which encompasses algorithms and techniques by which to explicitly manage the uncertainty representation. To this aim, we employ the concept of *orthopartitions* [23] as a general representation framework for both learning problems and use techniques based on *evidence theory* and *three-way decision*, to implement, respectively, generalized uncertain reasoning based on lower and upper probabilities (which, as we will show, arise naturally when considering orthopartitions), and meaningful criteria based on decision-theory to treat and manage uncertainty.

The rest of this work will be organized as follows: in Section 2 we will present the mathematical background necessary for the development of the techniques and approach that we propose. In Section 3, we will present a general background for the Three-way Out setting, illustrating both general-purpose techniques and algorithm-tailored ones to tackle this learning task. In Section 4, we will describe the Three-way In learning setting in greater detail, explaining both how this setting can occur and also algorithmic techniques to learn classifiers in this context. In Section 5, we will present an experimental validation of the proposed techniques compared with previous existing methods and, finally, in Section 6, we will make the concluding comments and suggest some future line of work.

2. Basic Notions

In this section, we give the mathematical background on decision tables, orthopairs and orthopartions that will be used in the following.

Definition 1. *A* multi–observer decision table *is a tuple* $\langle U, A, t, D\rangle$ *where*

- *U is a universe of objects of interest;*
- *A is a set of attributes (or features) that we use to represent objects in U. In particular, we define each attribute as a function $a : U \mapsto V_a$ where V_a is the domain of values that the attribute a can assume;*
- *$t \notin A \cup D$ is a distinguished decision attribute, that we assume to be the* real *(but, in general, possibly unknowable) decision associated with the object in U, we will denote with Y the domain of values of t;*
- *D, with $D \cap (A \cup \{t\}) = \emptyset$, is a set of decision attributes that represent the decisions (possibly incorrect) that a set of observers assign to objects in U; when $|D| = 1$ we will use simply the notation d and call $\langle U, A, t, d\rangle$ simply* decision table [24]*. We will assume that $\forall d \in D$, $V_d = \mathcal{P}(Y)$.*

Based on this notion of decision table, we can now formally define the notions of c-ambiguity and r-ambiguity.

Definition 2. *Let dist be a distance function $dist : U \times U \mapsto \mathbb{R}^+$ and $M = \langle U, A, t, D\rangle$ a multi-observer decision table. Let $X\Delta Y = (X \setminus Y) \cup (Y \setminus X)$, where $X, Y \subseteq U$. Let $aggr : \mathcal{P}(Y)^{|D|} \mapsto \mathcal{P}(Y)$ be an aggregation function mapping the different decision attributes to a single (but possibly set-valued) decision attribute. Then*

- *M is* c-ambiguous *if*

  $\exists x_1, x_2 \in U : dist(x_1, x_2) < \epsilon$ *such that* $aggr[D(x_1)]\Delta aggr[D(x_2)] \neq \emptyset$
- *M is* inconsistent *if*

  $\exists x_1, x_2 \in U : dist(x_1, x_2) < \epsilon$ *such that* $aggr[D(x_1)] \cap aggr[D(x_2)] = \emptyset$
- *We say that M is* r-ambiguous *if $\exists x \in U$ such that $|aggr[D(x)]| > 1$.*

We can make two observations: first, M could be both r-ambiguous and c-ambiguous; second, inconsistency implies c-ambiguity. Indeed, if M is inconsistent on x_1, x_2 then $aggr[D(x_1)]\Delta aggr[D(x_2)] = aggr[D(x_1)] \cup aggr[D(x_2)] \neq \emptyset$. The reverse does not hold in general, e.g., let $aggr[D(x_1)] = \{1, 2\}$, $aggr[D(x_2)] = \{2, 3\}$., then M is c-ambiguous but not inconsistent.

It is then easy to observe that Definition 2 corresponds to the following intuitive concepts: c-ambiguity means that we are uncertain about the true class assignment for at least an object, r-ambiguity means that we have very similar objects but with potentially incompatible classifications, inconsistency is a stronger form of c-ambiguity in which the classifications of the two objects are definitely incompatible. Furthermore, notice that we made no assumptions on the correctness or "noise-freeness" of the available decision attributes (those in D). This means that there may be an instance x s.t. $\forall d \in D, t(x) \notin d(x)$ (i.e. we admit the possibility of noisy classifications, while in superset learning [5] or in traditional rough-set based analysis this possibility is rejected). In particular, if M is inconsistent on x_1, x_2 then either the decision associated to at least one of x_1, x_2 is noisy (i.e., $t \notin aggr[D(x_1)] \vee t \notin aggr[D(x_2)]$) or we are forced to reject standard smoothness or locality assumptions (i.e., there may be very close instances with different classifications).

The goal of the Machine Learning endeavor is, given a decision table $\langle S \subset U, A, D\rangle$, to recover the true decision attribute t, that is find an approximation, also called predictive model f, such that f well approximates t. In general, f may be expressed as a single element of Y or as a probability distribution defined over Y, representing the belief that the model assigns to the different alternatives. Other types of structures defined over Y (e.g. the class of the orderings over Y) are also possible. In particular, in the following, we will be interested in the case where $f : U \mapsto \mathcal{P}(Y)$, i.e., the labeling given by f assigns subsets of Y to each instance $x \in U$. With this representation we model *ambiguous* or *uncertain* assignments: that is, if $f(x) = \{y_1, ..., y_k\}$ then it is uncertain whether x belongs to class y_1, ..., or class y_k.

2.1. **Orthopairs and Orthopartitions.**

To express our partial knowledge we will use the notion of orthopartions, further, we will use the corresponding entropy functions to measure their intrinsic uncertainty [23].

Definition 3. *An* orthopair [25] *over the universe U is a pair of sets $O = \langle P, N\rangle$ such that $P, N \subseteq U$ and $P \cap N = \emptyset$, with P and N standing, respectively, for* positive *and* negative. *From these two sets we can also define a third set, called* boundary*, as $Bnd = (P \cup N)^c$.*

An orthopair represents the uncertainty of our knowledge on a set: specifically, the status of the elements in the boundary is uncertain (i.e., they can or cannot belong to the given set). Thus, a given orthopair stands for a collection of consistent sets.

Definition 4. *We say that an orthopair $O = (P_O, N_O)$ is* consistent *with a set $C \subset U$ if:*

$$x \in P_O \implies x \in C \land x \in N_O \implies x \notin C \tag{2.1}$$

Basically, if we consider C to represent a concept defined over U, i.e., a set of objects sharing a certain property, then a consistent orthopair O can be considered as an *approximation*, given by a lack of knowledge, that we have for that concept. Under this interpretation the set P_O contains the objects that surely are in C, while the set N_O contains the objects that surely are not in C. On the other hand, the boundary Bnd_O contains the objects whose belonging to the target concept C is uncertain. This may occur for different reasons. If the orthopair O represents the output of a classification algorithm, then the objects in the boundary may represent objects on which the classifier opted to abstain for lack of confidence, e.g., the prediction score of the best option is too low to conclusively claim it, due to uncertainty or noise in the data. As we will show, properly constructing the boundary (hence the abstention decision given by the predictor), allows the classifier to retain a high accuracy and precision but with increased reliability, avoiding over-commitment to potentially wrong decisions. On the other hand, if we have a degree of uncertainty with respect to the target concept C that we want to learn, then we can represent this uncertain and incomplete labelling using an orthopair, turning the analysis of these data into a problem of *semi-supervised* of *superset* [5] learning.

We can define different orderings between orthopairs, in particular we say that O_1 is *more informative* than O_2, denoted $O_1 \geq_I O_2$ if $P_2 \subseteq P_1$ and $N_2 \subseteq N_1$.

Two orthopairs O_1, O_2 are *disjoint* if it holds that:

$$P_1 \cap P_2 = \emptyset \tag{2.2a}$$

$$P_1 \cap Bnd_2 = \emptyset \land Bnd_1 \cap P_2 = \emptyset \tag{2.2b}$$

Definition 5. *An* orthopartition *is a collection* $\mathcal{O} = \{O_1, ..., O_n\}$ *of orthopairs such that the following axioms hold:*

$$\forall O_i, O_j \in \mathcal{O},\ \ O_i, O_j \text{ are disjoint} \tag{2.3a}$$

$$\bigcup_i (P_i \cup Bnd_i) = U \tag{2.3b}$$

$$\forall x \in U(\exists O_i \text{ s.t. } x \in Bnd_i) \implies (\exists O_j \text{ with } i \neq j \text{ s.t. } x \in Bnd_j) \tag{2.3c}$$

$$|\mathcal{O}| \leq |U| \tag{2.3d}$$

An orthopartition can be considered as a generalization of a partition or, in the ML context, of a multi–class classification, in which we can express *partial knowledge* with respect to the membership of the objects in the concept classes. This interpretation is given by extending the definition of *consistency with* orthopartitions.

Definition 6. *We say that a partition* π *is* consistent with an orthopartition $\mathcal{O}$ *iff* $\forall O_i \in \mathcal{O}$, $\exists!\, S_i \in \pi$ *such that* S_i *is consistent with* O_i*. We denote with* $\Pi_{\mathcal{O}}$ *the set of all partitions consistent with* $\mathcal{O}$*:* $\Pi_{\mathcal{O}} = \{\pi | \pi \text{ is consistent with } \mathcal{O}\}$*.*

Given an element $x \in U$ and an orthopartition $\mathcal{O}$ we define the *boundary set* of x as $Bnd(x) = \{O_i \in \mathcal{O} | x \in Bnd_i\}$. As with single orthopairs, the boundary set contains the objects whose class assignment cannot fully be determined from only the data.

The *entropy* of a partition π is defined as:

$$H(\pi) = -\sum_i \frac{|\pi_i|}{|U|} \cdot log_2 \frac{|\pi_i|}{|U|} \tag{2.4}$$

We can extend the definition of entropy to orthopartitions as follows (other definitions can be found in [23]):

$$H_{bet}(\mathcal{O}) = -\sum_i P_{bet}(O_i) \cdot log_2 P_{bet}(O_i) \tag{2.5}$$

where $P_{bet}(O_i) = \frac{1}{|U|} \cdot (|P_i| + \sum_{x \in Bnd_i} \frac{1}{|Bnd(x)|})$. We notice that H_{bet} gives a precise value by assigning a different weight to the instances on which we are uncertain, i.e., the ones belonging to the boundary. Starting from the definition of entropy, it is also possible to define a generalized mutual information measure as:

$$I_{bet}(\mathcal{O}_1, \mathcal{O}_2) = H_{bet}(\mathcal{O}_1) + H_{bet}(\mathcal{O}_2) - H_{bet}(\mathcal{O}_1 \wedge \mathcal{O}_2) \tag{2.6}$$

where $\mathcal{O}_1 \wedge \mathcal{O}_2 = \{\langle P_i \cap P_j, N_i \cup N_j \rangle | O_i \in \mathcal{O}_1 \text{ and } O_j \in \mathcal{O}_2\}$.

3. Three–way output

In this section, we will describe in greater detail the Three-way Out (TWO) learning setting. In this context, the chosen data representation, i.e., the selected features and/or their level of granularity, is such that a form of *c-ambiguity* arises. Thus, the chosen data representation does not allow us to distinguish different objects that are either identical or "too near" in the sample space, similarly to the concept of *indiscernibility* in standard Pawlak's rough sets [24] or generalized rough sets [26]. In fact, if these two or more indiscernible objects are associated to different classifications (this concept is usually known as *inconsistency* in rough set terminology) then we have a classification-level ambiguity: given a new instance the classifier would not be able to provide a clear-cut and error-free classification. The most meaningful approach to deal with this type of ambiguity, which amounts to a lack of evidence for certain cases or instances, consists in allowing the classifier to abstain, even partially, that is excluding some of the alternative possible classifications. This approach to classification has already been explored in three-way decision theory [6], that recently attracted great interest in the domain of *granular computing* [9] and Machine Learning [27], *possible rules* in Rough Set Theory [14] and *cautious classifiers* [7]. In doing so, there is a trade-off between the coverage of a classifier algorithm, i.e., the instances on which the classifier provides a decision, and its reliability and robustness to classification errors, while striving to keep as large an accuracy as possible: the goal is to learn classifiers that are still as precise as possible, but express a prediction only when *they are sufficiently confident*. While most three-way decision techniques and applications focus on the binary case, the approaches that we present in the following sections is also suited to the multi-class case, a problem which is typically tackled using *sequential three-way decisions* (e.g. see [12] as a recent example), in which a multi-class decision problem is converted into a sequence of binary (three-way) ones. Our work, on the other hand, is not based on a sequential approach. Instead, we employ a multi-class optimization-based approach to three-way decisions in order to infer an optimal orthopartition where the class boundaries represent the objects whose assignment to a specific class would undermine the reliability of the classifier. Furthermore, harnessing the fact that orthopartitions naturally define belief measures we also discuss evidence-theoretic approaches to information fusion that can be employed to implement ensemble-based and optimization-based solutions to TWO learning problem. More specifically in Section 3.1 we describe a general decision-theoretic approach (that can be seen as a generalization of the standard three-way decision-theoretic framework), while in Sections 3.2 and 3.4 we present algorithm-tailored techniques that are obtained by modifications of standard Machine Learning algorithms.

3.1. **Decision-Theoretic Approach.** In this section, we will present two strategies for converting probabilistic classifiers into three–way classifiers, generalizing the work in [28].

Let $A : X \mapsto \mathcal{PR}(Y)$ be a probabilistic classifier: i.e. for each $x \in X$, $A(x)_i$ represents the probability that algorithm A assigns to the event that x belongs to class $y_i \in Y$. We will also denote by $A(x)^*$ the ordering of $A(x)$ in terms of decreasing probability scores.

The first strategy is based on the idea that when different alternatives have probabilities which are *too close to each other* then, choosing among them is not a justified decision. In order to formalize this idea, let $\epsilon \in [0, 1]$, $d : [0, 1]^2 \mapsto \mathbb{R}$ a distance function and $A(x)^*$ as defined above. We say that

Definition 7. *A probabilistic classifier $A(x)$ is (m, ϵ)-ambiguous, where $m \in \{1, ..., |Y|\}$, if*

$$\forall i \leq m \;\; d(A(x)^*_1, A(x)^*_i) \leq \epsilon$$

Furthermore, we say that $A(x)$ is maximally (m^*, ϵ)-ambiguous if

$$m^* = max_m\{A(x)^* \text{ is } (m, \epsilon)\text{-ambiguous}\} \tag{3.1}$$

Equation 3.1 provides a simple criterion for converting a probabilistic classifier into a three–way classifier:

$$A(x)^{tw}_{amb} = \{A(x)^*_1, ..., A(x)^*_{m^*}\} \tag{3.2}$$

Evidently, this transformation has a time complexity of $O(m \cdot log\, m)$, required for ordering $A(x)$, or $O(m)$ in case $A(x)^*$ is already available.

Example 1. *Let $Y = \{1, 2, 3, 4, 5\}$ and A be a classifier such that, for a given instance x we have $A(x) = \langle 0.2, 0.3, 0.15, 0.1, 0.25\rangle$. Then, if we set $\epsilon = 0.1$, we have that $A(x)^*_1 = 0.3$ (thus, in particular the most probable alternative is 2), thus $A(x)^{tw}_{amb} = \{1, 2, 5\}$.*

As regards the second strategy, which is a generalization of previous work on three-way classification [28], it is based on a decision-theoretic framework and consists in balancing the costs of errors and abstentions.

Let

$$E = \begin{bmatrix} 0 & \epsilon_{12} & \cdots & \epsilon_{1|Y|} \\ \epsilon_{21} & 0 & \cdots & \epsilon_{2|Y|} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{|Y|1} & \epsilon_{|Y|2} & \cdots & 0 \end{bmatrix} \tag{3.3}$$

be the error cost matrix, i.e., $E_{i,j}$ is the cost of predicting y_j where the true class is y_i. To define the cost of abstention, it is not sufficient to set a constant value of α, otherwise the when the three-way algorithm decides to abstain, it will always output the full set of classes Y:

Theorem 1. [29] *If the abstention cost α is constant, then the minimal cost solution is always $Z = Y$, where Z is the output of the classifier.*

Thus, let $\alpha : \mathcal{P} \mapsto \mathbb{R}$ be a monotonically increasing set-valued function with $\alpha(Z) = 0$ $\forall Z$ s.t $|Z| = 1$ and where $\alpha(Z)$ represents the cost of abstaining among the alternatives in Z.

Definition 8. *Let $Z \subseteq Y$ representing a partial abstention decision, we define the* risk *of decision Z as:*

$$R(Z) = \alpha(Z) \cdot \sum_{y_i \in Z} A(x)_i + \sum_{y_j \notin Z} A(x)_j \cdot \frac{\sum_{y_i \in Z} \epsilon_{ji}}{|Z|} \tag{3.4}$$

This function defines the cost of taking the generalized decision Z, given the error and abstention costs and the evidence at hand (modeled by the probabilities in $A(x)$). Thus, the decision minimizing this objective function can be considered the most sensible (i.e., least risky) decision to be made by the classifier. Equation 3.4 provides another criterion for transforming a probabilistic classifier into a three–way one, namely

$$A(x)^{tw}_{dec} = argmin_Z\{|Z| : Z \in argmin_{Z'} R(Z')\} \tag{3.5}$$

That is, we select the generalized decision resulting in the minimal possible risk: in this case the algorithm should decide to abstain (totally or partially) when taking a single decision would incur in an unwarranted risk, that is, when, based on the costs of taking a wrong conclusion and the probability of the top-ranked alternatives, the risk would be greater than choosing to abstain.

Evidently, the basic procedure to obtain this transformation requires enumerating all the possible subsets of Y, thus its time complexity is $O(2^m)$. Interestingly, if $\forall i, j\ \ \epsilon_{ij} = \epsilon$ with ϵ constant, we can greatly reduce the computational cost. Indeed, as shown in [28], in this case the optimization problem in Equation 3.5 can be reformulated as:

$$argmin_j \alpha(j) * \sum_{i=1}^{j} A(x)^*_i + \epsilon * \sum_{i=j+1}^{m} A(x)^*_i \tag{3.6}$$

that can be solved in time $O(n)$ (if $A(x)^*$ is already available) using dynamic programming.

Example 2. *Let* $\epsilon = 1$ *and* $\alpha(Z) = \frac{|Z|-1}{|Y|-1}$*, with* $Y = \{1, 2, 3, 4, 5\}$*.*
Let A *be a classifier such that, for a given instance* x *we have*

$$A(x) = \langle 0.2, 0.3, 0.15, 0.1, 0.25\rangle$$

Then we have that: $R(\{2\}) = 0.7$*,* $R(\{1, 2\}) = 0.625 = R(\{1, 2, 5\})$*,*
$R(\{1, 2, 3, 5\}) = 0.775$*,* $R(Y) = 1$*. Thus* $A(x)^{tw}_{dec} = \{1, 2\}$*.*

The two just outlined strategies can be applied independently from the learning algorithm. However, it can be noted that these strategies do not consider the information about the ambiguity intrinsic in the data during the training phase, but only in the phase of predictions. When selecting a specific model class it is also possible to define model–specific strategies, as we are going to explain in the following sections.

3.2. **Decision Trees.** In [29] the authors propose a Three–Way Decision Tree (TWDT) model that more directly links the learning algorithm and three-way classification. Let $\mathcal{DT} = \langle S, A, t\rangle$ be a decision table with $S = \{x_1, ..., x_{|S|}\} \subseteq X$ and $A = \{a_1, ..., a_m\}$. Let $S^a_i = \{x \in S | v_a(x) = v^a_i\}$ be the set of instances that have value v^a_i for feature a. If a is a continuous attribute, then, given a threshold value v^a_i, we can consider

$$v_a(x_k) = \begin{cases} 1 & v_a(x_k) \geq v^a_i \\ 0 & \text{otherwise} \end{cases}. \tag{3.7}$$

The optimal classification C^a_i for S^a_i is the classification obtained by solving the decision rule described in Section 3. This class assignment is done locally on the tree nodes, and not only on the final output of the classifier. That is, if $Pr(y|S^a_i) = \frac{|\{x_k \in S^a_i : d(x_k) = y\}|}{|S^a_i|}$ then

$$C^a_i = argmin_{Z \subseteq Y} \alpha(Z) \cdot \sum_{y \in Z} Pr(y|S^a_i) + \sum_{y \notin Z} Pr(y|S^a_i) \cdot \frac{\sum_{y \in Z} \epsilon_{ji}}{|Z|} \tag{3.8}$$

Since this classification determines an orthopartition $\mathcal{O}_a$, we can then compute the *mutual information* of $\mathcal{O}_a$ w.r.t. S as described in Equation (2.6) and choose the feature a^* which results in the maximum mutual information value, and then recur on the subsets of S determined by feature a^* until a termination criterion is met.

Note that, especially in the more general case described by Equations (3.4) and (3.5), this approach requires the resolution of a complex decision optimization problem, whose complexity, in general, is exponential in $|Y|$, at each level of the tree. Furthermore, this split criterion essentially amounts to selecting the optimal split node on the basis of a generalized version of the accuracy. Accuracy is non-smooth, a property that is known to be detrimental to the induction of decision trees and tree-based ensembles. In fact, more efficient procedures can be obtained when using the criterion defined by Equation (3.2).

In order to cope with this limitations, we define a new criterion. Let $dist : X \times X \mapsto \mathbb{R}$ be a distance function defined among instances, then given a threshold value r, we can define the r-neighborhood of $x_k \in X$ as:

$$r[x_k] = \{x_l \in D_i^a | dist(x_k, x_l) \leq r\} \tag{3.9}$$

We can then compute the probabilities $Pr_i^a(y|x_k) = \frac{|\{x_l \in r[x_k] | d(x_l)=y\}|}{|r[x_k]|}$ and then the $\epsilon - ambiguity$ class of the top-alternative $y^{x_k} = argmax_y Pr_i^a(y|x_k)$, that is

$$\epsilon[y^{x_k}] = \{y' \in Y | d(Pr_i^a(y), Pr_i^a(y^{x_k})) < \epsilon\} . \tag{3.10}$$

On this basis, we can construct the orthopartition $\mathcal{O}_i^a = \{O_{y_1}, ..., O_{y_{|Y|}}\}$ given by the following assignment rule:

- If $\epsilon[y^{x_k}] = \{y_j\}$ then $x_k \in P_{y_j}$;
- If $\epsilon[y^{x_k}] = \{y_1, ..., y_n\}$ then $x_k \in Bnd_{y_1}, ..., Bnd_{y_n}$
- If $y_j \notin \epsilon[y^{x_k}]$ then $x_k \in N_{y_j}$.

Example 3. *Consider the set of instances in Table 1.*

Instance	a_1	a_2	d
x_1	*4.11*	*-4.73*	*2*
x_2	*-6.29*	*7.21*	*0*
x_3	*3.61*	*0.64*	*1*
x_4	*-1.44*	*-14.70*	*1*
x_5	*0.14*	*-9.93*	*2*
x_6	*1.45*	*-1.72*	*1*

Table 1. Example dataset

Then, if we consider instance x_1, with feature a_1 and threshold value $v_i^a = 1.45$ then $D_1^{a_1} = D \setminus \{x_2, x_4, x_5\}$ and $D_0^{a_1} = \{x_2, x_4, x_5\}$ according to equation (3.7). If we set $r = 6$, then $r[x_1] = \{x_1, x_3, x_6\}$, $Pr_1^{a_1}(1|x_1) = \frac{2}{3}$ and $Pr_1^{a_1}(2|x_1) = \frac{1}{3}$. Thus, if we set $\epsilon = 0.33$, it holds that $\epsilon[y^{x_1}] = \{1, 2\}$ thus $x_1 \in Bnd_1, Bnd_2$ and $x_1 \in N_0$.

The rationale behind this criterion is that we can consider the objects in $r[x_k]$ to be ambiguous with x_k, that is, given our desired level of granularity, expressed by the parameter r or, equivalently, by a number nn of nearest neighbors, we are not able to distinguish x_k from objects in $r[x_k]$. Thus, when deciding which classes to assign to x_k, we should consider that the ambiguity on the level of objects translates into a possible ambiguity, given by ϵ ambiguity, at the level of the classes. Given the orthopartition $\mathcal{O}_i^a$ we can compute its entropy according to Equation (2.5) and select the attribute-split value combination resulting in the minimal average, w.r.t all values i of attribute a, entropy value. This process is repeated until the leaves are reached (that is, a termination criterion has been satisfied) and each leaf L corresponds to a specific orthopartition $\mathcal{O}_L = \{O_1^L, ...O_{|Y|}^L\}$. This orthopartition can then be used to assign a class to new instances by selecting the class according to the following Equation, which selects the class having highest probability:

$$class(\mathcal{O}_L) = max_{y \in Y} \frac{|P_y^L| + \sum_{x \in Bnd_y^L} \frac{1}{|\{y' \in Y : x \in Bnd_{y'}^L\}|}}{|L|} \tag{3.11}$$

3.3. **Random Forests.** In [28] the authors also defined an ensemble learning procedure, inspired by Random Forests. Basically, the learning process, as in standard Random Forest learning, first induces a set of n TWDT estimators, which we denote as $T_1, ..., T_n$. Each of these TWDT estimators can be viewed as an orthopartition $\mathcal{O}_i = \{\langle P_{y_1}, N_{y_1}\rangle, ...\langle P_{y_k}, N_{y_k}\rangle\}$ on the set of instances X, which assigns a set of labels $T_i(x) \subseteq Y$ to each instance $x \in X$ according to one of the approaches presented in Section 3.2.

Let $x \in X$ be a new instance to classify, then the ensemble of trees $T_1, ..., T_n$ determines the following *basic belief assignment* (BBA), in the sense of *evidence theory* [30]:

$$m(S) = \frac{|\{T_i | T_i(x) = S\}|}{n}. \tag{3.12}$$

This BBA could then be transformed to a probability distribution using the *pignistic transformation* [31] $p(y_j) = \sum_{S \ni y_j} \frac{m(S)}{|S|}$, obtaining a probabilistic classifier to which the decision procedure described in Equation (3.5) could be applied.

This procedure can be with respect to two different aspects, defining both a new ensemble combination strategy and a new decision procedure:

(1) First of all, as regards the combination of the different tree predictions, the trees are used to construct a single BBA by means of a naive counting measure technique. Another approach is to consider each tree as a source of information, interpreting their answers as their encoding of the belief $x \in T_i(x) = Z$. It is then natural to view the question of how to aggregate the votes of the Decision Trees under an *information fusion* perspective [32]. In particular we consider each decision tree as providing a different *simple* bba m_i s.t. $m_i(T_i(x)) = 1 - s, m_i(Y) = s$ and $m_i(Z) = 0$ for all other $Z \subseteq Y$, where $s \in (0, 1)$. We then obtain the aggregated mass function using the combination rule defined by Smets in Transferable Belief Model [31]

$$m_x(Z) = \bigotimes m_x^i(Z) = \sum_{A_1 \cap ... \cap A_n = Z} m_x^1(A_1) \cdot ... \cdot m_x^n(A_n) \tag{3.13}$$

(2) With respect to the decision criterion, after having computed the aggregate bba as explained above, we can obtain a set-valued aggregate decision by using the *interval dominance* [33] criterion (thus, without performing the pignistic transformation of m_x). For each $y \in Y$ we compute $I(y) = [Bel(y), Pl(y)] = [m_x(y), \sum_{y \in Z} m_x(Z)]$ and we select as the final solution the set $\{y \in Y | \nexists y' \in Y Bel(y) \geq Pl(y')\}$.

Notably, the proposed aggregation algorithm being based on Smets' combination rule, this approach does not require a re-normalization of the masses in order to assign zero mass to the empty set. Thus, the Three-Way Random Forest classifier can express a complete abstention, i.e., completely refuse to provide any answer. If we denote as $RF(x)$ the decision expressed by the Random Forest algorithm then we say that $RF(x) =?$ if

$$\forall y \in Y, \quad m(\emptyset) \geq Pl(y) \tag{3.14}$$

that is, the empty set is not interval dominated. This additional condition is inspired by the fact that in TBM the frame of discernment is usually thought as not necessarily exhaustive and, in fact, the empty set is usually thought as encoding the belief that the true value is outside the considered domain.

Example 4. *Consider an instance x and a Random Forest RF composed of three trees T_1, T_2, T_3 s.t. $T_1(x) = \{1,2,3\}$, $T_2(x) = \{1,4\}$ and $T_3(x) = \{1,2,4\}$ with $s = 0.1$. Then m_x is defined as follows $m_x(\{1,2\}) = m_x(\{1,4\}) = 0.081$, $m_x(Y) = 0.001$, $m_x(\{1,2,3\}) = m_x(\{1,2,4\}) = 0.009$ and $m_x(\{1\}) = 0.729$. Thus $Bel(2) = Bel(3) = Bel(4) = 0$, $Pl(3) = 0.01$, $Pl(2) = Pl(4) = 0.1$, $Bel(1) = 0.729$ and $Pl(1) = 1$. Consequently 1 is the single non-dominated alternative and thus the results is $RF(x) = 1$.*

3.4. **Optimization–based Learning.** Several ML approaches (e.g. logistic regression, support vector machines, deep learning, ...) are based on a mathematical optimization framework, in which the learning process consists in optimizing the value of a loss function with respect to the parameters of the algorithm. While general *global search* techniques could be used towards this purpose, the *gradient descent* algorithm and its variations, can be used to more efficiently solve the optimization problem when the loss can be represented as an (at least) sub–differentiable function, also providing convergence and optimality conditions when the objective function satisfies certain mathematical properties, e.g., *convexity*, *smoothness*, *Lipschitz continuity*. It is easy to note that the decision rule described in Equation (3.4), that could be seen as a generalized version of the standard *0–1 loss*, is not convex nor smooth:

Theorem 2. [28]*The loss function determined by the decision procedure described in Section 3 is not convex and not smooth.*

Proof. Let $D(x^i) = \begin{cases} Z* & \exists Z^* \text{which solves Eq. (3.5)}, \\ \hat{y^i} & otherwise \end{cases}$

Then the loss of algorithm A w.r.t to instance x is $L(x) = \begin{cases} 0 & C(x) = D(x) \\ \tau & C(x) \in D(x) \\ \epsilon & otherwise \end{cases}$

Clearly, $L(x)$ is not convex. Also, given that the function is not even differentiable, is evidently non–smooth. □

In [28], a convex upper bound of the loss function is defined as a piece–wise linear approximation also providing a one-vs-one scheme, based on evidence theory, for extending the above introduced loss function to multiclass problems.

A limitation of this approach is that it is not amenable to using the *gradient descent* algorithm or *interior point* [34] methods, in fact the approximate loss $L(w)$ is not smooth, nor even differentiable, while it admits a sub–gradient and thus can be solved using *sub–gradient methods* [34], which, however, are less computationally efficient. Another limitation is that the loss function is not directly applicable to multi–class problems, requiring the usage of a *one–vs–one* scheme whose complexity, however, is quadratic in $|Y|$.

We propose a new convex and natively multi-class formulation, inspired by the work of Berrada et al. on *top-k classification* [35] and the definition of *ϵ-ambiguity.* Let A be a scoring classifier, i.e., a classifier that, given an instance $x \in S$, returns a set of scores $A_y(x)$, one for each $y \in Y$. Let $\epsilon \in \mathbb{R}^+$ be a confidence threshold; $A(x)^* = max_{y \in Y} A_y(x)$, with y^* the corresponding class label and $A(x)_{[\epsilon]} = \{y \in$

$Y | A_{y^*}(x) - A_y(x) < \epsilon\}$. Evidently if $t(x) \in A(x)_{[\epsilon]}$ then, according to the set confidence threshold, $t(x)$ is indistinguishable from y^* and thus we should assign low loss values to these cases. A direct generalization of Crammer and Singer formulation of the hinge loss can be given as follows:

$$L_\epsilon(A(x), y) = max\{A(x)^m_{[\epsilon]\setminus y} + \epsilon - A_y(x), A(x)^*_{\setminus y} + 1 - A_y(x)\} \tag{3.15}$$

where $A(x)_{\setminus y}$ denotes the values of $A(x)$ when restricted to $Y \setminus \{y\}$ and $A(x)^m_{[\epsilon]\setminus y} = minA(x)_{[\epsilon]\setminus y}$.

Example 5. *Let $Y = \{1, 2, 3, 4, 5\}$ and A be a classifier such that, for a given instance x we have $A(x) = \langle 0.2, 0.3, 0.15, 0.1, 0.25\rangle$. Then, if we set $\epsilon = 0.1$ and $y = 2$ we have that $A(x)_{[\epsilon]\setminus 2} = \{1, 5\}$ and, hence, $A(x)^m_{[\epsilon]\setminus y} = 0.2$. Thus $L_\epsilon(A(x), 2) = max\{0.2 + 0.1 - 0.3, 0.25 + 1 - 0.3\} = max\{0, 0.05\} = 0.05$. In fact, the difference between $A_2(x)$ and $A_5(x)$ is less than the margin 1, thus the loss should penalize this score in order to increase the margin.*

It is easy to verify that if $A(x)$ can be expressed as a convex function on the parameters, e.g. for SVMs, or logistic regression if the scores are obtained via the softmax transformation of the underlying linear estimator, then the loss is also convex.

Theorem 3. *L_ϵ is convex.*

Proof. Both terms inside the *max* are linear, thus also convex, furthermore the point-wise maximum of convex functions is also convex. □

We can also show that this loss is well-behaved, in fact:

- When $A_y(x) = A(x)^*$ then the left part of the maximization assumes its minimal value, which is 0. The same holds for the right part, in this case the actual value depends on both the margin between $A_y(x)$ and $A(x)^*$ and the value of ϵ.
- When A_y is outside $A(x)^*_{[\epsilon]}$ then the right part of the loss becomes dominating and thus drives the optimization towards including the true label y inside the ϵ-ambiguity subset.

From this formulation we can also define a smoothed, and, thus, differentiable, version following the *temperature-based smoothing* proposed in [35], obtaining:

$$L^s_\epsilon(A(x), y) = \tau \cdot ln(\sum_{\mathbf{y} \in Y^k} e^{\delta(y,\mathbf{y}) + \sum_{i=1}^k A(x)_{[\epsilon]\setminus y}}) - \tau \cdot ln(\sum_{\mathbf{y} \in Y^k_y} e^{\sum_{i=1}^k A(x)^k}) \tag{3.16}$$

where $\tau > 0$ is a smoothing parameter, $\delta(y, \mathbf{y})$ is defined as

$$\delta(y, \mathbf{y}) = \begin{cases} \epsilon & y \in A(x)^*_{[\epsilon]} \\ 1 + A(x)^*_{\setminus y} & otherwise \end{cases}$$

Y^k is the set of k-tuples of classes in Y, and Y^k_y is the set of k-tuples containing the true class label y. Note that, while the smoothed loss L^s_ϵ is convex and smooth, and thus we can ensure convergence to a global optimum using standard out-of-the-box optimizers, it is expensive from a time complexity point of view: indeed, in order to compute the value of the loss, a summation over all possible k-tuples in Y is required and thus computing the value of the loss requires $O(|Y|!)$ evaluations. A trade-off should then be considered among ease of convergence, for which L^s_ϵ is favored, and computational efficiency, for which L_ϵ is favored, also taking in consideration the possibility of using approximate evaluation techniques, e.g., Monte Carlo based approaches, that could be used to speed-up the computation of L^s_ϵ and the size of Y.

4. Three–way input

While Three–Way Output denotes a single phenomenon, i.e., a classifier emitting a set–valued classification, with Three–Way Input (TWI) we denote two different phenomena leading to a form of *r-ambiguity*, i.e., the presence of set–valued values in the training set (which is the input to the learning process):

(1) The target attribute d is set–valued, this setting is also called *learning from imprecise/partial labels* [20, 19]: a set-valued classification can be seen as a partial abstention of the labeler in establishing a precise and clear-cut decision;

(2) The training set is a multi–observer decision table (i.e. $|D| > 1$) which is converted to a standard decision table in which the target attribute may be set–valued, in order to preserve the disagreement among the observers.

In both cases the goal of the learning task is to train a classifier that is able to *disambiguate*, that is to reduce as much as possible the original *r-ambiguity.* Seminal work in this field has been initiated by [19] that introduced the concept of learning from ambiguous labels, now more commonly called partial labels, that sparked some interest in the research community [36] with several approaches focusing on optimization-based (see e.g. [37, 38]), instance-based (e.g. k-nearest neighbors) approaches [39] or hybrid approaches [40]. On the other hand the study of multi-observer decision tables has been largely side-stepped in the ML community, where the common approach amounts to the selection of, for each instance, the most common label (in classification settings) or the mean value (in regression settings) [41, 42, 43]. We believe that more generalised *transformations* could be employed to preserve the multi-faceted and information-rich nature of multi-observer labelings [44]. However, we leave the full development of the multi-observer setting to a future work and here, we will focus on the first case.

In particular, in Sections 4.1 and 4.2 we will describe learning algorithms for the TWI setting, by first presenting methods proposed in the literature and then presenting our newly defined algorithms, both in optimization-based learning and for decision trees and random forests. These techniques, as in the TWO learning setting, employ orthopartitions as uncertainty representation framework to model uncertain assignments of objects to classes, and methods rooted in evidence theory, three-way decision and interval arithmetics to enable learning and inference on these structures.

4.1. **Optimization–based Learning.** This learning setting has been studied under the names of *learning from ambiguous labels* [19], *learning from partial labels* [20, 45] or also *superset label learning* in the literature. In these works, a standard optimization–based learning framework is described as follows. Let $A_y(x)$ be the score assigned by algorithm A to class y for instance x and $L(\cdot) : \mathbb{R} \mapsto \mathbb{R}^+$ be a loss function. If $y^* \in Y$ is defined as $y^* = argmax_y A_y(x)$, then we can define the following three strategies :

$$L_{max}(A(x), S) = L(max_{y \in S} A_y(x)) + \sum_{y \notin S} L(-A_y(x)) \tag{4.1}$$

$$L_{\psi}(A(x), S) = L(\frac{1}{|S|} \sum_{y \in S} A_y(x)) + \sum_{y \notin S} L(-A_y(x)) \tag{4.2}$$

$$L_{avg}(A(x), S) = \frac{1}{|S|} \sum_{y \in S} L(A_y(x)) + \sum_{y \notin S} L(-A_y(x)) \tag{4.3}$$

These formulations are based on different assumptions: L_{max} and L_{ψ} assume that the highest scoring alternative is likely to be the correct one (and thus, can be considered as *optimistic* loss functions) and thus typically tend to maximize the score of one alternative. On the other hand L_{avg} tends to favor distributions $A(x)$ weighting the alternatives according to the available information, by optimizing the average loss value among the possible alternative labels. It is also to notice that the following result holds:

Proposition 1. *If L is convex and decreasing then*

$$L_{max}(A(x), S) \leq L_{\psi}(A(x), S) \leq L_{avg}(A(x), S)$$

Proof. The result directly follows from Jensen inequality. □

This framework, however, does not take in consideration the uncertainty and ambiguity which is intrinsic in this learning setting, on the contrary, this is eliminated by making some form of assumptions, e.g. *optimism*, which is especially prevalent in the L_{max} formulation. In fact, it is easy to see that any case of r-ambiguity, and hence every Three-way In learning problem, naturally defines a distribution over the possible values of the loss function, determined by the fact that different possible consistent classifications can be obtained from the original ambiguous one.

Such an estimate of the uncertainty range on the loss function true value can be given as:

$$L_{int}(A(x), S) = [min_{y \in S} L(A(x), y), max_{y \in S} L(A(x), y)] \tag{4.4}$$

Let $w \in R^d$ be a parameter vector representing our classifier. For instance, in a neural network w is the set of the weights of all the connections. The interval–valued loss $L_{int}(A(x), S)$ naturally induces a pair of parametrizations $\langle w_{min}, w_{max} \rangle$, that are obtained optimizing for, respectively, the minimum and the maximum values of the loss function.

In order to explain how to use this interval–valued information, we consider the case of binary linear classification. Let x be an instance of the training set and assume, without loss of generality, $d(x) = 1$. We should find which of w_{min}, w_{max} results, respectively, in the minimum and maximum loss value for $L(A(x), 1)$. Evidently, $w_M = max_{i \in \{min,max\}} w_i \cdot x$ (resp., w_m is the opposite one) will result in the minimum L_l (resp., maximum L_u) value of the loss function. Thus, the optimizer should then modify the parameters w_M (resp., w_m) so to optimize for the value of L_l (resp., L_u).

If, on the other hand, $l(x) = \{0, 1\}$ we can, for each of the two labels, determine the interval–valued losses $L(y = 1, w) = [L^1_l, L^1_u], L(y = 0, w) = [L^0_l, L^0_u]$ and we can establish the value of $L(y = \{0, 1\}, w) = [min\{L^1_l, L^0_l\}, max\{L^1_u, L^0_u\}]$. This formulation can be extended to a multi–class model. However, in this case, in order to determine the interval–valued loss for an instance x, solving a multi–objective optimization problem (MOOP) is required for determining the values of w resulting in the extremes of the loss value distribution. Thus, the learning framework consists of four steps:

(1) Starting from initial random parameters, induce the interval–valued loss defined in Equation (4.4);
(2) Update the parameters in order to optimize for the loss, thus determining a pair of parametrizations;
(3) Propagate forward the pair of parametrizations finding the corresponding minimum and maximum values of the loss function given the current parameter values, solving a multi–objective optimization problem;
(4) Repeat steps 2 and 3 until convergence or out of time.

A *stochastic gradient descent*-inspired version of this procedure, that we termed *Stochastic Interval-Valued Optimization* (SIVO), is described in Algorithm 1.

```
Data: Dataset DT
Result: optimal interval-valued parameters $[w^{min}, w^{max}]$
Set $w$ to an initial random candidate;
Select instance $x \in DT$ with $|d(x)| > 1$;
$L_{int}(A(x), S) = [min_{y \in S} L(A(x), y), max_{y \in S} L(A(x), y)]$;
Update $\langle w^{min}, w^{max} \rangle$ in order to optimize for $L_{int}$;
while not converged or out of budget do
    Select instance $x \in DT$;
    Find $w_M \in \langle w^{min}, w^{max} \rangle$ s.t. $max_{y \in S} L_{w_M}(A(x), y)$ is maximal;
    Find $w_m \in \langle w^{min}, w^{max} \rangle$ s.t. $max_{y \in S} L_{w_m}(A(x), y)$ is minimal;
    $L_{int}(A(x), S) = [min_{y \in S} L_{w_m}(A(x), y), max_{y \in S} L_{w_M}(A(x), y)]$;
    Update $\langle w^{min}, w^{max} \rangle$ in order to optimize for $L_{int}$;
end
```

Algorithm 1: Stochastic Interval-Valued Optimization (SIVO) Algorithm

Note that to extend this approach to multi–layered non–linear classifiers (e.g. deep learning algorithms, that can be understood as generalizations of logistic regression) requires, in the forward phase, to solve multiple MOOPs layer by layer.

Interestingly, the SIVO algorithm also allows us to obtain a Three-way Out classifier directly, without further applications of the techniques described in Section 3. In fact, considering the final parametrizations $\langle w^{min}, w^{max} \rangle$ and a new instance x to be classified, for each $y \in Y$ we can compute $L_{int}(A(x), y) = [L_l^y, L_u^y]$.

Definition 9. *We say that a classification y* dominates *y' if*

$$L_u^y \leq L_l^{y'} \quad or \quad (L_l^y = L_l^{y'} = 0 \quad and \quad L_u^y \leq L_u^{y'})$$

Then, the output classification for instance x is $\{y | \nexists y' \in Y,\ y' \text{ dominates } y\}$, that is, the set of non-dominated possible classifications.

Example 6. *Let $Y = \{1, -1\}$, $x = \langle 1, 2, -3, 1 \rangle$ with $d(x) = \{-1, 1\}$ and consider a linear classifier A with initial parameter configuration $w = \langle 1, 1, 1, 1 \rangle$ and loss* *function $L = max\{0, 1 - y(w \cdot x)\}$. Then $L_{int}(A(x), \{-1, 1\}) = [0, 2]$ and, the new parameter vector becomes $w^{min} = \langle 0, -1, 4, 0\rangle$ and $w_{max} = \langle 1, 1, 1, 1\rangle$. Then, if another instance $x' = x$ would be given as input to A we could distinguish three cases:*

(1) If $d(x') = -1$ then $L_{int}(A(x'), -1) = [0, 2]$ and the new parametrization following the update would become $w^{min} = w^{max} = \langle 0, -1, 4, 0\rangle$;
(2) If $d(x') = 1$ then $L_{int}(A(x'), -1) = [0, 15]$ and the new parametrization following the update would become $w^{min} = w^{max} = \langle 1, 1, 1, 1\rangle$;
(3) If $d(x') = \{0, 1\}$ then $L_{int}(A(x'), \{0, 1\}) = [0, 15]$ and the parametrization would not change.

Thus, if the optimization would be stopped after the first iteration, then, the resulting classification would be -1 (since it dominates the alternative 1).

4.2. **Decision Trees and Random Forests.** In a standard Decision Tree learning algorithm, we consider, at each internal node N_i and for each attribute a, a possible split point p_a (the technique to determine the split point depends on the specific adopted learning algorithm) and evaluate that split by computing its induced mutual information:

$$\begin{aligned} L(p_a, N_i) &= P(v_a \geq p_a|N_i) \cdot \sum_{y \in Y} P(y|v_a \geq p_a, N_i) \cdot log_2 P(y|v_a \geq p_a, N_i) \\ &+ P(v_a < p_a|N_i) \cdot \sum_{y \in Y} P(y|v_a < p_a, N_i) \cdot log_2 P(y|v_a < p_a, N_i) \\ &= P(v_a \geq p_a|N_i) \cdot H(S|v_a \geq p_a, N_i) \\ &+ P(v_a < p_a|N_i) \cdot H(S|v_a < p_a, N_i) \end{aligned} \tag{4.5}$$

Then, we select the attribute a and the split point p_a corresponding to the minimal value of $L(p_a, N_i)$. In a leaf node N_l, a decision label $d(N_l)$ is selected, according to the Bayes optimal decision rule:

$$d(N_l) = argmax_{y \in Y} P(y|N_l) \tag{4.6}$$

These formulas are not directly applicable in the Three–way In setting, since the labeling of the instances does not form a partition but, more generally, an orthopartition. This means, however, that we can use the definitions of entropy given for orthopartitions in Section 2. We can, thus, redefine the value of $L(P_a, N_i)$ by using H_{bet}, comparing attributes and split points in the same way as for classical Decision Trees.

As regards the leaves of the tree, we observe that each such leaf N_l defines a *basic belief assignment* (bba) m:

$$m_{N_l}(Z \subseteq Y) = \frac{|\{x \in S | t(x) = Z \wedge x \in N_l\}|}{|N_l|} \tag{4.7}$$

from which we can compute the corresponding *pignistic probability*:

$$p_{bet}(y) = \sum_{y \in Z} \frac{m(Z)}{|Z|} = \frac{1}{|N_l|} \cdot (|P_y| + \sum_{x \in Bnd_y} \frac{1}{Bnd(x)}) \tag{4.8}$$

This line of reasoning can be directly extended from Decision Trees to Random Forests. Basically we induce the n Decision Trees $T_1, ..., T_n$ to form the forest, then, each such tree T_i defines a bba m_i. From these we can obtain a global bba m_{RF} using the rule of combination:

$$m_{RF}(Z) = (\bigotimes_i m_i)(Z) = \sum_{A_1 \cap \ldots \cap A_n = Z} m_1(A_1) \cdot \ldots \cdot m_n(A_n) \tag{4.9}$$

from which we can again compute the pignistic probability:

$$p_{bet}(y) = \sum_{y \in Z} \frac{m_{RF}(Z)}{|Z|} = \sum_{y \in Z} \sum_{A_1 \cap \ldots \cap A_n = Z} \frac{m_1(A_1) \cdot \ldots \cdot m_n(A_n)}{|Z|} \tag{4.10}$$

Furthermore, as also described for the SIVO algorithm in Section 4.1, we can also directly obtain a three-way classifier from Three-way In Decision Trees or Random Forests by applying the interval dominance criterion on the basic belief assignment m_{N_l} or m_{RF}.

5. Experimental Validation and Discussion

In order to assess the validity and efficacy of the proposed algorithms, in both the Three-way Out and Three-way In learning settings, we performed two sets of experimental validations: in Section 5.1 we report the experimental setting and obtained results in the Three-way Out case, while in Section 5.2 we report the same information for the Three-way In case.

5.1. **Experiments and Results: Three-way Out learning.** As regards the evaluation of Three-way Out classification algorithm, our experiments represent an expansion of the experiments reported in [28]. Thus, we considered both classical algorithms: *k–nearest neighbors* (KNN), *logistic regression* (LR), *Naive Bayes* (NB), *SVM*s, *random forest* (RF) and their respective three-way versions (prefixed with TW in the following) considering both the decision-theoretic formulation (the comparison between these and the classical algorithms was initially reported in [28]) and the ϵ-ambiguity based one. We considered also the algorithm for random forests presented in Section 3.2, DIFID-TWRF in the following, using distance-based split attribute selection, the information fusion base classifier combination criteria and the interval dominance decision criteria. Finally, the last algorithm is a linear classifier based on the L_ϵ loss function defined in 3.4, named L_ϵ-TWLC in the following.

The experiments have been performed on a set of standard datasets from the UCI repository: *Iris*, *Wine*, *Digits*, *Breast cancer*, *Yeast*, *Olivetti faces* and a real-world medical datasets provided by IRCCS Galeazzi, one of the major research hospitals in Italy. It will be named *SF12* in the following and its target is to predict eventual worsening of mental health of patients. See Table 2 for information about all the datasets.

For each algorithm and dataset, we performed hyper-parameter selection and computed the accuracy values averaged over a 5-fold cross-validation. As accuracy we considered the average accuracy acc_A over all possible assignments of objects in the boundaries and also the 95% confidence intervals around the mean is reported in the following.

For the decision-theoretic three-way out algorithms we selected $\epsilon = 1$ and $\alpha(Z) = \frac{|Z|}{|Y|}$, while for the ϵ-ambiguity three-way out algorithms, the L_ϵ LR algorithm and the distance-based implementation of Random Forests described in 3.2 we set $\epsilon = 0.3$. The hyper-parameters settings and the references to the tested algorithms are

Dataset	# instances	# attributes	# classes
Iris	150	4	3
Wine	178	13	3
Digits	1797	64	10
Breast cancer	569	30	2
Olivetti faces	400	4096	40
Yeast	1484	8	10
SF12	462	10	2

Table 2. Number of instances, attributes and classes for the tested datasets.

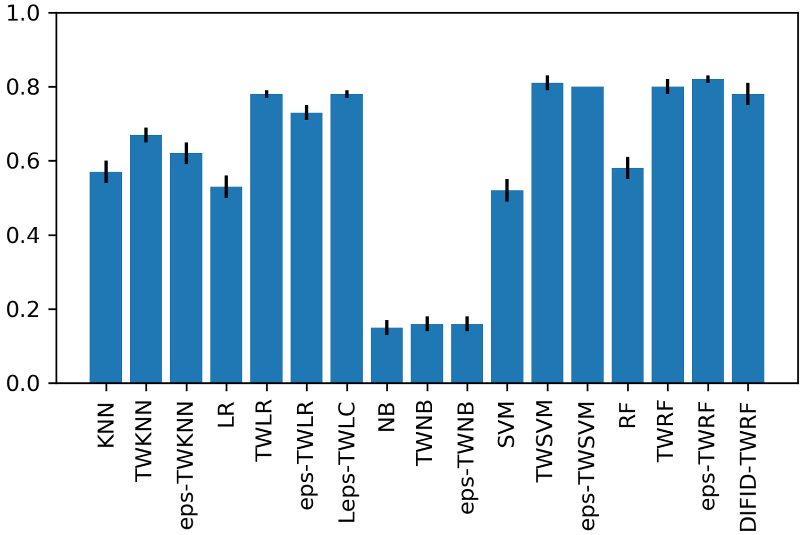

Figure 5. Accuracies and 95% confidence intervals for the standard and three-way out classification algorithms on the Yeast dataset.

reported in Table 3. The results of these experiments, some of which have been previously reported in [28], are reported in Table 4 and the results for the Yeast, Wine and SF12 datasets are also reported in Figures 5, 6 and 7.

In order to compare the different algorithms, and assess the statistical significance of the differences in performances, we employed the Friedman test, a standard hypothesis testing procedure to assess if any alternative algorithm consistently performs better than others. More precisely, we used the post-hoc procedure [46] for pair-wise comparisons between pairs of algorithms and Li's correction [47] for multiple testing correction, in order to not over-estimate the p-values of the tests. The average ranks registered for the 10 best performing algorithms are reported in Table 5.

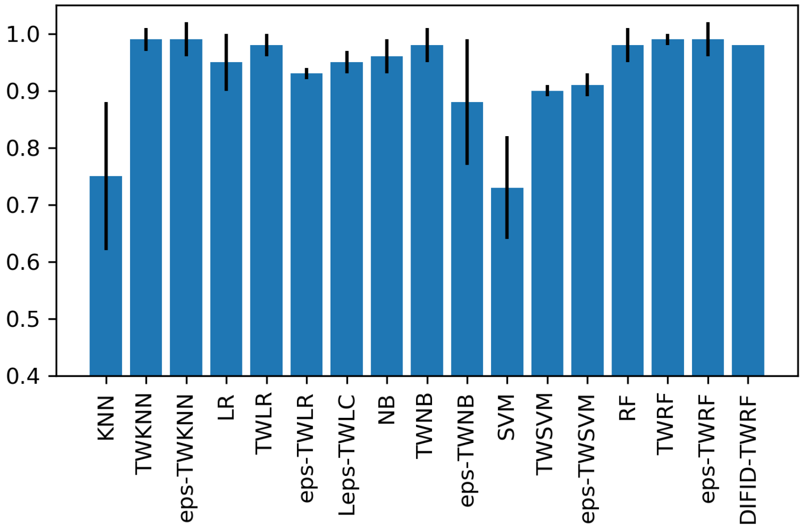

FIGURE 6. Accuracies and 95% confidence intervals for the standard and three-way out classification algorithms on the Wine dataset. The y axis starts from 0.4 in order to better highlight the differences among the algorithms.

Algorithm	Ref. inside article	Hyper-parameters
TW	Eq. (3.5)	$\epsilon = 1, \alpha = \frac{\lvert Z\rvert}{\lvert Y\rvert}$
ϵ-TW	Eq. (3.2)	$\epsilon = 0.3$
L_ϵ-LR	Sec. 3.4	$\epsilon = 0.3$
DIFID-TWRF	Sec. 3.2	$\epsilon = 0.3, s = 0.25$

TABLE 3. References and hyper-parameters settings for tested algorithms

As can be seen from these tables the best performing algorithm was the TWRF algorithm, and all three-way out versions of Random Forests were among the top three-performing algorithms and separated by a small margin, i.e., no statistically significant difference was found between their performances at the standard threshold $\alpha = 0.05$.

More generally, for each algorithm, the decision-theoretic three-way version performed better than both the classical and ϵ-ambiguity versions. The comparisons among RF-TWRF, SVM-TWSVM and LR-TWLR were statistically significant at $\alpha = 0.05$ while no statistically significant differences were found when comparing decision-theoretic and ϵ-ambiguity algorithms. Generally the ϵ-ambiguity version compared favorably with the classical version, producing better performing algorithms in the case of LR, SVM and RF (both SVM and RF comparisons were statistically significant) and the only two cases for which the opposite happened were not statistically significant. Also, for both the ad-hoc developed algorithms,

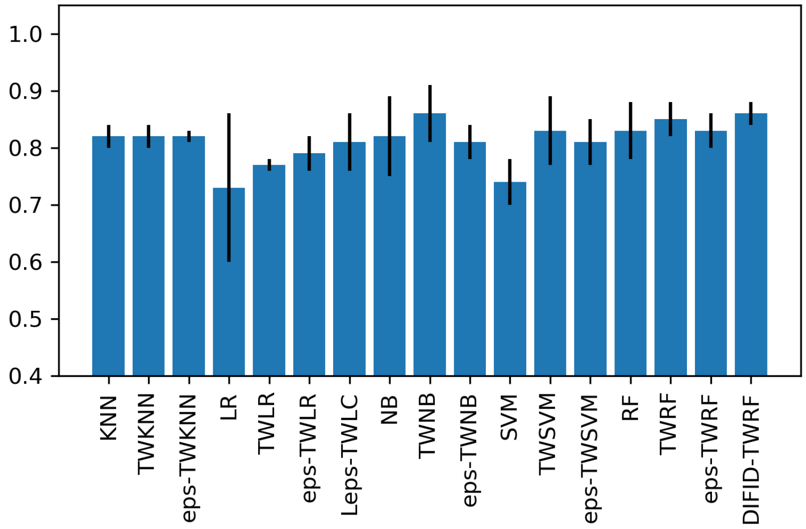

Figure 7. Accuracies and 95% confidence intervals for the standard and three-way out classification algorithms on the SF12 dataset. The y axis starts from 0.4 in order to better highlight the differences among the algorithms

Algorithm	Iris	Wine	Breast	Digits	Yeast	Faces	SF12
KNN	0.98 ± 0.03	0.75 ± 0.13	0.93 ± 0.04	0.98 ± 0.03	0.57 ± 0.03	0.90 ± 0.16	0.82 ± 0.02
TW-KNN	1.00 ± 0.00	0.99 ± 0.02	0.99 ± 0.01	0.90 ± 0.00	0.67 ± 0.02	0.89 ± 0.01	0.82 ± 0.01
ϵ-TW-KNN	0.96 ± 0.04	0.99 ± 0.03	0.96 ± 0.02	0.95 ± 0.05	0.62 ± 0.03	0.81 ± 0.05	0.82 ± 0.01
LR	0.95 ± 0.06	0.95 ± 0.05	0.95 ± 0.02	0.93 ± 0.04	0.53 ± 0.03	0.96 ± 0.03	0.73 ± 0.13
TW-LR	0.96 ± 0.01	0.98 ± 0.02	0.98 ± 0.01	0.96 ± 0.02	0.78 ± 0.01	0.98 ± 0.01	0.77 ± 0.01
ϵ-TW-LR	0.94 ± 0.00	0.93 ± 0.01	0.96 ± 0.02	0.96 ± 0.01	0.73 ± 0.02	0.87 ± 0.12	0.79 ± 0.03
L_ϵ-TW-LC	0.97 ± 0.01	0.95 ± 0.02	0.98 ± 0.01	0.96 ± 0.01	0.78 ± 0.01	0.99 ± 0.02	0.81 ± 0.05
NB	0.95 ± 0.04	0.96 ± 0.03	0.94 ± 0.03	0.81 ± 0.06	0.15 ± 0.02	0.82 ± 0.03	0.82 ± 0.07
TW-NB	0.97 ± 0.03	0.98 ± 0.03	0.95 ± 0.03	0.83 ± 0.05	0.16 ± 0.02	0.84 ± 0.02	0.86 ± 0.05
ϵ-TW-NB	0.91 ± 0.05	0.88 ± 0.11	0.95 ± 0.02	0.81 ± 0.07	0.16 ± 0.02	0.86 ± 0.03	0.81 ± 0.03
SVM	0.98 ± 0.03	0.73 ± 0.09	0.94 ± 0.02	0.97 ± 0.02	0.52 ± 0.03	0.79 ± 0.05	0.74 ± 0.04
TW-SVM	0.98 ± 0.01	0.90 ± 0.01	0.96 ± 0.01	1.00 ± 0.00	0.81 ± 0.02	0.87 ± 0.04	0.83 ± 0.06
ϵ-TW-SVM	0.94 ± 0.02	0.91 ± 0.02	0.93 ± 0.05	0.97 ± 0.02	0.80 ± 0.00	0.87 ± 0.02	0.81 ± 0.04
RF	0.97 ± 0.03	0.98 ± 0.03	0.96 ± 0.02	0.94 ± 0.02	0.58 ± 0.03	0.93 ± 0.02	0.83 ± 0.05
TW-RF	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.80 ± 0.02	0.98 ± 0.01	0.85 ± 0.03
ϵ-TW-RF	0.98 ± 0.02	0.99 ± 0.03	0.97 ± 0.02	0.96 ± 0.02	0.82 ± 0.01	0.98 ± 0.01	0.83 ± 0.03
DIFID-TW-RF	0.99 ± 0.01	0.98 ± 0.00	1.00 ± 0.01	0.98 ± 0.02	0.78 ± 0.03	0.97 ± 0.02	0.86 ± 0.02

Table 4. Measured 95% confidence intervals, centered around the mean accuracy, for the considered datasets and algorithms.

Alg.	TWRF	DIFID-TWRF	ϵ-TWRF	L_ϵ-TWLC	TWLR	TWSVM	TWKNN	RF	KNN/ϵ-TWLR/ϵ-TWSVM
Rank	2.14	2.28	2.71	3.71	4.00	4.14	4.42	4.86	5.86

Table 5. Average ranks of the top 10 performing algorithms.

L_ϵ-TWLC and DIFID-TWRF, the obtained performances were statistically significantly better than the respective classical algorithms, although no statistically significant difference was found among these algorithms and the respective decision-theoretic three-way out algorithms.

From these observations we can conclude that three-way out algorithms offer a trade-off among accuracy (and reliability) and *coverage*, i.e. the points that are classified: they sacrifice coverage in order to obtain predictions that are more accurate and more reliable because they reflect the uncertainty intrinsic in the training data. It can also be observed that, compared to decision-theoretic three-way algorithms, the ϵ-ambiguity implementations provide comparable performance but with less parameters to set and, if the most general formulation of the decision-theoretic criterion is employed, increased computational efficiency. Indeed, time complexity is log-linear instead of exponential in the number of classes. Similar observations can be made with respect to the DIFID-TWRF and L_ϵ-TWLC algorithms.

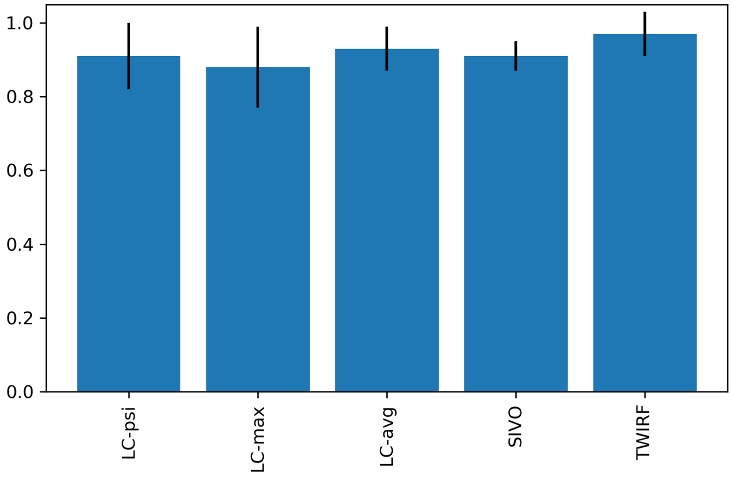

FIGURE 8. Accuracies and 95% confidence intervals for the three-way in classification algorithms on the Iris dataset.

5.2. **Experiments and Results: Three-way In learning.** As regards the Three-way In learning setting we compared five different algorithms: four variants of linear classifiers based on the loss functions L_ψ, L_{avg}, L_{max} and the SIVO algorithm presented in Section 4.1, and the random forest algorithm described in Section 4.2, using the H_{bet} entropy definition, called TWI-RF in the following. In order to evaluate the algorithm we considered a set of 4 different standard datasets from the UCI repository: *Iris*, *Breast cancer*, *Wine*, *Digits* and two synthetic datasets:

- A 5 class isotropic Gaussian classification problem, with 10^5 instances and $\sigma = 10$. This dataset was created in order to have an adversarial dataset for which the true classification would be difficult to recover, due to the large σ value that results in the classes to have large overlaps.
- A 2 class circle classification problem, with 10^6 instances and $\sigma = 0.1$.

For all these datasets we perturbed the original labels in order to evaluate the ability of the Three-way In algorithms to recover the original labels. The perturbation was generated as follows:

- For each instance x, collect the 10-nearest neighbors $N_{10}(x)$;
- Assign to instance x the set of labels $\{d(x')|x' \in N_{10}(x)\}$.

In order to evaluate the algorithms we performed a 5-fold cross-validation, training the algorithms on the perturbed data obtained via the above described procedure and then testing the algorithms against the original labels of the validation instances, in order to assess to which degree they were able to recover the correct labelling.

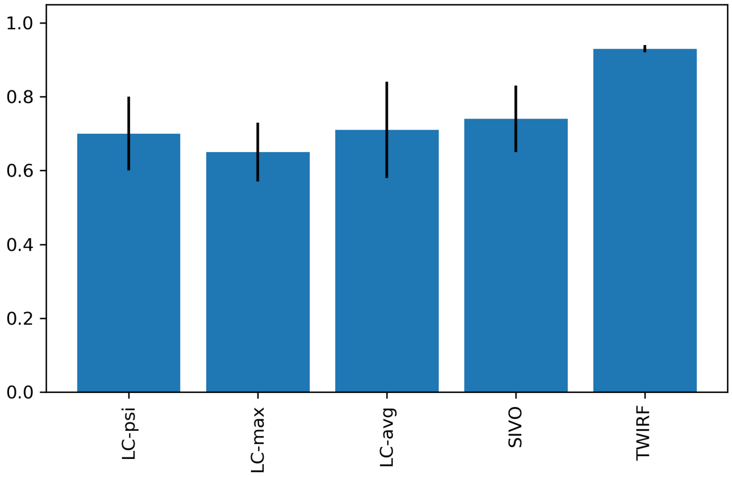

Figure 9. Accuracies and 95% confidence intervals for the three-way in classification algorithms on the Digits dataset.

For each algorithm and dataset we registered the average accuracy over the different folds and the 95% confidence intervals around the mean. These results are illustrated in Table 6, while the results for the Iris, Digits and Circles datasets are shown in Figures 8, 9 and 10.

Algorithm	Iris	Wine	Breast	Digits	Gaussian	Circles
LC_ψ	0.91 ± 0.09	0.83 ± 0.04	0.67 ± 0.14	0.70 ± 0.10	0.64 ± 0.03	0.54 ± 0.04
LC_{max}	0.88 ± 0.11	0.82 ± 0.05	0.63 ± 0.14	0.65 ± 0.08	0.65 ± 0.02	0.53 ± 0.04
LC_{avg}	0.93 ± 0.06	0.86 ± 0.02	0.63 ± 0.13	0.71 ± 0.13	0.61 ± 0.06	0.53 ± 0.03
SIVO	0.91 ± 0.04	0.87 ± 0.04	0.65 ± 0.12	0.74 ± 0.09	0.63 ± 0.04	0.54 ± 0.04
TWI-RF	0.97 ± 0.04	0.87 ± 0.11	0.92 ± 0.03	0.93 ± 0.01	0.68 ± 0.02	0.78 ± 0.01

Table 6. Measured 95% confidence intervals, centered around the mean accuracy, for the considered datasets and algorithms.

As can be seen from the tables the Random Forest-based algorithm provided significantly better performance than all linear classifier-based methods, i.e. the best accuracy even when taking into account the possible oscillations of performance

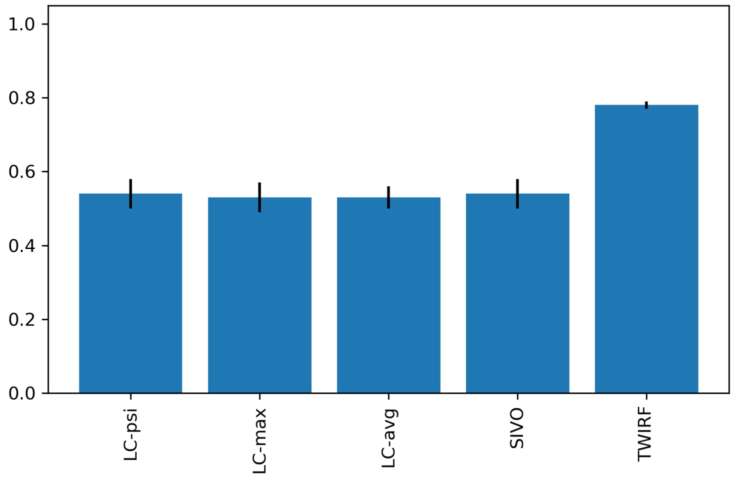

FIGURE 10. Accuracies and 95% confidence intervals for the three-way in classification algorithms on the Circles dataset.

given by the confidence intervals. A possible reason for this behavior is that all the four other models make very strong assumptions, given that they are linear classifier, and thus require, at least approximately, linear separability of data in order to achieve good performance. This is particularly evident in the Circles dataset in which the linearity assumption obviously doesn't hold. On the other hand, in general, Random Forest are complex non-linear classifiers and thus able to better retrieve hidden patterns. Furthermore, the formulation of Random Forests that we defined does not require a retraining for each different possible label and thus, at least in principle, it is more efficient. It is to note, however, that only the difference among TWI-RF and the other algorithms was found to be statistically significant (at $\alpha = 0.05$, and using the hypothesis testing procedure described in Section 5.1), and thus further studies in order to find significant advantages, if any, of any technique should be considered.

6. Conclusions

In this article, we studied the ambiguity occurring in Machine Learning, from a twofold perspective: both as a problem affecting the input of the learning process, and as a potential resource to make the output of classifiers apter for sound human decision making.

In particular, we presented techniques to represent and manage this type of uncertainty in the training data that is fed into the learning algorithm, what we called *Three-way In*), and also techniques to represent ambiguity and uncertainty in the output of a learned model, what we called *Three-way Out.* More specifically, in this work we provided a unified approach, based on orthopartitions, three-way decisions and evidence theory, to address both problems under the same mathematical framework and, possibly, in a unified end-to-end learning process. In so doing, our approach allows to build Machine Learning systems capable to take ambiguous and partial data in their training, as well as to reflect this uncertainty in the classification of new data points for which no sufficient evidence is available.

To this aim, we first provided a general description of the sources and instances in which these types of ambiguity could arise; how they can be represented by using orthopartitions; and how these data representation can be managed by some Machine Learning algorithms, by providing a set of different classification techniques based on Decision Tree and ensemble or optimization-based methodologies.

In order to assess the validity of the proposed techniques and their capability to properly manage ambiguity in both the Three-way In and Three-way Out learning settings, we performed a set of experiments in which we compared the proposed techniques with state-of-the-art approaches, and obtained promising results. These results support the claim that the proper representation and management of ambiguity in the data, instead of hiding or ignoring it, can bring advantages in machine-learning settings.

In particular, we also claim that the proposed techniques are significant in real-world problems, more importantly so in critical tasks in which ambiguity and, more in general, uncertainty is *intrinsic* and to some extent unavoidable, as in medicine and law. In such settings, we recall that multi-faceted and multi-view representations of the data are common, as different raters are often involved in labeling problems [41, 42]: the capability of directly using and conveniently communicating the ambiguity encountered by the algorithm in recommending a class could be critical to deliver reliable Machine Learning-based Decision Support Systems. This also addresses the recent call for explainable, interpretable and accountable methods [48, 49].

Indeed, in our view, abstention in the ML input is a way to trade reliability off with completeness: i.e., to improve the former, we can decrease the latter. On the other hand, abstention in ML output is a way to trade (decision) accuracy with efficiency, as a computational system providing decision makers with unresolved advice implies that these have to look for and consider more evidence, even beyond the available data.

Given these observations, we believe that the following future works should be investigated:

- in multi-rater tasks, the traditional approaches consist of converting a learning problem on a multi-observer decision table to traditional single-decision learning problems, e.g. by taking the majority vote of the raters. This approach, although simpler than applying three-way in strategies, involves an information loss. Therefore, experiments that compare possible advantages of directly using Three-way In strategies should be conducted.
- Extensions of the proposed techniques to more general uncertainty types (e.g. vagueness), and models (e.g. fuzzy or possibility-based approaches) could be of interest in order to address different types of imperfect information or rich data representations that can arise in real-world problems.
- While in this work we considered only uncertainty and ambiguity affecting the target (or decision) variables, these problems could also affect all other predictor variables, e.g. the predictor features could contain missing, set-valued or interval-valued data [22]; therefore, a generalization of

the proposed approaches to these cases could extend the generality and applicability of the proposed framework.

- Finally, a study of the learnability properties (e.g. sample complexity bounds) of the considered learning settings, in the vein of [50], should attract further research efforts.

References

[1] R. C. Fox, Medical uncertainty revisited, SAGE Publications Ltd, London, 2000, Ch. 13, pp. 409–425.

[2] S. Hatch, Uncertainty in medicine, BMJ 357.

[3] F. Cabitza, D. Ciucci, R. Rasoini, A giant with feet of clay: On the validity of the data that feed machine learning in medicine, in: F. Cabitza, C. Batini, M. Magni (Eds.), Organizing for the Digital World, Springer International Publishing, Cham, 2019, pp. 121–136.

[4] Min-Ling Zhang, Zhi-Hua Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering 18 (10) (2006) 1338–1351.

[5] E. Hüllermeier, W. Cheng, Superset learning based on generalized loss minimization, in: A. Appice, P. P. Rodrigues, et al. (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2015, pp. 260–275.

[6] Y. Yao, An outline of a theory of three-way decisions, in: J. Yao, Y. Yang, R. Słowiński, S. Greco, H. Li, S. Mitra, L. Polkowski (Eds.), Rough Sets and Current Trends in Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 1–17.

[7] C. Ferri, J. Hernández-Orallo, Cautious classifiers, in: ROC Analysis in Artificial Intelligence, 1st International Workshop, ROCAI-2004, 2004, pp. 27–36.

[8] Y. Hechtlinger, B. Póczos, L. A. Wasserman, Cautious deep learning, CoRR abs/1805.09460.

[9] Y. Yao, Three-way decision and granular computing, International Journal of Approximate Reasoning 103 (2018) 107 – 123.

[10] J. Pang, X. Guan, J. Liang, B. Wang, P. Song, Multi-attribute group decision-making method based on multi-granulation weights and three-way decisions, International Journal of Approximate Reasoning 117 (2020) 122 – 147. doi:10.1016/j.ijar.2019.11.008.

[11] Y. Li, L. Zhang, Y. Xu, Y. Yao, et al., Enhancing binary classification by modeling uncertain boundary in three-way decisions, IEEE Transactions on Knowledge and Data Engineering 29 (7) (2017) 1438–1451.

[12] X. Yang, T. Li, H. Fujita, D. Liu, A sequential three-way approach to multi-class decision, International Journal of Approximate Reasoning 104 (2019) 108 – 125.

[13] Y. Zhang, D. Miao, J. Wang, Z. Zhang, A cost-sensitive three-way combination technique for ensemble learning in sentiment classification, International Journal of Approximate Reasoning 105 (2019) 85 – 97.

[14] J. W. Grzymala-Busse, Lers-a system for learning from examples based on rough sets, in: R. Słowiński (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Springer Netherlands, Dordrecht, 1992, pp. 3–18.

[15] A. Wojna, R. Latkowski, Rseslib 3: Library of rough set and machine learning methods with extensible architecture, in: J. F. Peters, A. Skowron (Eds.), Transactions on Rough Sets XXI, Springer Berlin Heidelberg, Berlin, Heidelberg, 2019, pp. 301–323.

[16] W. Li, X. Jia, L. Wang, B. Zhou, Multi-objective attribute reduction in three-way decision-theoretic rough set model, International Journal of Approximate Reasoning 105 (2019) 327 – 341.

[17] B. Sang, L. Yang, H. Chen, W. Xu, Y. Guo, Z. Yuan, Generalized multi-granulation double-quantitative decision-theoretic rough set of multi-source information system, International Journal of Approximate Reasoning 115 (2019) 157–179.

[18] H.-Y. Zhang, S.-Y. Yang, Three-way group decisions with interval-valued decision-theoretic rough sets based on aggregating inclusion measures, International Journal of Approximate Reasoning 110 (2019) 31 – 45.

[19] E. Hüllermeier, J. Beringer, Learning from ambiguously labeled examples, in: A. F. Famili, J. N. Kok, J. M. Peña, A. Siebes, A. Feelders (Eds.), Advances in Intelligent Data Analysis VI, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 168–179.

[20] T. Cour, B. Sapp, B. Taskar, Learning from partial labels, J. Mach. Learn. Res. 12 (2011) 1501–1536.

[21] J. Dai, Q. Hu, J. Zhang, H. Hu, N. Zheng, Attribute selection for partially labeled categorical data by rough set approach, IEEE Transactions on Cybernetics 47 (9) (2017) 2460–2471.

[22] M. Hu, Y. Yao, Structured approximations as a basis for three-way decisions in rough set theory, Knowledge-Based Systems 165 (2019) 92 – 109.

[23] A. Campagner, D. Ciucci, Orthopartitions and soft clustering: Soft mutual information measures for clustering validation, Knowledge-Based Systems 180 (2019) 51–61.

[24] Z. Pawlak, Rough Sets: Theoretical Aspects Of Reasoning About Data, Vol. 9, 1991.

[25] D. Ciucci, Orthopairs and granular computing, Granular Computing 1 (3) (2016) 159–170.

[26] Z. Pawlak, A. Skowron, Rough sets: Some extensions, Information Sciences 177 (1) (2007) 28–40.

[27] H. Li, L. Zhang, X. Zhou, B. Huang, Cost-sensitive sequential three-way decision modeling using a deep neural network, International Journal of Approximate Reasoning 85 (2017) 68 – 78.

[28] A. Campagner, F. Cabitza, D. Ciucci, Three–way classification: Ambiguity and abstention in machine learning, in: T. Mihálydeák, F. Min, G. Wang, M. Banerjee, I. Düntsch, Z. Suraj, D. Ciucci (Eds.), Rough Sets, Springer International Publishing, Cham, 2019, pp. 280–294.

[29] A. Campagner, D. Ciucci, Three-way and semi-supervised decision tree learning based on orthopartitions, in: J. Medina, M. Ojeda-Aciego, et al. (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations, Vol. 854 of Communications in Computer and Information Science, Springer International Publishing, 2018, pp. 748–759.

[30] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.

[31] P. Smets, R. Kennes, The transferable belief model, Artificial Intelligence 66 (2) (1994) 191 – 234.

[32] L. Xu, A. Krzyzak, C. Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Transactions on Systems, Man, and Cybernetics 22 (3) (1992) 418–435.

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

[34] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, 1st Edition, Springer Publishing Company, Incorporated, 2014.

[35] L. Berrada, A. Zisserman, M. P. Kumar, Smooth loss functions for deep top-k classification, CoRR abs/1802.07595.

[36] J. Hernández-González, I. n. Inza, J. A. Lozano, Weak supervision and other non-standard classification problems, Pattern Recogn. Lett. 69 (C) (2016) 49–55.

[37] J. Cid-Sueiro, Proper losses for learning from partial labels, in: Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS'12, Curran Associates Inc., USA, 2012, pp. 1565–1573.

[38] E. Hüllermeier, Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization, Int. J. Approx. Reasoning 55 (7) (2014) 1519–1534.

[39] M.-L. Zhang, F. Yu, Solving the partial label learning problem: An instance-based approach, in: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, AAAI Press, 2015, pp. 4048–4054.

[40] W. Huang, S. Feng, L. Sun, C. Lang, Partial label learning via low rank representation and label propagation, in: Proceedings of the 10th International Conference on Internet Multimedia Computing and Service, ICIMCS '18, ACM, New York, NY, USA, 2018, pp. 32:1–32:7.

[41] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115.

[42] V. Gulshan, L. Peng, M. Coram, et al., Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs, Jama 316 (22) (2016) 2402–2410.

[43] C.-M. Svensson, R. Hübler, M. T. Figge, Automated classification of circulating tumor cells and the impact of interobsever variability on classifier training and performance, Journal of immunology research 2015 (2015) 573165–573165.

[44] F. Cabitza, A. Campagner, D. Ciucci, New frontiers in explainable AI: Understanding the GI to interpret the GO, in: Proceedings of the CD-MAKE 2019 conference, Vol. 11713 of LNCS, 2019, pp. 27–47.

[45] R. Jin, Z. Ghahramani, Learning with multiple labels, in: Advances in neural information processing systems, 2003, pp. 921–928.

[46] R. Eisinga, T. Heskes, B. Pelzer, M. Te Grotenhuis, Exact p-values for pairwise comparison of friedman rank sums, with application to comparing classifiers, BMC Bioinformatics 18 (2017) 1 – 68.

[47] J. D. Li, A two-step rejection procedure for testing multiple hypotheses, Journal of Statistical Planning and Inference 138 (6) (2008) 1521 – 1527.

[48] R. Goebel, A. Chander, K. Holzinger, et al., Explainable AI: The new 42?, in: A. Holzinger, P. Kieseberg, A. M. Tjoa, E. Weippl (Eds.), Machine Learning and Knowledge Extraction, Springer International Publishing, Cham, 2018, pp. 295–303.

[49] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, H. Mller, Causability and explainability of artificial intelligence in medicine, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9 (4) (2019) e1312.

[50] L.-P. Liu, T. G. Dietterich, Learnability of the superset label learning problem, in: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, JMLR.org, 2014, pp. II–1629–II–1637.

E-mail address: `a.campagner@campus.unimib.it`

E-mail address: `federico.cabitza@unimib.it`

E-mail address: `davide.ciucci@unimib.it`

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Via Bicocca degli Arcimboldi 8 – 20126 Milano, Italy