Learning from Fuzzy Labels: Theoretical Issues and Algorithmic Solutions

Andrea Campagner[a]

[a] *IRCCS Istituto Ortopedico Galeazzi, Milano, Italy*

Abstract

In this article we study the problem of learning from fuzzy labels (LFL), a form of weakly supervised learning in which the supervision target is not precisely specified but is instead given in the form of possibility distributions, that express the imprecise knowledge of the annotating agent. While several approaches for LFL have been proposed in the literature, including generalized risk minimization (GRM), instance-based methods and pseudo label-based learning, both their theoretical properties and their empirical performance have scarcely been studied. We address this gap by: first, presenting a review of the previous results relative to the sample complexity and generalization bounds for GRM and instance-based methods; second, studying both their computational complexity, by proving in particular the impossibility of efficiently solving LFL using GRM, as well as impossibility theorems. We then propose a novel pseudo label-based learning method, called Random Resampling-based Learning (RRL), which directly draws from ensemble learning and possibility theory and study its learning- and complexity-theoretic properties, showing that it achieves guarantees similar to those for GRM while being computationally efficient. Finally, we study the empirical performance of several state-of-the-art LFL algorithms on wide set of synthetic and real-world benchmark datasets, by which we confirm the effectiveness of the proposed RRL method. Additionally, we describe directions for future research, and highlight opportunities for further interaction between machine learning and uncertainty representation theories.

Key words: Machine Learning, Weakly Supervised Learning, Fuzzy Labels, Ensemble Learning, Statistical Learning Theory, Possibility Theory

1. Introduction

In recent years, applications of machine learning (ML) have spread into both research and industry. Arguably, one of the major driving forces behind this growth has been the increasing availability of successful and general-purpose learning algorithms (such as deep neural networks, kernel methods or ensemble

Email address: a.campagner@campus.unimib.it (Andrea Campagner)

learning) for the task of *supervised learning*, i.e., the task of learning from sets of fully labeled data, together with the wide availability of such labeled data in a multitude of public repositories and for a variety of tasks, ranging from computer vision to NLP, and from medicine to finance.

However, not all ML tasks fit neatly into the supervised learning category, and restriction to this specific family of tasks could be too much of a limitation in many natural settings. Indeed, in many contexts the need to collect fully labeled data can represent a bottleneck, due to the costs and time which may be required to produce such annotated data, that could in turn prevent the application of standard supervised learning pipelines. To address this limitation, in recent years, increasing attention has been given to the study of weakly supervised learning settings [84]. Weakly supervised learning refers to machine learning tasks situated in the spectrum between fully supervised and fully unsupervised learning [66], and encompasses various tasks such as multiple-instance learning [85], learning from aggregate data [19] and learning from imprecise data [46, 36].

In this latter case, in particular, the data and annotations can be imprecise or partial. Some examples include semi-supervised learning, but also more general tasks [36] such as evidential labels learning [30, 33, 67], in which partial labels are represented through belief functions (or, equivalently, mass functions); learning from fuzzy labels [34, 46], in which partial labels are represented through possibility distributions, and superset learning [11, 57], in which partial labels are represented by exclusive sets of alternatives. In general, in learning from imprecise data, one is typically interested in two different tasks [46], namely: *learning*, i.e., finding a good model of the data, and *disambiguation*, i.e., finding a good way to *precisify* the imprecise data. Thus, learning from imprecise data represents an area of interaction between uncertainty representation, for the definition of data representation and reasoning formalisms, and machine learning, for the development of algorithms and techniques whose aim is to address the two above-mentioned tasks. Despite the importance and practical relevance of learning from imprecise data in a variety of settings, so far research has mainly focused on specific tasks (in particular, semi-supervised learning and superset learning), while research on more general representations has been more limited. Furthermore, while several general-purpose algorithmic techniques (including generalized risk minimization [22, 30, 43, 46, 49, 72], instance-based methods [4, 7, 51, 79, 82] or pseudo label-based learning [52, 58, 80]) have been developed to address these learning tasks, their theoretical and empirical properties have not yet been widely studied [11, 15, 56, 59].

In this article, we address the above-mentioned gap by studying the problem of learning from imprecise data, considering both the theoretical perspective as well as the empirical one. We focus, in particular, on the problem of learning from fuzzy labels (LFL), i.e., the setting in which imprecision only affects the target and is represented using possibility distributions (i.e., epistemic fuzzy sets) over the set of possible label values. The decision to focus on this problem is due to multiple reasons, including:

– its wide applicability, indeed the LFL problem emerges naturally in several

settings, such as learning from anonymized data [71], learning from multi-rater data [19] or self-regularized learning [54];

– the relative easiness of acquiring fuzzy-labeled data in comparison with other forms of imprecise data [23, 47];

– its generality, indeed the LFL problem arises as a natural generalization of other common tasks in ML, such as semi-supervised and superset learning, by allowing to express the uncertainty about the labels in a more general form, as well as by allowing some form of label noise (i.e., errors in labeling).

We start our contribution by reviewing recent results on a theoretical characterization of the LFL task for two popular learning strategies (i.e., generalized risk minimization (GRM) and instance-based learning), which is grounded in the theory of fuzzy random sets [26, 31]. We then show that the LFL problem is computationally and learning-theoretically harder than standard supervised learning, in particular we prove that:

1. learnability guarantees for both GRM and instance-based learning methods are, in general, distribution-dependent;

2. while GRM enjoys learnability guarantees that almost match those for supervised learning, it is in general computationally hard to solve the GRM problem;

3. instance-based learning methods, while being computationally efficient, require exponentially larger sample sizes to meet the same learnability guarantees as GRM.

To address these limitations, we propose a novel pseudo label-based algorithm, called Random Resampling-based Learning (RRL), by which we show that a simple combination of ideas drawn from possibility theory [39] and ensemble learning achieves consistency (under weak assumptions about the data-generating distribution) as well as learnability guarantees that almost match those of GRM, while being computationally easy to train. This contribution, to our knowledge, is the first theoretical analysis of pseudo label-based learning for general learning from imprecise data tasks.

Our theoretical contributions are accompanied by an empirical validation, by which we evaluate several algorithms for the LFL task on a large set of benchmark datasets that encompasses both synthetic data as well as real-world problems. Through our empirical analysis we confirm the effectiveness of the proposed RRL algorithm, as well as of other state-of-the-art algorithm based either on GRM or regularized instance-based learning. More generally, we confirm the results of our theoretical analysis and, in particular, we show that:

1. while GRM achieves accuracy comparable with RRL, it does so at the cost of a much higher execution time;

2. while instance-based learning methods report lower accuracy than other state-of-the-art methods, their combination with dimensionality-reduction techniques achieves competitive performance.

Finally, we discuss directions for future research, emphasizing the need for better cross-fertilization between uncertainty representation theory and machine learning as a way to address more general weakly supervised learning tasks.

2. Background

The LFL problem is based on the use of fuzzy sets as a way to represent imprecise information about a target variable of interest. For this reason, in this section, we provide a basic overview on possibility theory and probability theory on fuzzy sets (through random fuzzy sets).

2.1. Possibility Theory and Random Fuzzy Sets

As described in the Introduction, LFL is the task of learning from imprecise information that is represented in the form of a possibility distribution (or, equivalently, a fuzzy set) over a given set of interest. These uncertainty representation models can be defined as follows:

Definition 2.1. *A possibility measure [38] is a function* $\Pi : 2^X \mapsto [0,1]$ *satisfying the following three properties:*

$$\Pi(\emptyset) = 0; \tag{1}$$

$$\Pi(X) = 1; \tag{2}$$

$$\forall A, B \subseteq X, \Pi(A \cup B) = \max\{\Pi(A), \Pi(B)\}. \tag{3}$$

If X *is countable,* Π *can be equivalently represented by a possibility distribution* π*, i.e., a function* $\pi : X \mapsto [0,1]$*, s.t.* $\Pi(A) = \sup_{x \in A} \pi(x)$*. For any given* $x \in X$*, we call the value* $\pi(x)$ *the* possibility degree *of* x*.*

Intuitively, a possibility distribution can be understood as a fuzzy set over X [81], whereas the membership degrees denote the plausibility of the elements. In this article we will focus on possibility measures defined on a finite set X: thus, without loss of generality, we will represent any possibility measure Π only in terms of the corresponding possibility distribution π.

A possibility distribution π is *normalized* if $\exists x \in X$ s.t. $\pi(x) = 1$. We denote with $\mathcal{F}(X)$ the collection of normalized possibility distributions over X. Given $\alpha \in L$ we denote with $\pi^\alpha = \{x : \pi(x) \geq \alpha\}$ the α-cut of π, and with $\pi^{\alpha+} = \{x : \pi(x) > \alpha\}$ the corresponding strong α-cut.

We note that possibility distributions can be associated with two different semantics: an *ontic* and an *epistemic* one. In this article, we only consider the *epistemic* semantics [26] of such distributions: thus, the possibility degree $\pi(x)$ is taken to represent the plausibility of value x compared with other elements $y \in X$. In particular, if $\pi(x) \geq \pi(y)$ then x is considered to be at least as plausible as y, and, if $\pi(x) > \pi(y)$ then x is more plausible than y.

The LFL setting (defined formally in Section 3.1) assumes that the data is generated i.i.d. from a joint distribution on fuzzy sets. Such distributions can be represented in terms of random fuzzy sets, which arise as the main object of study in the generalization of random set theory to the case of fuzzy sets. Thus, we first recall some basic notions from random set theory [26, 61].

Definition 2.2. *Given a set X, a random set (also, mass function) is defined as a probability distribution over 2^X, that is $m : 2^X \mapsto [0,1]$ s.t. $\sum_{A \subseteq X} m(A) = 1$. From any mass function, we define three set functions, namely the* belief *(Bel),* plausibility *(Pl) and* commonality *(Q) functions:*

$$Bel(A) = \sum_{B \in 2^X : B \subseteq A} m(B); \tag{4}$$

$$Pl(A) = \sum_{B \in 2^X : A \cap B \neq \emptyset} m(B); \tag{5}$$

$$Q(A) = \sum_{B \in 2^X : A \subseteq B} m(B). \tag{6}$$

Each of the functions given in Definition 2.2 (i.e., m, Bel, Pl and Q) can be considered an equivalent representation of a random set, in the sense that, starting from any of them, the others can be easily computed. In particular, it is easy to observe that Bel and Pl are dual of each other, that is, for any set $A \in 2^X$, it holds that:

$$Bel(A) = 1 - m(\emptyset) - Pl(A^c),$$
$$Pl(A) = 1 - m(\emptyset) - Bel(A^c),$$

where, for any set $A \in 2^X$, A^c denotes the set complement of A. With each mass function m, one can associate a collection of *focal sets*, which is formally equivalent to the *support*[1] of m, that is

$$\mathcal{S}(m) = \{A \in 2^X : m(A) > 0\}.$$

When the focal sets of m are nested (i.e., for each $A, B \in \mathcal{S}(m)$, either $A \subseteq B$ or $B \subseteq A$), then the plausibility function Pl is a possibility measure. More generally, with each mass function m we can associate a possibility distribution $pl : X \mapsto [0,1]$, called the *contour function* of m and defined as:

$$pl(x) = Pl(\{x\}) = Q(\{x\}).$$

We refer the reader to [28, 32, 61, 74] for extensive discussions on the mathematical formalism and semantics of random sets.

Finally, we recall some basic notions about the generalization of random set theory to the case of fuzzy events [26, 31].

[1] We recall that the *support* of a probability distribution $P : X \to [0,1]$ is the set of elements that have probability greater than 0, i.e. $supp(P) = \{x \in X : P(x) > 0\}$.

Definition 2.3. *Let X be a set. A* fuzzy random set *$\tilde{m}$ is defined as a probability distribution over the collection of fuzzy sets on X. That is, $\tilde{m} : \mathcal{F}(X) \mapsto [0, 1]$ and $\int_{\mathcal{F}(X)} \tilde{m}(\pi) = 1$. When the support of $\tilde{m}$ is finite, that is when:*

$$|\{\pi \in \mathcal{F}(X) : \tilde{m}(\pi) > 0\}| < \infty,$$

the normalization requirement above can equivalently be expressed as:

$$\sum_{\pi \in \mathcal{F}(X): \tilde{m}(\pi) > 0} \tilde{m}(\pi) = 1.$$

Belief, plausibility and commonality functions can be generalized to the setting of random fuzzy sets by using notions from generalized measure theory (i.e., the Choquet integral [42]): in such a setting, the belief, plausibility and commonality functions assign a value to each possible fuzzy event $\pi \in \mathcal{F}(X)$. As in this paper we only focus on singleton events, we only provide a restricted version of the above-mentioned definitions, and we refer the reader to [31] for a general definition that can be applied to any fuzzy set.

Definition 2.4. *A* singleton event *is a (normalized) possibility distribution that assigns possibility greater than 0 to a single element $x \in X$. That is, given $x \in X$, the singleton event $\tilde{x} \in \mathcal{F}(X)$ is defined by:*

$$\tilde{x}(x) = 1; \tag{7}$$

$$\forall x' \neq x \in X, \text{ it holds that } \tilde{x}(x') = 0. \tag{8}$$

Definition 2.5. *Let $\tilde{m}$ be a random fuzzy set and $\tilde{x}$ a singleton event. Then, we define the belief, plausibility and commonality of $\tilde{x}$ as:*

$$bel_{\tilde{m}}(\tilde{x}) = \tilde{m}(\tilde{x}), \tag{9}$$

$$pl_{\tilde{m}}(\tilde{x}) = \sum_{\pi \in \mathcal{F}(X): \pi(x) > 0} \tilde{m}(\pi), \tag{10}$$

$$q_{\tilde{m}}(\tilde{x}) = \sum_{\pi \in \mathcal{F}(X): \pi(x) = 1} \tilde{m}(\pi). \tag{11}$$

We note that in the case of fuzzy random sets $bel_{\tilde{m}}(\tilde{x}) \leq q_{\tilde{m}}(\tilde{x}) \leq pl_{\tilde{m}}(\tilde{x})$.

2.2. Supervised Learning and Learning Theory

In the framework of statistical learning theory [75] the starting point is the definition of a set Z, which is assumed to collect the features of interest of the objects to be studied. In supervised learning, Z is a product space $X \times Y$ with X being a vector space (i.e., the set of *feature vectors*) and Y being the target space: we will focus on the case of *classification*, whereas Y is finite.

Definition 2.6. *Let $Z = X \times Y$. A* learning problem *is a pair $(Z, \mathcal{D})$, where $\mathcal{D}$, called* data-generating distribution*, is a probability measure*[2] *over Z.*

[2] Formally, if Z is not countable, we need to assume the existence of a σ-algebra $\mathcal{B}$ of Z and $\mathcal{D}$ is a probability measure defined over $\mathcal{B}$. In the article we leave the σ-algebra $\mathcal{B}$ implicit, assuming a natural one can be defined over the domain of interest.

Intuitively, the data-generating distribution $\mathcal{D}$, and in particular the corresponding conditional distribution $\mathcal{D}(y|x)$, encodes a (not necessarily deterministic) dependency between input features and target labels.

Having fixed a learning problem (encoded by the pair $(Z, \mathcal{D})$), the goal of Machine Learning is to find a mapping f (or, more generally, a conditional density function) that provides a good approximation of $\mathcal{D}(y|x)$, based only on a finite sample S drawn from[3] $\mathcal{D}$. To formalize this notion, we introduce the concept of a *hypothesis space* $\mathcal{H}$, where each $h \in \mathcal{H}$ is a function $h : X \to \mathbb{R}^Y$, which represents the collection of functions from which a learning algorithm is allowed to select. The goodness of a hypothesis h is measured by means of a loss function $l : \mathcal{H} \times X \times Y \to \mathbb{R}$, that associates to each triple (h, x, y) a number that represents the cost of predicting label $h(x)$ when the true label is y. The most common example of a loss function is the so-called 0-1 loss function (also, accuracy), defined as:

$$l_{0-1}(h, x, y) = \begin{cases} 0 & y = \arg\max_{y \in Y} h^y(x) \\ 1 & y \neq \arg\max_{y \in Y} h^y(x) \end{cases}. \tag{12}$$

However, for computational complexity reasons, additional constraints are typically assumed on the loss function l.

Definition 2.7. *We say that a loss function l is* convex*, if it is convex in its first argument, i.e., it satisfies $l(\alpha h_1 + (1-\alpha)h_2, x, y) \leq \alpha l(h_1, x, y) + (1-\alpha)l(h_2, x, y)$. Similarly, we say that l is L-*Lipschitz *if, $\forall x, y \in X \times Y$, it holds that $|l(h_1, x, y) - l(h_2, x, y)| \leq L|h_1(x) - h_2(x)|$.*

Intuitively, convexity and Lipschitzness encode the notion that the loss function does not vary too rapidly and is sufficiently well-behaved.

Based on a loss function l, we can quantify the goodness of a hypothesis $h \in \mathcal{H}$ w.r.t. $\mathcal{D}$, by means of the *true risk*, that is the expected value of l w.r.t. $\mathcal{D}$. Formally, the true risk is defined as follows:

Definition 2.8 (True Risk)**.** *Let $\mathcal{D}$ be a data-generating distribution and l a loss function. Let f be a measurable function $f : X \to \mathbb{R}^Y$. The* true risk *of f is defined as :*

$$L_{\mathcal{D}}(f) = \int_{X \times Y} l(f, x, y) d\mathcal{D}(x, y). \tag{13}$$

Given a loss function l and a distribution $\mathcal{D}$, a natural requirement would be to find a function whose true risk is as small as possible. This notion is formalized through the following definitions:

Definition 2.9. *Let $\mathcal{D}$ be a data-generating distribution and l a loss function. The* Bayes classifier *is defined as:*

$$\hat{f} = \arg \min_{f \ measurable} L_{\mathcal{D}}(f). \tag{14}$$

[3]Thus, $\mathcal{D}$ is assumed to be a randomized mechanism that can be queried in order to obtain samples from Z.

Given an hypothesis space $\mathcal{H}$, the $\mathcal{H}$-relative Bayes classifier *is defined as:*

$$\hat{h} = \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h), \tag{15}$$

We say that the learning problem $(Z, \mathcal{D})$ is realizable *w.r.t. $\mathcal{H}$ if $L_{\mathcal{D}}(\hat{h}) = 0$.*

Intuitively, the Bayes classifier is the function (among all possible measurable functions) that minimizes the true risk. Similarly, the $\mathcal{H}$-relative Bayes classifier is the minimizer of the true risk within the hypothesis class $\mathcal{H}$.

Usually, $\mathcal{D}$ is assumed to be unknown, which means that the true risk of any hypothesis h cannot be directly computed. Instead, only a finite sample of data can be accessed: that is, we assume that any learning algorithm is given only a finite training set S sampled i.i.d. from $\mathcal{D}$. Then, the true risk of h can be estimated in terms of the corresponding *empirical risk* w.r.t. S, that is the average loss of h over S. Formally, the empirical risk is defined as follows:

Definition 2.10. *Let $(Z, \mathcal{D})$ be a learning problem, $\mathcal{H}$ a hypothesis space and l a loss function. Let $m \in \mathbb{N}$. Then, given $h \in \mathcal{H}$ and a finite training set $S = \langle (x_1, y_1), ..., (x_m, y_m) \rangle \sim \mathcal{D}^m$, the* empirical risk *of h w.r.t. S is defined as:*

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h, x_i, y_i). \tag{16}$$

We define *Empirical Risk Minimization* (ERM) to be any learning algorithm that, given a finite sample S, returns as output an hypothesis $h \in \mathcal{H}$ that minimizes the empirical risk. That is, formally:

Definition 2.11. *Let $\mathcal{H}$ be an hypothesis space, l a loss function. Then, an algorithm $ERM_{\mathcal{H}} : Z^{\omega} \mapsto \mathcal{H}$ is called an* empirical risk minimization *(ERM) algorithm if it satisfies:*

$$ERM_{\mathcal{H}}(S) \in \arg\min_{h \in \mathcal{H}} L_S(h),$$

where Z^{ω} denotes the collection of finite multi-sets over Z.

Aside from its intuitive appeal, ERM can be given a formal justification. Indeed, even though the true risk cannot be computed, the fundamental theorem of multi-class learning [27, 63] provides a way to bound the true risk of an ERM classifier in terms of its empirical risk and a measure of complexity of the hypothesis space $\mathcal{H}$. In the classification setting, the two most natural measures of complexity are the *Natarajan dimension* [63] ($d(\mathcal{H})$) and the *Rademacher complexity* [5] ($R(\mathcal{H})$) of $\mathcal{H}$.

Definition 2.12 (Shattering)**.** *Let $C \subseteq X$ be a subset of the feature space, and $f_0, f_1 : C \to Y$ be two functions, s.t. $\forall x \in C, f_0(x) \neq f_1(x)$: f_0, f_1 represent a partition of the instances in C into two different classes. We say that $\mathcal{H}$* shatters *$C \subseteq X$ if it holds that, $\forall B \subseteq C$, $\exists h \in \mathcal{H}$ s.t.:*

– $\forall x \in B, h(x) = f_0(x)$;

– $\forall x \notin B, h(x) = f_1(x)$.

Definition 2.13 (Natarajan Dimension)**.** *Let $\mathcal{H}$ be a hypothesis space. We say that $\mathcal{H}$ has Natarajan dimension d, denoted with $d(\mathcal{H}) = d$, if:*

– $\exists C \subseteq X, |C| = d$ *s.t.* $\mathcal{H}$ *shatters* C;

– $\nexists, C \subseteq X$, *with* $|C| > d$, *s.t.* $\mathcal{H}$ *shatters* C.

If for all $n \in \mathbb{N}$, $\exists C \subseteq X$, with $|C| = n$, s.t. $\mathcal{H}$ shatters C, we say that $d(\mathcal{H}) = \infty$.

Intuitively, the fact that $\mathcal{H}$ shatters a subset $C \subset X$ means that the hypotheses in $\mathcal{H}$ are sufficiently rich to be able to represent all possible (two-)partitions of C: thus, the Natarajan dimension naturally represents a measure of the richness of a hypothesis class $\mathcal{H}$, defined in terms of the ability of $\mathcal{H}$ to arbitrarily discriminate between pairs of classes from Y, for all sets of data whose cardinality is smaller than $d(\mathcal{H})$.

In regard to the Rademacher complexity[4], this can be defined as follows:

Definition 2.14. *Let S a finite dataset, $\mathcal{H}$ a hypothesis class and l a loss function. The* (empirical) Rademacher complexity *of $\mathcal{H}$, w.r.t. S, is defined as:*

$$R(\mathcal{H}, S) = \frac{1}{|S|}\mathbb{E}_{\sigma \sim \{\pm 1\}^m}\left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \sigma_i l(h, x_i, y_i)\right] \tag{17}$$

Intuitively, the Rademacher complexity measures the ability of $\mathcal{H}$ to fit random noise over any given training set S, where the random noise is modeled by means of the Rademacher distribution (i.e., the distribution that selects value 1 with probability $\frac{1}{2}$, and value -1 with probability $\frac{1}{2}$). We note that, in contrast to the Natarajan dimension, the Rademacher complexity of an hypothesis class $\mathcal{H}$ is defined relative to a training set: this implies that, in principle, it would be possible to compute the Rademacher complexity of any hypothesis class when given a fixed training set S. However, it can be shown that the problem of computing the Rademacher complexity is computationally hard (see, e.g.,[75]).

As briefly hinted at, the Natarajan dimension and the Rademacher complexity can be used to bound the value of the true risk of a learning algorithm, based on its empirical risk. Theorems 2.1 and 2.2, thus, provide an intuitive justification for the ERM algorithm.

[4]In the literature, the term Rademacher complexity can be used to denote either the *empirical Rademacher complexity*, as defined in Eq. (17), or its expectation $\mathcal{R}(\mathcal{H}) = \mathbb{E}_{S \sim \mathcal{D}^m}[R(\mathcal{H}, S)]$. The two versions of the Rademacher complexity are, obviously, related to each other: in particular, the expectation version can be bounded, with high probability, by the empirical one. In this article we will only refer to the empirical Rademacher complexity as it provides stronger bounds that, in contrast with the expectation version, are data-dependent: thus, when we state *Rademacher complexity* we always mean the empirical version of this complexity measure.

Theorem 2.1 (Natarajan Dimension-based Generalization Bound [27]). *Let $\mathcal{H}$ be an hypothesis class with Natarajan dimension d. Let l be the 0-1 loss. For each $\epsilon, \delta \in (0,1)$ and any distribution $\mathcal{D}$, then, with probability greater than $1-\delta$, if $ERM_{\mathcal{H}}$ is given a dataset S of size $m \geq n_0$ with*

$$n_0 = O(\frac{d \cdot ln(|Y|) + ln(\frac{1}{\delta})}{\epsilon^2}),$$

it holds that $|L_{\mathcal{D}}(ERM_{\mathcal{H}}(S)) - L_S(ERM_{\mathcal{H}}(S))| \leq \epsilon$.

Theorem 2.2 (Rademacher complexity-based Generalization Bound). *Let l be a loss function, and $\mathcal{H}$ be an hypothesis class. For each $\delta \in (0,1)$ and any distribution $\mathcal{D}$, then, with probability greater than $1-\delta$, if $ERM_{\mathcal{H}}$ is given a dataset S of size m, it holds that*

$$|L_{\mathcal{D}}(ERM_{\mathcal{H}}(S)) - L_S(ERM_{\mathcal{H}}(S))| \leq 2R(\mathcal{H}, S) + O(\sqrt{\frac{\ln(1/\delta)}{m}}),$$

where $R(\mathcal{H}, S)$ is the Rademacher complexity of $\mathcal{H}$ w.r.t. S, and the constants in the $O(\cdot)$ term depend only on l.

We notice that the bound provided by Theorem 2.1 is non-vacuous only if $\mathcal{H}$ has finite Natarajan dimension: this excludes many non-parameteric models, for which, in general, $d(\mathcal{H}) = \infty$. While, in principle, Theorem 2.2 can be applied to any hypothesis class, in practice it can be difficult to estimate or bound the Rademacher complexity of $\mathcal{H}$. However, similar bounds can also be derived for non-parametric approaches that satisfy some smoothness regularities, such as nearest neighbors methods:

Theorem 2.3 (Generalization Bound for Instance-based Models [75]). *Let X be a d-dimensional vector space, $Y = \{0,1\}$, $\eta_y(x) = \mathcal{D}(y=1|x)$ and assume that $\forall y, \eta_y$ is L-Lipschitz. Let $S \sim \mathcal{D}^m$ be a training set, and, for each $x \in X$, let $N(x,S)$ be the collection of nearest neighbors of x in S. Let $kNN(S) : X \to Y$ be the hypothesis defined by*

$$kNN(S)(x) = \arg\max_{y \in Y} \sum_{(x_i, y_i) \in S} \mathbb{1}_{y_i = y}.$$

Then it holds that:

$$\begin{aligned}\mathbb{E}(L_{\mathcal{D}}(kNN(S))) = (1 + \sqrt{\frac{8}{k}}) L_{\mathcal{D}}^{Bayes} \\ + (6c\sqrt{d} + k) m^{\frac{-1}{d+1}},\end{aligned}$$

where the expectation is w.r.t. the sampling of a training set S of size m from $\mathcal{D}$ and $L_D^{Bayes} = L_{\mathcal{D}}(\hat{f})$, where $\hat{f}$ is the Bayes classifier.

2.3. Superset Learning and Learning Theory

All the results presented in Section 2.2 only apply in the setting of supervised learning: in Section 3 we provide a generalization of these results to a weakly supervised learning setting, namely the LFL setting. As we will show, one fundamental difference between the two settings relates to the fact that Theorems 2.1 and 2.2 are distribution-free: that is, the given bounds on the generalization error do not depend on the data-generating distribution $\mathcal{D}$, but only on the hypothesis class $\mathcal{H}$ (and the loss function l). By contrast, the results we will prove in Section 3 will have distribution-dependent terms: we emphasize that this property is typical of settings that generalize supervised learning. To illustrate this phenomenon, we briefly focus on a restricted case of the LFL problem, called *superset learning*, which has been previously studied in the literature.

In superset learning, the data-generating distribution $\mathcal{D}$ is defined over the set $X \times Y \times 2^Y$: hence, instances are triples of the form (x, y, C). The interpretation of such an instance is that the set C represents the partial knowledge of the annotator about the true class label y: in general, one assumes that the *superset condition* holds, that is, it is assumed that $y \in C$. Thus, superset learning can be seen as a generalization of semi-supervised learning (in which case, for every instance (x, y, A) either $C = \{y\}$ or $C = Y$).

The definition of true risk in superset learning is the same as in supervised learning (see Eq. (13)). However, any learning algorithm has access only to a finite sample of the form $S = \{(x_1, C_1), \ldots, (x_m, C_m)\}$, sampled from the marginal distribution $\mathcal{D} \downarrow (X \times 2^Y)$ defined by:

$$\mathcal{D} \downarrow (X \times 2^Y)(x, C) = \sum_{y \in Y} \mathcal{D}(x, y, C).$$

A way to extend the empirical risk minimization to this setting is to lift the definition of empirical risk to the superset learning setting. For simplicity, we refer here to the *optimistic risk* formulation considered in [56] (as the only theoretical results for superset learning apply to this formulation), while a general discussion of learning criteria for superset learning (as a special case of LFL) is given in Section 3.1. In the setting of superset learning, the optimistic risk is defined as:

$$L_S^O(h) = \frac{1}{m} \sum_{i=1}^{m} \min_{y \in C_i} l(h, x_i, y)$$

Based on this extension of the empirical risk, the authors of [56] generalized Theorem 2.1 to the setting of superset learning as follows:

Theorem 2.4 (Generalization Bound for Superset Learning [56])**.** *Let $\mathcal{H}$ be an hypothesis class with Natarajan dimension d. Let $\mathcal{D}$ be a distribution, and let α be defined as:*

$$\alpha = sup_{(x,y) \in X \times Y} \{\mathcal{D}(C|x, y) : \mathcal{D} \downarrow (X \times Y)(x, y) > 0, l \neq y)\}.$$

Assume that $(Z, \mathcal{D})$ is realizable w.r.t. $\mathcal{H}$. For each $\epsilon, \delta \in (0, 1)$ if any learning algorithm $A : (X \times 2^Y)^\omega \to \mathcal{H}$ that minimizes the optimistic risk $L_S(O)$ is given

a dataset S of size $m \geq n_0$ with:

$$n_0 = O\left(\frac{1}{\ln\left(\frac{2}{1+\alpha^*}\right)\epsilon}\left(d \cdot ln\left(\frac{d|Y|^2}{\ln\left(\frac{2}{1+\alpha^*}\right)\epsilon}\right) + ln\frac{1}{\delta}\right)\right),$$

then with probability greater than $1-\delta$, it holds that $L_{\mathcal{D}}(A(S)) \leq \epsilon$.

As can be seen, the bound given in Theorem 2.4 is similar to the one given in 2.1: the main difference among the two bounds is the presence (in Theorem 2.4) of an additional *distribution-dependent* parameter α. Intuitively, the closer α is to 1, the harder the corresponding learning problem is: indeed, α represents the *degree of ambiguity* of the data-generating distribution $\mathcal{D}$, that is the probability that an incorrect label $l \neq y$ would always be included in the superset label C. This is in contrast with Theorem 2.1 which provides a distribution-free bound: indeed, the learnability guarantee given in Theorem 2.4 only holds conditionally, under the assumption that the degree of ambiguity α is sufficiently close to 0. As we will show in Section 3, distribution-conditionality is an inherent limitation in learning from imprecise data.

3. Learning from Fuzzy Labels

In this section we study the LFL problem from a theoretical perspective, based on statistical learning theory and computational complexity. First, in Section 3.1 we provide an introduction to the LFL problem from a formal perspective, as well as illustrate connections with other learning problems and potential applications. In Sections 3.2 and 3.3 we review some previous results on the learnability of the LFL problem as well as provide new positive results in this sense. We focus, in particular, on two specific learning algorithms, namely generalized risk minimization (in Section 3.2) and generalized nearest neighbors (in Section 3.3). We also prove negative results related to the LFL problem, by first providing a No Free Lunch Theorem and then studying the computational complexity of generalized risk minimization. To address the above-mentioned limitations, in Section 3.4 we propose a novel pseudo label-based learning algorithm and study its learning-theoretic and complexity-theoretic properties, providing promising results.

3.1. Learning from Fuzzy Labels: Background

As mentioned in the Introduction, the LFL problem is a form of weakly supervised learning, and particularly a generalization of the problem of supervised learning, in which the true label associated with each instance x is not observed, but only an imprecise version of it is available, represented as a possibility distribution π_x over the class labels. Formally:

Definition 3.1. *Let $Z = X \times Y$ be the sample space, then a LFL problem is defined as a pair $(Z, \tilde{m})$, where the* data-generating distribution *$\tilde{m}$ is a random fuzzy set defined over $Z \times \mathcal{F}(Y)$. We denote with $\tilde{m}(\cdot|x, y)$ the conditional probability distribution of fuzzy labels given a precise instance (x, y).*

The value $\tilde{m}(x, y_x, \pi_x)$ denotes the probability of observing the imprecisely labeled pair (x, π_x) when y_x is the true label associated with x. Thus, $\tilde{m}$ can be understood as a distribution that governs the sampling of data triples and specifies, for each precise instance (x, y_x) and (imprecise) fuzzy label π_x, the probability of observing (x, y_x, π_x). The data-generating distribution $\tilde{m}$ is assumed to satisfy the following condition, called *weak superset assumption*:

Definition 3.2 (Weak Superset Assumption)**.** *Let $\tilde{m}$ be a data-generating distribution. Then, we assume that :*

$$\tilde{m}(\pi_x(y_x) > 0 | x, y_x) = pl(\tilde{y}_x) = 1, \tag{18}$$

where $\tilde{y}_x$ denotes the fuzzy (singleton) event $\tilde{y}_x(\hat{y}) = \begin{cases} 1 & \hat{y} = y \\ 0 & \text{otherwise.} \end{cases}$.

Intuitively, the weak superset assumption asserts that the correct label y_x is never considered impossible. If $\tilde{m}$ also satisfies the stronger condition:

$$\tilde{m}(\pi_x(y_x) = 1 | x, y_x) = q(\tilde{y}_x) = 1, \tag{19}$$

then $\tilde{m}$ is said to satisfy the *strong superset assumption*[5], which intuitively states that the correct label y_x is always considered to be fully possible. The definition of an LFL problem is illustrated in Example 3.1.

Example 3.1. *Let us consider a simple binary classification problem from an hypothetical medical setting. Let $X = [-1, 1]$ be the measurement value for a certain analyte in a blood sample and $Y = \{-1, 1\}$, where 1 means that a certain disease is present. Assume that there exists a linear function $g(x) = w \cdot x$ such that, for each $x \in X$, the corresponding target label is deterministically defined by a function $f : X \times [0, 1] \to Y$, defined as $f(x) = sign(g(x))$. Furthermore, assume, for simplicity, that each possible value of x is equally probable.*

Assume, however, that the function g is unknown. Thus, the true target label $f(x)$ is not available: given a patient, a doctor can only examine the features x to provide a diagnosis for the patient. In particular the doctor describes its confidence about whether any given patient has or not the disease using a possibility distribution π: given x, with probability $1 - \epsilon$, it holds that $\pi_x(f(x)) = 1, \pi_x(-f(x)) = \eta$; while, with probability ϵ, it holds $\pi_x(-f(x)) = 1, \pi_x(f(x)) = \eta$, where $\eta \in [0, 1)$ is the confidence the doctor assigns to the diagnosis of which he or she is least confident about, whereas $1 - \epsilon \in [0, 1)$ is the accuracy of the doctor.

This problem can modeled as a LFL problem. Indeed, we can define the

[5]The strong superset assumption corresponds to the superset assumption from the *superset learning* setting, introduced in Section 2.3. Superset learning can be expressed as a special case of LFL in which $\tilde{m}$ is actually a random set, i.e., $\tilde{m}(x, y_x, \pi_x) > 0$ iff $\forall y \in Y, \pi_x(y) \in \{0, 1\}$, which also satisfies the strong superset assumption.

data-generating distribution $\tilde{m}$ as the probability density

$$\tilde{m}(x,y,\pi) = \begin{cases} \frac{1-\epsilon}{2} & \text{if } y = f(x) \wedge \pi(y) = 1 \\ \frac{\epsilon}{2} & \text{if } y = f(x) \wedge \pi(y) = \eta \\ 0 & \text{otherwise} \end{cases}.$$

Then, $\tilde{m}$ satisfies the weak superset assumption iff $\eta > 0$ or $\epsilon = 0$: in particular if $\epsilon = 0$ it also satisfies the strong superset assumption.

Given a data-generating distribution $\tilde{m}$, we define the following relevant distribution parameters (see also Example 3.2):

Definition 3.3. *Let $\tilde{m}$ be a data-generating distribution. Let $q_{x,y}(\tilde{u}) = q_{\tilde{m}(\cdot|x,y)}(\tilde{u})$, $pl_{x,y}(\tilde{u}) = pl_{\tilde{m}(\cdot|x,y)}(\tilde{u})$. Then, we define the following parameters:*

- Ambiguity, $\alpha = \sup_{(x,y)\in X\times Y}\{q_{x,y}(\tilde{u}) : \tilde{m}\downarrow(X\times Y)(x,y) > 0, u \neq y\}$;
- Lower Knowledge, $k_* = \inf_{(x,y)\in X\times Y}\{q_{x,y}(\tilde{y}) : \tilde{m}\downarrow(X\times Y)(x,y) > 0\}$;
- Upper Knowledge, $k^* = \sup_{(x,y)\in X\times Y}\{q_{x,y}(\tilde{y}) : \tilde{m}\downarrow(X\times Y)(x,y) > 0\}$;
- Falsifiability, $\phi = sup_{(x,y)\in X\times Y}\{pl_{x,y}(\tilde{u}) : \tilde{m}\downarrow(X\times Y)(x,y) > 0, u \neq y\}$.

where $\tilde{m}\downarrow(X\times Y)$ is the marginal of $\tilde{m}$ over $X\times Y$, defined pointwise by:

$$\tilde{m}(x,y) = \int_{\mathcal{F}(Y)} \tilde{m}(x,y,\pi)d\tilde{m}.$$

Intuitively, similarly to the case of superset learning, the parameters α, k_*, k^* and ϕ implicitly represent the hardness of (learning from) $\tilde{m}$. The value of α represents a bound on the probability that an incorrect label u would be considered as maximally plausible: in particular, $\alpha = 1$ only when, with probability 1 over the sampling of instance $(x,y,\pi) \sim \tilde{m}$, it exists $y' \in Y$ with $y' \neq y$ and $\pi(y') = 1$. k_* (resp., k^*) represents a lower (resp., upper) bound on the probability with which the correct label y could be correctly identified: in particular, we note that, by the weak superset assumption, it holds that $0 < k_* \leq k^*$ and $k^* = 1$ iff the strong superset assumption holds. Finally, ϕ represents a bound on the probability that any hypothesis h would be able to correctly discriminate the correct label y from an incorrect one: indeed, $\phi = 0$ only when, with probability 1 over $(x,y,\pi) \sim \tilde{m}$, it holds that $\pi(y') > 0$ iff $y' = y$.

Example 3.2. *Let $\tilde{m}$ be the data-generating distribution defined in Example 3.1. Then, it holds that: $\alpha = \epsilon$, $k_* = k^* = 1-\epsilon$ and $\phi = \begin{cases} 1 & \text{if } \eta > 0 \\ \epsilon & \text{otherwise} \end{cases}$.*

Based on the definition of data-generating distribution, we generalize the definition of the true risk of a hypothesis, which, equivalently to the fully supervised setting, can be defined as:

$$L_{\tilde{m}}(h) = \int l(h,x,y_x)d\tilde{m}.$$

As in the supervised setting, however, any learning algorithm $A : (X \times \mathcal{F}(Y))^\omega \mapsto \mathcal{H}$ is unable to access the full data-generating distribution $\tilde{m}$, but rather only a finite sample S generated from it. Furthermore, and in contrast with supervised learning, we note that S is not sampled directly from $\tilde{m}$, as A does not have access to the true label y_x for a triple (x, y_x, π_x), but rather from the marginal *imprecise data-generating distribution*:

Definition 3.4. *Let $\tilde{m}$ be a data-generating distribution. Then, the corresponding* imprecise data-generating distribution *is defined as the marginal distribution $\tilde{m} \downarrow (X \times \mathcal{F})$, s.t.:*

$$\tilde{m} \downarrow (X \times \mathcal{F}(Y))(x, \pi) = \sum_{y \in Y} \tilde{m}(x, y, \pi). \tag{20}$$

The precise data-generating distribution *is defined as the marginal distribution $\tilde{m} \downarrow (X \times Y)$, s.t.:*

$$\tilde{m} \downarrow (X \times Y)(x, y) = \int_{\pi \in \mathcal{F}(Y)} \tilde{m}(x, y, \pi) d\tilde{m}. \tag{21}$$

Example 3.3. *Let $\tilde{m}$ be the data-generating distribution defined in Example 3.1. The imprecise data-generating distribution $\tilde{m} \downarrow (X \times \mathcal{F}(Y))$ derived from $\tilde{m}$ can be defined as the probability density:*

$$\tilde{m} \downarrow (X \times \mathcal{F}(Y))(x, \pi) = \tilde{m}(x, f(x), \pi),$$

while the corresponding precise data-generating distribution $\tilde{m} \downarrow (X \times Y)$ can be defined as the probability density:

$$\tilde{m} \downarrow (X \times Y)(x, y) = \begin{cases} \frac{1}{2} & \text{if } y = f(x) \\ 0 & \text{otherwise} \end{cases}$$

A learning algorithm A can only access an *imprecise training set* $\tilde{S} \in (X \times \mathcal{F}(Y))^\omega$, drawn from $\tilde{m} \downarrow (X \times \mathcal{F}(Y))$. This implies that the empirical risk cannot be directly computed based on $\tilde{S}$ and, therefore, no straightforward generalization of the ERM principle exists. To address this shortcoming, several approaches have thus been proposed in the literature: in this paper we focus on three main approaches, namely *generalized risk minimization* (GRM), *generalized nearest neighbors* (GNN) and *pseudo label-based learning*, whose properties are studied, respectively in Sections 3.2, 3.3 and 3.4.

Before getting to the theoretical analysis of the above-mentioned methods, we briefly discuss the relevance of LFL problems for practical applications. LFL represents a general but natural setting that extends other previously studied weakly supervised learning settings, including semi-supervised learning and superset learning, and allows to flexibly represent a wider variety of learning problems. In particular, semi-supervised learning corresponds to the case where:

$$\tilde{m}(\pi_x(y_x) \in \{0, 1\} \wedge |\pi_x^0| \in \{1, |Y|\} | x, y_x) = 1,$$

that is, either only the correct label or the full set of possible labels is observed. By contrast, superset learning (already introduced in Section 2.3) can be formulated as the special case of LFL in which:

$$\tilde{m}(\pi_x(y_x) \in \{0,1\}|x, y_x) = 1,$$

and $\tilde{m}$ satisfies the strong superset property: intuitively, in superset learning only a set of labels is observed, and this set is always assumed to contain the true, unknown label (see also Section 2.3).

By way of this additional flexibility, the LFL setting can be understood as a way to formalize different natural learning settings. Here we briefly recall some of these settings to illustrate the wide applicability of the problem studied in this paper. Obviously, LFL can be understood as a natural setting to model learning problems in which the annotating agent lacks complete knowledge about the task under consideration and instead only has a partial and imprecise conceptualization of it. Moreover, the LFL setting has been applied to model learning from crowdsourced labels [19] problems, as well as learning with noisy labels [54, 55] problems. In the case of learning from crowdsourced labels, for each instance x multiple annotating agents $o_1, \ldots, o_k$ provide each a label $y_x^1, \ldots, y_x^k$, where the y_x^i can potentially be distinct. A possibility distribution over Y is then obtained as:

$$\pi_x(y) = \frac{|\{o_i : y_x^i = y\}|}{\max_{y' \in Y} |\{o_i : y_x^i = y'\}|}.$$

In this setting, the weak superset property corresponds to the assumption that, for each instance x, at least one of the agents o_i provides the correct label y_x: when k is large, or Y is small, this assumption is not too strong. In the learning from noisy labels setting, by contrast, it is assumed that the observed samples (x, y_c) may be affected by labeling errors: that is, the annotating agent may err in the annotation process and hence it may happen that $y_c \neq y_x$. This problem, which has been widely studied in the machine learning literature [1, 10, 53, 64], can be recast in the LFL setting by converting the precise, but possibly incorrect, label y_c into a possibility distribution π_x^τ [54, 55] defined as:

$$\pi_x^\tau(y) = \begin{cases} 1 & y = y_c \\ \tau & \text{otherwise} \end{cases}.$$

3.2. Learning from Fuzzy Labels: Generalized Risk Minimization

In this section we study a popular approach for solving LFL problems, that in Section 3.1 we called GRM. The GRM method [46] is based on the idea of extending a loss function $l : \mathcal{H} \times X \times Y \to \mathbb{R}$ to a surrogate loss function over fuzzy labels $\tilde{l} : \mathcal{H} \times X \times \mathcal{F}(Y) \to \mathbb{R}$. Such a transformation can be performed by means of the following definition:

Definition 3.5 (Imprecise Loss)**.** *Let l be a loss function. Then, we define the* imprecise loss $\tilde{l}$ *based on l as:*

$$\tilde{l}(h, x, \pi) = \int_0^1 A(\{l(h, x, y') : y' \in \pi^\alpha\})d\alpha, \tag{22}$$

where $\pi^\alpha = \{y' \in Y : \pi(y) \geq \alpha\}$ *is the* α*-cut of the fuzzy label* π *and* A *is an* aggregation function *specifying how to aggregate different loss function values. The true risk and empirical risk of an hypothesis* $h \in \mathcal{H}$*, can be defined as:*

$$L_{\tilde{m}}(h) = \int l(h, x, y) d\tilde{m}, \tag{23}$$

$$L_{\tilde{S}}(h) = \frac{1}{m} \sum_{(x_i, \pi_i) \in \tilde{S}} \tilde{l}(h, x_i, \pi_i). \tag{24}$$

Example 3.4. *Let* $l(h, x, y') = (h(x) - y')^2$*, where* $\forall x \in X, h(x) \in \mathbb{R}$*,* $y' \in \{-1, 1\}$*. Assume that the imprecisely labelled instances* (x, π) *are generated from the imprecise data-generating distribution defined in Example 3.3. Then, with probability* $1-\epsilon$*, it holds that* $\tilde{l}(h, x, \pi) = (1-\eta)(h(x)-f(x))^2+\eta(h(x)+f(x))^2$*. Similarly, with probability* ϵ*, it holds that* $\tilde{l}(h, x, \pi) = (1 - \eta)(h(x) + f(x))^2 + \eta(h(x) - f(x))^2$*.*

Then, GRM is implemented by simply applying the ERM rule for $\mathcal{H}$ on the imprecise training set $\tilde{S}$ by considering the imprecise loss $\tilde{l}$. Formally, the GRM learning algorithm can be defined as follows:

Definition 3.6 (Generalized Risk Minimization)**.** *Let* l *be a loss function,* A *an aggregation operator and* $\tilde{l}$ *be the imprecise loss based on* l *and* A*. Then, an algorithm* $GRM_{\mathcal{H},A} : (X \times \mathcal{F}(Y))^\omega \to \mathcal{H}$ *is called a* generalized risk minimization *(GRM) algorithm if it satisfies:*

$$GRM^f_{\mathcal{H}}(\tilde{S}) \in \arg\min_{h \in \mathcal{H}} L_{\tilde{S}}(h).$$

Several versions of GRM have been proposed in the literature [36], based on different ways to select an appropriate aggregation function: popular alternatives include the average [22, 30, 49], the maximum [43, 44], the minimum [11, 46] or variants thereof [48]. While different choices of aggregation function correspond to different properties of the derived GRM rule [25, 24], in this article we will focus on the case of the minimum aggregation operator (usually called *optimistic risk minimization* [46] or minimin optimization [65]). We decided to focus on this setting, in particular, as it has been the focus of most previous investigations of GRM (in the superset learning setting) [11, 15, 24, 56]. Nonetheless, we believe that future work should also evaluate the theoretical properties of other variants of GRM, with particular reference to the so-called *pessimistic risk minimization* approach (also called minimax optimization, and based on the maximum aggregation operator), due to the importance of this method in robust learning and estimation [43].

3.2.1. Learning-theoretic Properties

As a first result concerning the learning-theoretic properties of GRM (and, specifically, of optimistic risk minimization) we provide a bound on the true risk of optimistic risk minimization, obtained by applying a technique similar

to Theorem 2.1. To this aim, consider the 0-1 loss: the corresponding imprecise loss can be shown to be equivalent to $\tilde{l}_{0-1}(h, x, \pi) = 1 - \pi(\arg\max_{y \in Y} h^y(x))$ [15]. Then, the following result holds:

Theorem 3.1 (LFL Generalization Bound for GRM [15])**.** *Let $\mathcal{H}$ be an hypothesis class with Natarajan dimension d. Let $\tilde{m}$ be a data-generating distribution satisfying the weak superset property. Let $\theta_{\tilde{m}} = \log_2(\frac{2}{1+\max\{\phi, 1-k_*\}})$, where ϕ, k_* are the respective Falsifiability and (Lower) Knowledge parameters. For each $\epsilon, \delta \in (0,1)$, then, with probability greater than $1-\delta$, if $GRM^f_{\mathcal{H}}$ is given an imprecise dataset $\tilde{S}$ of size $m \geq n_0$ with*

$$n_0 = O\left(\frac{1}{(\epsilon\theta_{\tilde{m}})^2}(d \cdot ln(\frac{d \cdot |Y|^2}{(\epsilon\theta_{\tilde{m}})^2}) + ln(\frac{1}{\delta}))\right),$$

it holds that $|L_{\tilde{m}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S}))| \leq \epsilon$.

Corollary 3.1. *Let $\mathcal{H}$ be a hypothesis class with Natarajan dimension d. Let $\tilde{m}$ be the data-generating distribution and $\theta_{\tilde{m}} = log_2(\frac{2}{1+\max\{\phi, 1-k_*\}})$, where ϕ, k_* are the respective Falsifiability and (Lower) Knowledge parameters. Let h^* be the classifier with minimal risk in $\mathcal{H}$, then with probability greater than $1-\delta$, when given an imprecise training set $\tilde{S}$ of size m, it holds that:*

$$|L_{\tilde{m}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L_{\tilde{m}}(h^*)| \leq \epsilon(m, \delta, \theta_{\tilde{m}}), \tag{25}$$

where $\epsilon(m, \delta, \theta_{\tilde{m}}) \in O\left(poly(\frac{1}{m}, \frac{1}{\delta}, \frac{1}{\theta_{\tilde{m}}})\right)$.

Proof. By summing and subtracting the empirical risk of the result of GRM (i.e., , it holds that $|L_{\tilde{m}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L_{\tilde{m}}(h^*)|$ (i.e., the gap between the true risk of the result of GRM and the true error of the $\mathcal{H}$-relative Bayes classifier) can be upper bounded by :

$$|L_{\tilde{m}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S})) + L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L_{\tilde{m}}(h^*)|,$$

which, by noting that $L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S})) \leq L^f_{\tilde{S}}(h^*)$ (since, by definition of GRM, the empirical risk of the GRM hypothesis is minimal among all hypothesis in $\mathcal{H}$)) in turn can be upper bounded by:

$$|L_{\tilde{m}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S})) + L^f_{\tilde{S}}(h^*) - L_{\tilde{m}}(h^*)|.$$

By the triangle inequality, this last term above can be upper bounded by :

$$|L_{\tilde{m}}(GRM^f_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S}))| + |L_{\tilde{m}}(h^*) - L^f_{\tilde{S}}(h^*)|.$$

Finally, by Theorem 3.1, the result follows, since we can show that the last term above can be upper bounded by $\epsilon(m, \delta, \theta_{\tilde{m}})$, where, following Theorem 3.1, $\epsilon(m, \delta, \theta_{\tilde{m}}) = O(\sqrt{\frac{d\ln(\frac{d|Y|^2}{m\theta^2_{\tilde{m}}}) + \ln(\frac{1}{\delta})}{m\theta^2_{\tilde{m}}}})$. □

Thus, the previous result shows that, if we assume the weak superset property holds and the distribution parameter $\theta_{\tilde{m}}$ is not too small for the learning problem at hand, then, by using the optimistic risk minimization algorithm the risk can be made arbitrarily close to that of the $\mathcal{H}$-relative Bayes classifier. Intuitively, the parameter $\theta_{\tilde{m}}$ directly represents the hardness of the problem of learning from the data-generating distribution $\tilde{m}$. Indeed, $\theta_{\tilde{m}}$ will be small when either ϕ is close to 1, or k_* is close to 0. In the first case, there exists at least one incorrect label y' which is always associated with a possibility degree strictly greater than 0. Uence, under the weak superset assumption, no learning algorithm that has access only to the fuzzy labels could ever detect a classification error whereby an incorrect label y' is predicted instead of the correct one y. In the second case, by contrast, the correct label y is never associated with a possibility degree equal to 1: since the fuzzy labels are assumed to be normalized, this implies that there exists an incorrect label y' that is always associated with a possibility degree equal to 1. Thus, any learning algorithm that favors labels with higher possibility degrees (such as GRM) will be tricked into an incorrect disambiguation. We note that the previous result is weaker than Theorem 2.1, due to the dependence on the $\theta_{\tilde{m}}$ parameter, which is distribution-conditional. One natural question, thus, is whether a distribution-free guarantee on the true risk could, in principle, be found. Theorem 3.2 provides a negative result, by showing that, without any assumption on $\theta_{\tilde{m}}$, it is impossible to solve the LFL problem with GRM.

Theorem 3.2. *Let $\mathcal{H}$ be a hypothesis class with Natarajan dimension d defined on a countable sample space $Z = X \times \{0,1\}$. Let $\delta \in [0,1]$ and let $\epsilon \in [0,1]$. Then, there exists distributions $\tilde{m}_1, \tilde{m}_2$ over Z s.t.*

$$\tilde{m}_1 \downarrow (X \times Y) = \tilde{m}_2 \downarrow (X \times Y),$$

but it holds, with probability greater than δ over the sampling of an imprecise training set $\tilde{S}$, that:

$$|L_{\tilde{m}_1}(GRM^f_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S}))| \leq \epsilon; \tag{26}$$

$$|L_{\tilde{m}_2}(GRM^f_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S}))| > \epsilon. \tag{27}$$

Proof. We actually prove a stronger result, by assuming that the learning problem is realizable, i.e., we start from a distribution $\mathcal{D}$ over $X \times Y$ s.t. it $\exists h \in \mathcal{H}$ with $L_{\mathcal{D}}(h) = 0$. Intuitively, the proof relies on the fact that, based on $\mathcal{D}$, we can easily construct two data-generating distributions $\tilde{m}_1$ and $\tilde{m}_2$ by requiring that $\tilde{m}_1 \downarrow (X \times Y) = \tilde{m}_2 \downarrow (X \times Y) = \mathcal{D}$, and then setting the distributional parameters $\theta_{\tilde{m}_1}$ and $\theta_{\tilde{m}_2}$, such that:

$$\sqrt{\frac{d\ln(\frac{d|Y|^2}{m\theta^2_{\tilde{m}_1}}) + \ln(\frac{1}{\delta})}{m\theta^2_{\tilde{m}_1}}} \leq \epsilon < \sqrt{\frac{d\ln(\frac{d|Y|^2}{m\theta^2_{\tilde{m}_2}}) + \ln(\frac{1}{\delta})}{m\theta^2_{\tilde{m}_2}}}. \tag{28}$$

To formalize this intuition, for $\tilde{m}_1$, we set $\tilde{m}_1(\pi|x,y) = 1$ iff $\pi = \tilde{y}$ (i.e., π is the singleton event $\tilde{y}$ corresponding to the true label y): this implies that $k_* = 1$ and $\phi = 0$, hence $\theta_{\tilde{m}_1} = 1$. By contrast, for $\tilde{m}_2$, fix an hypothesis $h \in \mathcal{H}$ s.t., for all $x \in X$, it holds that $\mathcal{D}(x, h(x)) = 0$. Then, we set $\tilde{m}_2(\tilde{h(x)}|x,y) = 1$: this implies that $k_* = 0$ and $\phi = 1$, hence $\theta_{\tilde{m}_2} = 0$. Furthermore, $L^f_{\tilde{S}}(h) = 0$, by definition, hence $h = GRM^f_{\mathcal{H}}(\tilde{S})$. However, by construction, $L_{\tilde{m}_2}(h) = 1$. The result follows by noting that the problem of learning from $\tilde{m}_1$ is equivalent to a supervised learning problem, hence Theorem 2.1 can be applied. □

Thus, Theorem 3.2 asserts that the learnability of LFL is strongly distribution-conditional: if the parameter $\theta_{\tilde{m}}$ is too small, then we cannot guarantee that the problem is solvable from finite samples (not even asymptotically). Furthermore, the situation where $\theta_{\tilde{m}}$ is small cannot be detected from an imprecise training set alone: indeed, by definition, without access to the true labels, it is impossible to estimate $\theta_{\tilde{m}}$. Despite this negative result, we remark that the assumption of $\theta_{\tilde{m}}$ being not too small is not overly restrictive: indeed, one would usually assume that the knowledge of the annotating agent is not too far from the true labels. In this sense, the above-mentioned assumption is analogous to the assumptions commonly made, e.g., in learning from noisy labels [1]: in this latter setting one typically requires that, even though the annotating agent may sometimes make a labeling error, the probability of such an event is not too large.

3.2.2. Complexity-theoretic Properties

A more impactful negative result for optimistic risk minimization concerns its computational complexity. In particular, Proposition 3.1 shows that, even if we restrict GRM to a very simple class of learning problems (which can be solved efficiently in the supervised learning setting), optimistic risk minimization does not admit any polynomial-time algorithm (unless P = NP):

Proposition 3.1. *Let $\tilde{S}$ be an imprecise training set obtained sampling i.i.d. from $\tilde{\mathcal{D}}$, where $X = \mathbb{R}^d$ and $Y = \{-1, 1\}$. Let $\mathcal{H}$ be the class of half-spaces*[6] *on X. Let $g : \mathbb{R}^d \times X \times Y \to \mathbb{R}$ be any loss function defined by $g(w, x, y) = l(y, \langle w, x\rangle)$, for some function $l : Y \times \mathbb{R} \to \mathbb{R}$ satisfying the following properties:*

1. *for each $(x, y) \in Z$, l is convex in w;*
2. *$sign(y) = sign(\langle w, x\rangle) \implies l(y, \langle w, x\rangle) < l(-y, \langle w, x\rangle)$;*
3. *if $sign(y - \langle w, x\rangle) = sign(y)$ then $l(y, \cdot)$ is monotonically increasing in $|y - \langle w, x\rangle|$.*

[6] The class of half-spaces is defined by associating with each vector $w \in \mathbb{R}^d$ an hypothesis $h_w \in \mathcal{H}$, with $h_w(x) = sign(\langle w, x\rangle)$.

Let $\tilde{g} : \mathcal{H} \times X \times \mathcal{F}(Y) \to \mathbb{R}$ be the imprecise loss obtained from g (see Definition 3.5) and defined as[7]*:*

$$\tilde{g}(h, x, \pi) = \int_0^1 A(\{l(h, x, y') : y' \in \pi^\alpha\})d\alpha. \tag{29}$$

If l does not also satisfy the following property:

$$\forall t \in [-1, 1] \text{ it holds that } l(y, t) = l(-y, t), \tag{30}$$

then, unless P = NP*, for any polynomial-time randomized learning algorithm A it holds that $|L^f_{\tilde{S}}(A_{\mathcal{H}}(\tilde{S})) - L^f_{\tilde{S}}(GRM^f_{\mathcal{H}}(\tilde{S}))| \geq \epsilon$ with probability greater than $1 - O(e^{-\epsilon d})$.*

Proof. It is easy show that for l satisfying conditions 1-3 in the theorem statement, it holds that, when $t \in [-1, 1]$, $l(1, t)$ is monotonic non-decreasing in $-t$ while $l(-1, t)$ is monotonic non-decreasing in t. Thus, unless $l(1, t) = l(-1, t)$ for any t in the same range, $\tilde{l}$ is not convex. In, particular, either there is at least a value $t \in [-1, 1]$ where $l(1, t) = l(-1, t)$ and $\tilde{g}$ is non-smooth, or $\tilde{g}$ is unbounded in $[-1, 1]$. Then, the result follows from [50], Theorem 1, by noting that $L^f_{\tilde{S}}$ is non-convex and non-smooth (or unbounded). □

As a consequence of the above result, since the hinge loss, the log-loss and quadratic loss (as well as most other commonly adopted loss functions) all satisfy conditions 1-3 but do not satisfy Eq. (30) (see also Example 3.5), it is easy to show that popular learning algorithms such as least squares linear regression, SVM or logistic regression (which admit a polynomial-time algorithm in the supervised learning setting) do not admit a polynomial-time extension based on optimistic risk minimization.

Example 3.5. *Let $l, \tilde{l}$ be, respectively, the loss function and the corresponding imprecise loss function defined in Example 3.4. Then, by Proposition 3.1 and since l is the quadratic loss, it holds that $\tilde{l}$ is non-convex and non-smooth. Indeed, let $\tilde{m}$ be the data-generating distribution defined in Example 3.1. Assume that $g(x) = x$ (hence $w = 1$) and, thus, $f(x) = sign(x)$. Let $x = 0.7$ (thus, $f(x) = 1$) and $\eta = 0.7$: thus, $\pi(1) = 1$ and $\pi(-1) = 0.7$. Then, for $\hat{w} \in [-1, 1]$, the loss functions $l(\hat{w}, x, f(x)), \tilde{l}(\hat{w}, x, \pi)$ are depicted in Figure 1, which clearly illustrates how $\tilde{l}$ is non-convex and non-smooth.*

3.3. Learning from Fuzzy Labels: Instance-based Models

As in the case of standard supervised learning, the results given in Section 3.2 only apply to parametric models trained through optimistic risk minimization.

[7] Assume, for simplicity and without loss of generality, that $1 = \pi(1) > \pi(-1)$. Then, the imprecise loss $\tilde{l}$ defined in Eq. (29) can be expressed as

$$\tilde{g}(h, x, \pi) = (1 - \pi(-1)) \cdot g(h, x, 1) + \pi(-1) \cdot g(h, x, -1).$$

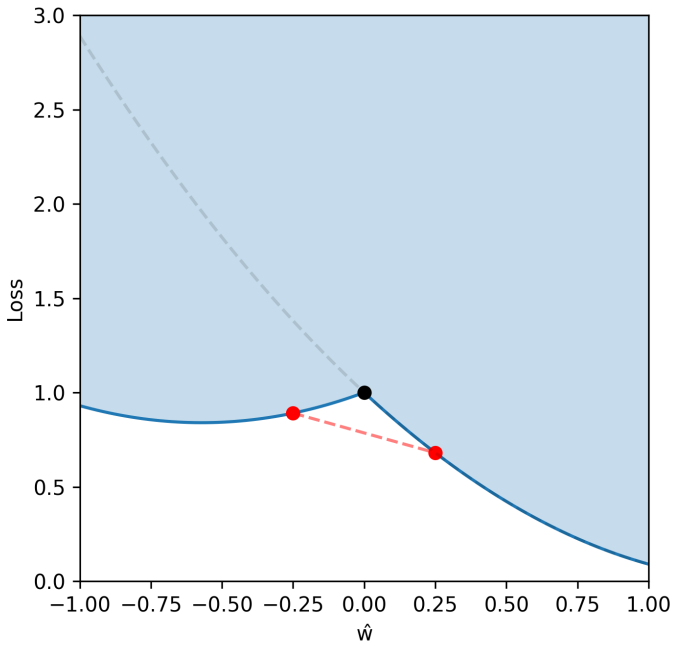


Figure 1: A graphical illustration of Proposition 3.1, for the squared loss (see Example 3.5). The dashed gray curve represents the precise squared loss l, while the blue curve represents the corresponding imprecise loss $\tilde{l}$. It is easy to observe that $\tilde{l}$ is non-convex, as illustrated by the fact that the dashed red line lies below the curve for $\tilde{l}$. Similarly, it is easy to see that $\tilde{l}$ is non-smooth, as $\tilde{l}$ is non-differentiable at the black point.

To address the above-mentioned limitations (particularly so, in regard to the computational complexity), several other approaches have been considered in the literature. Among others, instance-based methods [7, 29, 33, 35, 83] are particularly attractive as they are computationally efficient. In its simplest and most popular instantiation, this family of algorithms arises from a generalization of the nearest-neighbors learning rule to the LFL setting:

Definition 3.7. *For any instance $x \in X$ and (imprecise) training set $\tilde{S}$, denote with $N(x, \tilde{S})$ the collection of nearest neighbors of x in $\tilde{S}$. Then, generalized nearest neighbors (GNN) can be defined as:*

$$GNN(\tilde{S}, x) = \arg\max_{y \in Y} \left\{ \sum_{(x_i, \pi_i) \in N(x, \tilde{S})} \pi_i(y) \right\}. \tag{31}$$

Intuitively, GNN uses the possibility degree of each possible class y as a weight, thus favoring classes with higher possibility degree assigned to them.

Theorem 3.3, then, provides a generalization of Theorem 2.3 to the LFL setting for the GNN algorithm.

Theorem 3.3 (LFL Generalization Bound for GNN [15])**.** *Let* $X = [0,1]^d$, $Y = \{0, 1, \cdots, |Y| - 1\}$, *and* $\forall y \in Y$, *let* $\eta_y(x)$ *be defined as:*

$$\eta_y(x) = \tilde{m}(y|x) = \frac{\tilde{m} \downarrow (X \times Y)(x, y)}{\sum_{y \in Y} \tilde{m} \downarrow (X \times Y)(x, y)}.$$

Assume that $\forall y$, η_y *is c-Lipschitz. Let* $\tilde{S}$ *be an imprecise training set, with* $|\tilde{S}| = m$, *Furthermore, assume that* $\forall x, |N(x, \tilde{S})| = r$ *is a constant, independent of* d, Y, x, $\tilde{S}$, m *and* $\tilde{m}$. *Then* $\mathbb{E}[L_{\tilde{m}}(GNN(\tilde{S}))]$ *can be upper bounded by:*

$$(\alpha + 2k_* - k^*\alpha)|Y|L_{\tilde{m}}^{2-Bayes} + (1 + k^*\alpha - k_*) + (1 + \alpha + k^*\alpha)4c\sqrt{d}m^{\frac{-1}{d+1}}, \quad (32)$$

where, $\alpha, k_*, k*$ *are (respectively) the Ambiguity, Lower Knowledge and Upper Knowledge parameters for* $\tilde{m}$; *the expectation is computed w.r.t.* $\tilde{m}$; *and* $L_{\tilde{m}}^{2-Bayes} = \max_{y \in Y} L_{\tilde{m}_{\leftarrow y}}^{Bayes}$ *with* $\tilde{m}_{\leftarrow y}$ *being the distribution obtained from* $\tilde{m}$ *by applying the standard One-vs-Rest reduction*[8]

It is easy to observe that Theorem 3.3 implies that the expected risk of generalized nearest neighbors grows exponentially fast with the dimensionality of the feature space,: thus, as the number of features grows, the tendency of GNN to over-fitting similarly increases. As a consequence, solving LFL problem using GNN (and, more generally, instance-based methods) is sample-hard. While this problem can partially be addressed by applying feature selection or dimensionality reduction algorithms for LFL [18], it nonetheless shows that, in high-dimensional spaces, error rates for instance-based methods may converge very slowly to optimal ones. Similarly, it is easy to see that an impossibility theorem analogous to Theorem 3.2 holds also for GNN, as the expected error rate given in Eq. (32) depends explicitly on the parameters α, k_*, k^* of the data-generating distribution $\tilde{m}$. By contrast, compared to optimistic risk minimization, instance-based methods can be easily seen to be computationally efficient.

Proposition 3.2. *Let* X *be a* d*-dimensional vector space,* Y *be the label space and* $\tilde{S} \subset (X \times \mathcal{F}(Y))^m$ *be a training set. Let* $|N(x, \tilde{S})| = r$. *Then, the expected running time of generalized nearest-neighbors is* $O(rm|Y|d)$.

3.4. A Pseudo Label-based Approach for Learning from Fuzzy Labels

In Sections 3.2 and 3.3 we studied the learning-theoretic and complexity-theoretic properties of GRM and GNN and highlighted some relevant theoretical limitations for these two methods. Another significant limitation of optimistic risk minimization and generalized nearest-neighbors (or, more generally,

[8]Let $\mathcal{D}$ be a data-generating distribution defined on $X \times Y$, where $2 < |Y| < \infty$. Let $y \in Y$ be any specific class. Then, the One-vs-Rest reduction $\mathcal{D}_{\leftarrow y}$ of $\mathcal{D}$ is the data-generating distribution on $X \times \{0, 1\}$ defined pointwise by $\mathcal{D}_{\leftarrow y}(x, 1) = \mathcal{D}(x, y)$ and $\mathcal{D}_{\leftarrow y}(x, 0) = \sum_{y' \in Y : y' \neq y} \mathcal{D}(x, y')$.

instance-based methods) regards the fact that both these families of methods require the implementation of ad-hoc algorithms. In light of the availability of efficient, out-of-the-box implementations for many learning algorithms (in the supervised learning setting), this may be a significant limitation in terms of empirical performance. To address this limitation, in this Section we propose and study a novel algorithm based on a different class of models, namely pseudo label-based methods. Pseudo label-based learning [40, 52, 58, 80] is a generic approach for learning from weakly supervised data that is based on an iterative training process, which is summarized in Algorithm 1.

Algorithm 1 The meta-procedure for pseudo label-based learning.

procedure PSEUDO_LABEL_LEARNING(h: ML model, $\tilde{S}$: imprecise dataset, C: inclusion criterion)
  $S_0 \leftarrow$ select precise instances (x, y) from $\tilde{S}$
  $T \leftarrow \emptyset$
  while $T \neq S$ **do**
    Train h on S_i
    $T \leftarrow \{(x, h(x)) \in S : C(x, y, h) = \text{TRUE}\}$
    $S_{i+1} \leftarrow$ refine the precise instances in S_i based on h, T and $\tilde{S}$
  end while
  return h
end procedure

Intuitively, pseudo label-based learning methods involve an iteration between two different steps, that are subsequently repeated until an appropriate stopping condition is met. These two steps are as follows:

- a *selection step*, in which a precise dataset is constructed from the available imprecise training set: this amounts to associating with each instance (x, π_x) a precise *pseudo label* y';
- a *training step*, in which a standard supervised learning model is trained on the selected precise dataset, by considering the pseudo labeled instances (x, y'). Then, the labels predicted by the trained model h are used, together with an *inclusion criterion* (i.e., an algorithm $C : X \times Y \times \mathcal{H} \rightarrow \{\text{TRUE}, \text{FALSE}\}$), so as to select a subset of instances that meet some *quality requirement*[9]: these instances are considered as correctly *disambiguated* (i.e., the precise label associated with an instance x s.t. $C(x, y, h) = \text{TRUE}$ is considered *correct*). Finally, the model h, together with the set of instances T, is used to refine the precise pseudo labels.

In general the selection step may depend on the result of the training step, so that, in effect, the two steps may have a compounding effect on the final per-

[9] One common example of an inclusion criterion is to assign $C(x, y, h) = \text{TRUE}$ iff the classifier h assigns a large confidence score to label y for instance x.

formance of the trained model. In experimental comparisons [69], pseudo label-based methods have reported good empirical performance, comparable with, or better than, other state-of-the-art methods. Furthermore, pseudo label-based learning methods allow the use of efficient, out-of-the-box standard classifiers to implement the training step described above: indeed, the management of imprecise labels is performed as a pre-processing routine (in the selection step), while learning is entirely performed on precise (pseudo-labeled) instances, during the training step. Despite having these intuitively appealing characteristics, the theoretical properties of pseudo label-based methods have been investigated only in the semi-supervised learning setting [3], while the more general LFL setting has not been studied before. A possible reason for this gap may regard the complexity of studying the sequential, iterative dynamics of the more commonly used pseudo label-based learning methods (see Algorithm 1).

To address these limitations, we propose a novel pseudo label-based learning algorithm called Random Resampling-based Learning (RRL), whose pseudo-code formulation is given in Algorithm 2. RRL is based on the ensemble learning paradigm and employs a *parallel* composition of base classifiers, each of which is trained by means of a standard learning algorithm. In this section we will show that this implementation choice leads to an increased simplicity both in theoretical terms, as it allows to assume that iterations are independent of each other, as well as in computational terms, as it transforms the sequential routine in Algorithm 1 into a massively parallelizable one.

Algorithm 2 The RRL algorithm.

procedure RRL($\tilde{S}$: imprecise dataset, n: ensemble size, $\mathcal{H}$: base function class)
  $Ensemble \leftarrow \emptyset$
  for all iterations $i = 1$ to n **do**
    Draw a boostrap sample S' from $\tilde{S}$
    $Tr_i \leftarrow \emptyset$
    for all $(x, \pi) \in S'$ **do**
      Sample $\alpha \sim Uniform[0, 1]$
      Add (x, y') to Tr_i, where $y' \sim Uniform[\pi^\alpha]$
    end for
    Add base model $h_i \in \mathcal{H}$ trained on Tr_i to $Ensemble$
  end for
  return $Ensemble$
end procedure

Intuitively, the RRL algorithm can be understood as an extension of bagging-based ensemble learning to the LFL setting. In each of the bootstrap samplesm the precise pseudo labels to be associated with the imprecise instances (x, π_x) are drawn independently from a probability distribution compatible with π_x. In particular, the pseudo labels are drawn, for each $x \in X$, from the probability

distribution $\hat{Pr}_{\pi_x}$ defined by:

$$\hat{Pr}_{\pi_x}(y) = \int_0^{\pi_x(y)} \frac{d\alpha}{|\pi_x^\alpha|}. \tag{33}$$

The distribution $\hat{Pr}$ given in Eq. (33) is obtained by means of the possibility-probability transform [39] and is implemented by means of a two-stage sampling procedure (see Algorithm 2). First, an α-cut is selected uniformly at random; then, one element of the selected α-cut is drawn uniformly at random. Intuitively, this sampling procedure favors class labels having higher possibility degrees. The above-mentioned procedure is applied to obtain n bootstrap samples which are then used to train a corresponding number of base models. Finally the base models are aggregated by simple majority voting or averaging.

It is easy to observe that, from the point of view of computational complexity, the RRL algorithm is more efficient than optimistic risk minimization.

Proposition 3.3. *Let $\mathcal{H}$ be an hypothesis space and $A : (X \times Y)^\omega \mapsto \mathcal{H}$ a training algorithm whose computational time cost is upper bounded by the function $T_A : \mathbb{N} \to \mathbb{N}$. Then, given an imprecise training set $\tilde{S}$ s.t. $|\tilde{S}| = m$, and setting the ensemble size to n, the computational complexity of RRL is within $O(n(T_A(m) + m|Y|))$. In particular, if T_A is polynomially bounded, then RRL can be trained in polynomial time.*

Thus, if h can be trained in polynomial time, also RRL can be trained in polynomial time: this is in contrast with the case of optimistic risk minimization, which was shown to be NP-hard in the general case (see Theorem 3.1).

3.4.1. Learning-theoretic Properties of RRL

In regard to the generalization properties of RRL, we first note that the sampling scheme for the pseudo label can be given a formal justification, under weak assumptions about the data-generating fuzzy random set $\tilde{m}$. Such a result can be obtained by relating the distribution over labels $\hat{Pr}_{\pi_x}$ (see Eq. (33)) with the distribution over labels determined by the *imprecise Bayes classifier*.

Definition 3.8 (Imprecise Bayes Classifier)**.** *Let $\tilde{m}$ be a data-generating distribution, and let $\tilde{m} \downarrow (X \times \mathcal{F}(Y))$ be the corresponding imprecise data-generating distribution. Then, the* imprecise Bayes classifier *is defined by:*

$$f^* = \arg\min_{f \in \mathcal{M}} \mathbb{E}_{\tilde{S}} L_{\tilde{S}}(f), \tag{34}$$

where $\mathcal{M} = \{f : X \to Y : f$ measurable w.r.t. $\tilde{m} \downarrow (X \times \mathcal{F}(Y))\}$. That is, the imprecise Bayes classifier is the unique classifier with optimal performance among those that do not have access to the true labels.

Theorem 3.4. *Assume that $\tilde{m}$ satisfies the following calibration property: with probability 1 over $(x_i, y_i, \pi_i) \sim \tilde{m}$, it holds that $\tilde{m}(y_i|x_i, \pi_i) \leq \pi_i(y_i)$. Then, f^* given by $Pr(f^*(x) = y) = \hat{Pr}_{\pi_x}(y)$ is the imprecise Bayes classifier w.r.t. the l_2 loss among probability distributions and the uniform prior.*

Proof. The calibration property assumed in the statement of the theorem guarantees that, for each $x \in X$, the true probability distribution over Y lies in the credal set [45]:

$$\mathbb{P}_{x,\pi_x} = \{P \in \mathcal{P}(Y) : P(y) \leq \pi_x(y)\},$$

that is, the set of all probability distributions (over labels) that are upper bounded by the corresponding possibility degree. As a consequence of [39], Theorem 1, it follows that $\hat{Pr}_{\pi_x} \in \mathbb{P}_{x,\pi_x}$ and

$$\hat{Pr}_{\pi_x} = \arg \min_{P \in \mathbb{P}_{x,\pi_x}} \mathbb{E}_{P'}[(P' - P)^2], \tag{35}$$

where P' is selected uniformly from $\mathbb{P}_{x,\pi}$: that is, the possibility-probability transform $\hat{Pr}_{\pi_x}$ obtained from π_x is the center of mass of the credal set $\mathbb{P}_{x,\pi_x}$. We note that, since the credal set $\mathbb{P}_{x,\pi_x}$ is convex (by definition of credal set), then $\hat{Pr}_{\pi_x}$ is the unique minimizer of Eq. (35). Thus, among all possible distributions over Y, $\hat{Pr}_{\pi_x}$ is the unique one having minimal expected l_2 loss and the result follows. □

Corollary 3.2. *Assume there exists a consistent learning algorithm A for base class $\mathcal{H}$, that is, $\forall \epsilon > 0$:*

$$\lim_{m \to \infty} Pr\left[|L_{\tilde{m}}(A(S)) - L_{\tilde{m}}(\hat{f})| > \epsilon\right] = 0,$$

where, $\hat{f}$ is the Bayes classifier and the probability is w.r.t. the sampling of a (precise) training set S from the precise data-generating distribution $\tilde{m} \downarrow (X \times Y)$ (and any eventual randomization in algorithm A). Then, RRL is consistent and converges to the imprecise Bayes classifier f^*. That is, $\forall \epsilon > 0$:

$$\lim_{m \to \infty} Pr\left[|L_{\tilde{m}}(RRL(\tilde{S})) - L_{\tilde{m}}(f^*)| > \epsilon\right] = 0,$$

where the probability is w.r.t. the sampling of a training set $\tilde{S}$ from the imprecise data-generating distribution $\tilde{m} \downarrow (X \times \mathcal{F}(Y))$, and the randomization in Algorithm 2.

Proof. The result follows directly from Theorem 3.4, consistency of $\mathcal{H}$ and the definition of RRL. □

Thus, in the asymptotic regime wherein RRL is given access to the whole data-generating distribution $\tilde{m}$, Theorem 3.4 and Corollary 3.2 provide intuitive justification for the sampling scheme adopted in Algorithm 2. Indeed, the two results show that, under the above mentioned assumptions, the classifier given by RRL would be equivalent to the imprecise Bayes classifier. Nonetheless, it is easy to see that, in general, the ensemble classifier returned by RRL is not guaranteed to be the imprecise Bayes classifier, since the underlying data-generating distribution $\tilde{m}$ is unknown and in general cannot be estimated from finite samples. To address this shortcoming, we then study the generalization

properties of RRL, under two different assumptions about the base function class $\mathcal{H}$. Theorem 3.5 assumes that the base function class is a bounded convex set with finite Natarajan dimension d and that the loss function which is used to measure the accuracy of the classifiers is Lipschitz: this allows us to derive a finite sample bound based on the rich literature on random features [68].

Theorem 3.5. *Assume the base hypothesis class $\mathcal{H}$ is a bounded convex set in a Hilbert space of functions $X \to \mathbb{R}^Y$, with $sup_{x,h}|h(x)| \leq 1$ and Natarajan dimension d. Let p be the probability density over $\mathcal{H}$ determined by RRL and let $C = \min_{h\in\mathcal{H}} p(h) > 0$. Let $l : Y \times Y \to [0, 1]$ be a loss function which is L-Lipschitz w.r.t. its first argument. Then, when the RRL algorithm is executed on a imprecise training set $\tilde{S}$, with $|\tilde{S}| = m$, sampled i.i.d. from $\tilde{m}$, it returns a function $\hat{h} = \frac{1}{n}\sum_i h_i$ s.t. $|\min_{h\in\mathcal{H}} L_{\tilde{m}}(h) - L^f_{\tilde{S}}(\hat{h})|$ can be upper bounded by:*

$$(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}})\frac{|Y|L}{C}\sqrt{log\frac{6}{\delta}} + \sqrt{\frac{r \cdot ln(\frac{r|Y|^2}{\theta^2_{\tilde{m}}}) + ln\frac{3}{\delta}}{m\theta_{\tilde{m}}}} + \sqrt{\frac{K_n + ln\frac{3m}{\delta}}{2(m-1)}}, \qquad (36)$$

with probability greater than $1-\delta$ over the sampling of the $\tilde{S}$ and the randomized execution of RRL. Furthermore the term K_n in Eq. (36) depends only on $\tilde{m}$, m and $r = \max\{n, d\}$.

Proof. Since $\mathcal{H}$ is a class satisfying the assumptions given in the statement, each $h \in \mathcal{H}$ can be expressed as $h = \int_{\mathcal{H}} \alpha(f)f df$, with $\int_{\mathcal{H}} \alpha(h)dh = 1$ and $\forall h \in \mathcal{H}, \alpha(h) \geq 0$. Let $h^* = \arg\min_{h\in\mathcal{H}} L_{\tilde{m}}(h)$. Assume the learning algorithm A for $\mathcal{H}$ is deterministic, and let $S_1, \ldots, S_n$ be the bootstrap samples randomly selected in any randomized execution of RRL. Denote with $h_i = A(S_i)$ and let $h^+ = \arg\min_{h\in\mathcal{H}}\{L^f_{\tilde{S}}(h) : h = \sum_i \alpha_i h_i \wedge \sum_i \alpha_i = 1 \ \wedge \forall_i \alpha_i \geq 0\}$. Then, the generalization gap $|L_{\tilde{m}}(h^*) - L^f_{\tilde{S}}(\hat{h})|$ can be upper bounded by:

$$|L_{\tilde{m}}(h^*) - L_{\tilde{m}}(h^+)| + |L_{\tilde{m}}(h^+) - L^f_{\tilde{S}}(h^+)| + |L^f_{\tilde{S}}(h^+) - L^f_{\tilde{S}}(\hat{h})|.$$

Thus, the risk of RRL can be estimated by bounding the three terms above separately. By [68], Theorem 1, and noting that l being L-Lipschitz implies that $\tilde{l}$ is $L|Y|$-Lipschitz, the first term can be upper bounded by

$$(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}})\frac{|Y|L}{C}\sqrt{log\frac{6}{\delta}}.$$

For the second term, note that function h^+ can be expressed as a linear classifier defined over a n-dimensional feature space A, where A is the space obtained by convex combinations of functions in the ensemble returned by the RRL algorithm. Since $\mathcal{H}$ has Natarajan dimension d and is convex, the Natarajan dimension of the above-mentioned linear classifier is $r = O(\max\{n, d\})$. Thus, the second term can be bounded, by Theorem 3.1, as:

$$\sqrt{\frac{r \cdot ln(\frac{r|Y|^2}{\theta^2_{\tilde{m}}}) + ln\frac{3}{\delta}}{m\theta_{\tilde{m}}}}.$$

Finally, noting that $L^f_{\tilde{S}}(h^+) \leq L^f_{\tilde{S}}(\hat{h})$ and $L^f_{\tilde{S}} = \mathbb{E}_{S^f_i} L_{S^f_i}$, where S^f_i is a bootstrap dataset sampled i.i.d. from $\tilde{S}$, and h^+ can be written in the form $h^+ = \sum_i \alpha_i h_i$, the third term can be upper bounded by a simple argument based on PAC-Bayes learning (see [78], Theorem 1) as:

$$\sqrt{\frac{KL(\alpha||u) + ln\frac{3m}{\delta}}{2(m-1)}},$$

where α is the probabilitity distribution s.t. $P(h_i) = \alpha_i$, u is the probability distribution s.t. $P(h_i) = \frac{1}{n}$, and KL is the Kullback-Leibler divergence. Letting $K_n = \mathbb{E}_{S\sim\tilde{D}}\mathbb{E}_{h_1\sim p,\ldots,h_n\sim p} KL(\alpha||u)$, the result follows. □

Thus, Theorem 3.5 shows that, as the training set size m and the number of ensembled models n grows to infinity, RRL converges to optimistic risk minimization w.r.t. to the hypothesis class $\mathcal{H}$. Indeed, the first and last terms of Eq. (36) converge to 0 with a rate that is equivalent to the square root of the abovementioned parameters. However, while Theorem 3.5 can be applied to obtain generalization bounds for RRL with linear or kernel methods as base classifiers, the same does not hold for the case of tree-based classifiers: indeed, such classes of classifiers do not satisfy the assumptions in Theorem 3.5. Tree-based classifiers, however, are among the most commonly used base classifiers for ensemble methods, due to their computational efficiency and good performance [70].

Then, we prove an alternative result that can be applied in settings that are more similar to those considered in standard ensemble learning methods. In particular, assuming the classifiers in the ensemble are independent of each other, we derive a tail bound on the probability of error of the averaged hypothesis returned by Algorithm 2.

Theorem 3.6. *Let l_{0-1} be the 0-1 loss. Let $\mathcal{H}$ be a class of hypotheses whose Natarajan dimension is d. Let $\mathcal{H}_A \subseteq \mathcal{H}$ be the set of hypotheses returned by Algorithm 2. Let $\hat{h}$ be the function obtained by averaging the hypotheses in $\mathcal{H}_A$. Let γ_T, γ_V be defined as:*

$$\gamma_T = \max_{h\in\mathcal{H}_A} L^f_{\tilde{S}}(h) + \epsilon \leq \frac{1}{2}, \tag{37}$$

$$\gamma_V = \max_{h\in\mathcal{H}_A} L^f_v(h) + \sqrt{\frac{\log(2/\delta)}{2m^h_v}} \leq \frac{1}{2}, \tag{38}$$

where ϵ is defined as $\epsilon = 2\sqrt{\frac{2d(\ln(m_T)+\ln(|Y|))}{m_T}}$. Then, assuming the $h \in \mathcal{H}_A$ err independently, the following inequalities hold jointly with probability greater

than $1 - 2\delta$*:*

$$1 - L_D(h) \geq \frac{1}{2}(1 - \sqrt{1 - e^{\frac{-K\gamma_T^2}{1-\gamma_T^2}}}), \tag{39}$$

$$1 - L_D(h) \geq \frac{1}{2}(1 - \sqrt{1 - e^{\frac{-K\gamma_V^2}{1-\gamma_V^2}}}), \tag{40}$$

$$L_D(h) \leq e^{-n \cdot KL(\frac{1}{2}||\gamma_V)}, \tag{41}$$

where m_T *is the size of the training set,* m_v^h *is the size of the out-of-bag validation set for base classifier* h*,* $L_v^f(h)$ *is the out-of-bag error for base classifier* h*, and* $KL(a||b) = a \log \frac{a}{b} + (1-a) \log \frac{1-a}{1-b}$ *is the Kullback-Leibler divergence between two Bernoulli variables (with mean* a, b*).*

Proof. Inequality (39) follows by applying Slud's inequality [9, 75] to $\mathcal{H}_A$, by noting that $L_D(h)$ is distributed as a Bernoulli random variable whose parameter p is upper bounded by γ_T and $\hat{h}$ errs on an instance x iff at least $K/2$ hypotheses in $\mathcal{H}_A$ also err. Inequality (40) similarly follows by Slud's inequality, bounding $L_D(h)$ with the validation error derived by direct application of Hoeffding's inequality. Finally, inequality (41) follows from Chernoff's bound for binomial distributions [2] applied to the out-of-bag validation error. □

It is easy to notice that, under the assumption of independence among the base classifiers, Theorems 3.4 and 3.6 imply that, as the number of ensembled base classifiers n grows to infinity, the performance of RRL converges to that of the imprecise Bayes classifier. This result is analogous to the consistency of Random Forest in the standard supervised learning setting [8]. Nonetheless, even though widely assumed in the literature on ensemble methods [8], the assumption of independence of the base classifiers is rather strong and, in general, cannot be guaranteed to hold as n grows. Thus, when independence of the base classifier does not hold, in practice RRL may have a rate of convergence that is much smaller than exponential or may even fail to be consistent [41].

In any case, we want to highlight two differences between Theorems 3.5 and 3.6. On the one hand, the two theorems apply to different base function classes. Indeed, while Theorem 3.5 applies to convex base classes it cannot be applied to tree-based models, as mentioned above; by contrast, Theorem 3.6 cannot be directly applied to convex base classes as these latter enjoy stability guarantees [75] that violate the independence assumption. On the other hand, Theorem 3.6 directly bounds the l_{0-1} loss generalization error of RRL, while Theorem 3.5 only provides a bound in terms of a surrogate convex loss l for which, in general, it holds that $l_{0-1} \leq l$. Thus, Theorem 3.5 is less informative than Theorem 3.6 whenever the l_{0-1} loss (that is, *accuracy*) is the target metric.

Concluding this section, it is not hard to observe that the RRL algorithm provides a trade-off among the positive characteristics of instance-based methods and optimistic risk minimization. Similarly to instance-based methods, the time complexity of RRL is polynomial as long as the time required to train

the base classifiers is also polynomial. This is in contrast with optimistic risk minimization, for which the associated learning problem was shown to be, in general, computationally hard. On the other hand, RRL shares favourable risk bounds with optimistic risk minimization. Indeed, in general, the generalization error of RRL increases only polynomially with the dimensionality of the input space, whereas, by contrast, the generalization error of instance-based methods grows exponentially fast w.r.t. the dimensionality of the feature space X. Furthermore, it can easily be seen that, under the conditions of Theorem 3.5, the generalization error of RRL asymptotically tends (as the sample size m and the number of ensembled models n both grow to infinity), to the bound shown in Theorem 3.1 for optimistic risk minimization. Nonetheless, it can be noted that the above-mentioned error bounds suffer from the same limitations that were previously mentioned in Sections 3.2 and 3.3. In particular, the obtained bounds depend on hardness parameters of the data-generating distribution which in general are unknown and cannot be estimated from data. Thus, it can be difficult to apply the derived bounds in real-world settings when no information about such parameters is available. For this reason, even more so than for standard supervised ML, experimental evaluation is of paramount importance in the validation of LFL algorithms. Section 4, then, will be devoted to the assessment of state-of-the-art methods for LFL tasks.

4. Experimental Analysis

As a complementary focus to the above theoretical analysis, the aim of this section will be to discuss the empirical validation and experimental comparison of state-of-the-art LFL algorithms, based on a large benchmark suite, encompassing both synthetic and real-world datasets. We considered, in particular, the following algorithms:

- two pseudo label-based learning algorithms, namely: the RRL algorithm, described in Section 3.4, using decision trees as base model; and the state-of-the art POP algorithm (denoted as PLC), introduced in [80] (itself being a modification of the progressive identification learning algorithm [40, 58]), using a multi-layer perceptron as base model;
- two variants of instance-based methods, namely: *generalized nearest neighbors* (denoted as GNN), i.e., the instantiation of learning rule (31) where $N(x, \tilde{S})$ is the set of k nearest neighbors of x; and *generalized radius neighbors* (denoted as GRN), i.e., the instantiation of learning rule (31) where $N(x, \tilde{S})$ is the set of all instances at distance smaller than ϵ from x, for ϵ a threshold hyper-parameter. For the case of GNN, the hyper-parameter k was set to 5 neighbors[10], while, for the case of GRN, the hyper-parameter ϵ was optimized during training;

[10] This value was selected as default in analogy with the default recommended value in the scikit-learn library (see `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`).

– a hybrid pseudo label and instance-based learning method, called DELIN [4, 79, 82] (denoted as DELIN). DELIN combines a pseudo label-based learning approach for dimensionality reduction, based on linear discriminant analysis, with an instance-based classification method, based on generalized nearest neighbors. The two algorithms are iteratively and alternatively executed to improve the classification performance of a GNN classifier, by addressing the curse of dimensionality. Since the number of reduced dimensions is a hyper-parameter, this was optimized during training and validation. For the GNN classifier, as before, the number of neighbors was set to 5;

– two implementations of GRM, namely: a version of GRM based on linear SVM learning and the hinge loss as base loss (denoted as GRMSVM); and a version of GRM based on a single hidden layer multi-layer perceptron and the cross-entropy loss as base loss (denoted as GRMNN).

For all of the above-mentioned algorithms we considered the reference implementation provided in the scikit-weak [20] library[11].

All the algorithms were evaluated on contaminated versions of standard precise benchmark datasets from the UCI collection [37], as well as on real imprecise datasets. The full list of datasets is reported in Table 1. For the precise benchmark datasets two different contamination models were considered:

– fully random contamination: this contamination model represents a generalization of the random contamination model, adopted in [65] for the superset learning setting, to the LFL setting, and has been used in the context of information elicitation from questionnaires in [14, 73]. For each training instance x, we assign the correct label y a possibility degree $\pi(y) = 1$. By contrast, for each wrong label $y' \neq y$, we draw n Bernoulli random variables with success rate ϵ, denoted as $\{B_i(\epsilon)\}_{i=1}^{n}$, and define the possibility degree $\pi(y')$ as

$$\pi(y') = \frac{\sum_{i=1}^{n} B_i}{n},$$

that is, the possibility degree of y' is the number of observed successes. Equivalently, y' is associated with possibility degree $\pi(y')$ with probability:

$$Binom(\pi(y'); n, \epsilon) = \binom{n}{\lceil \pi(y') \cdot n \rceil} \epsilon^{\lceil \pi(y') \cdot n \rceil} (1-\epsilon)^{n - \lceil \pi(y') \cdot n \rceil},$$

where $Binom(\cdot; n, \epsilon)$ is the binomial probability distribution with parameters n and ϵ. Intuitively, fully random contamination can be understood as a labeling process by which n experts are asked to assess whether label y' applies to instance x, and then taking the possibility degree y' as

[11] `https://github.com/AndreaCampagner/scikit-weak`

being the (normalized) number of experts that gave a positive answer. In our experiments, we set $n = 100$ and considered different values for $\epsilon \in \{0.1, 0.25, 0.5, 0.7, 0.9\}$. We note that fully random contamination ensures that the strong superset assumption holds;

– label relaxation contamination [55]: this contamination model was proposed in the setting of learning from noisy labels [55, 54], as a generalization of *label smoothing* (a regularization approach commonly adopted in deep learning [62]). The intuitive idea of label relaxation contamination is that each precise label, which may potentially be noisy, is transformed into a fuzzy label: thus, label relaxation transforms a learning from noisy labels problem into a LFL one. In particular, in the experiments, we used a k-nearest neighbors model to implement the label relaxation process. For each training instance x, the $k \in \{3, 5, 7\}$ nearest neighbors of x (including x itself) are selected and each label y is assigned a possibility degree $\pi(y)$:

$$\pi(y) = \frac{|\{x' \in N(x, S) : (x', y) \in S\}|}{max_{y' \in Y} |\{x' \in N(x, S) : (x', y') \in S\}|}.$$

Thus, the disagreement about the labels among the nearest neighbors of x is interpreted as a measure of noisiness, and the (normalized) frequency of each label is understood as a measure of the plausibility of that label. Notice that for this contamination model the possibility degree of the correct class label y for a given instance x is always $\pi(y) > 0$ (as x itself is included in the set of its nearest neighbors) but, in general, it may happen that $\pi(y) \neq 1$: thus, only the weak superset assumption holds (but, in general, not the strong one).

For the real-world imprecise datasets, 5 different tasks were considered:

– Circulating Tumor Cells detection [19, 76, 77] from fluorescence microscopy, as an example of a learning from multi-rater problem. In particular, fuzzy labels are obtained by consensus among 11 raters: each rater provided a precise label y and the possibility degree of label y was computed as

$$\pi(y) = \frac{\text{num. of raters who proposed label } y}{\max_{y'} \text{num. of raters who proposed label } y'};$$

– COVID-19 diagnosis from routine laboratory exams [12], as an example of a learning from noisy labels problem. In this dataset, the fuzzy labels were obtained by considering the sensitivity and specificity of a RT-PCR swab test and computer imaging. In particular, let $\text{Sens(swab)}, \text{Spec(swab)}$ be the sensitivity and specificity for the swab test and $\text{Sens(img)}, \text{Spec(img)}$ be the corresponding values for the imaging test. Then, for instance x, define the evidence for the positive class as:

$$e(+1|x) = \text{Sens(swab)} \mathbb{1}_{\text{swab}(x)=+1} + \text{Sens(img)} \mathbb{1}_{\text{img}(x)=+1},$$

and similarly define the evidence for the negative class $e(-1|x)$. Then, the possibility degree $\pi(y)$ for class $y \in \{-1, +1\}$ was defined as:

$$\pi(y) = \frac{e(y|x)}{\max_{y' \in \{-1,1\}} e(y'|x)};$$

– Knee lesion detection [13] from magnetic resonance imaging, as a second example of a learning from multi-rater problem. In this case the fuzzy labels are obtained by confidence-weighted consensus among 12 raters: that is, if rater r_i associated a confidence of $c_i(y)$ to label y, then

$$\pi(y) = \frac{\frac{1}{12}\sum_{i=1}^{12} c_i(y)}{\max_{y'} \frac{1}{12}\sum_{i=1}^{12} c_i(y')};$$

– Spine surgery invasiness prediction [16], as an example of a semi-supervised learning task. In this a single rater labeled all instances as either non-invasive, invasive or uncertain;

– Sagittal misalignment assessment [17], as an example of a superset learning task. In this case, two medical specialists annotated all the instances in the datasets, and the sets of labels were obtained by simply selecting, for each instance x, all the labels associated with x.

All algorithms were evaluated in a 10-repeated 5-fold cross-validation experimental setting, to take into account sensitivity to initialization and randomization. In particular, all models were evaluated in terms of balanced accuracy, in order to measure the models' error rate also under conditions of label imbalance, and running time (in ms), as a measure of computational efficiency. For the synthetically contaminated datasets, balanced accuracy was evaluated by comparison with the known ground truth labeling (which was not available to the learning algorithms during training). For the real-world imprecise datasets, instead, balanced accuracy was evaluated on a subset of the data whose labels were precise. That is: for the ctc, mri and spine datasets, the test sets encompassed only instances on which all raters proposed the same label; for the covid dataset, the test set encompassed only instances on which the two diagnostic tests provided the same diagnosis; while for the invasiveness dataset the test set encompassed only instances rated as invasive or non-invasive. Statistical analysis of the results was performed by means of a ranking-based comparison, using Friedman test with Wilcoxon post-hoc procedure [6].

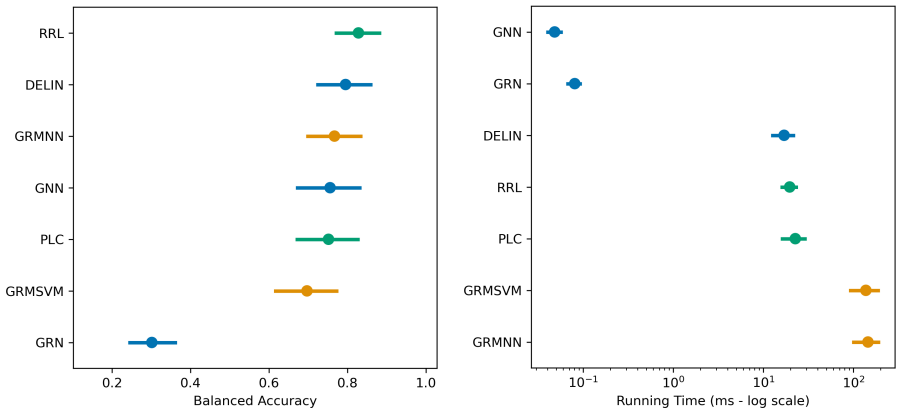
Results of the experimental analysis are reported in Figures 2a and 3, in terms of balanced accuracy, and Figures 2b and 4, in terms of running time.

In terms of balanced accuracy, the three best models were RRL, DELIN and GRMNN. In particular, RRL was the best algorithm in terms of both raw balanced accuracy as well as average ranks. Furthermore. even though the performance of RRL and DELIN were not statistically significantly different, RRL reported better performance on average and it was also significantly better than all other considered algorithms. Similarly, no significant difference

Table 1: List of datasets considered in the experimental comparison of LFL algorithms.

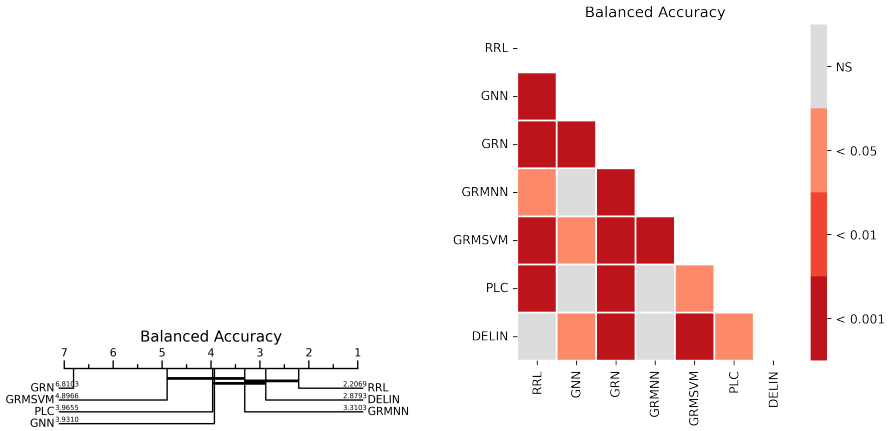
	Classes	Features	Instances
UCI datasets			
avila	10	10	20768
banknote	2	4	1372
cancerwisconsin	2	9	683
car	4	16	864
credit	2	61	1000
crowd	6	28	10845
diabetes	2	8	768
digits	10	62	5620
frog-family	4	22	7195
frog-genus	8	22	7195
frog-species	10	22	7195
hcv	4	12	582
htru	2	8	17898
ionosfera	2	33	351
iranian	2	45	7032
iris	3	4	150
mice	8	78	972
mushroom	6	99	5644
myocardial	2	111	1700
obesity	7	31	2111
occupancy	2	5	20560
pen	10	16	10992
robot	4	24	5456
sensorless	11	48	20000
shill	2	9	6321
sonar	2	60	208
vowel	11	9	990
wifi	4	7	2000
wine	3	13	178
Imprecise Datasets			
ctc	2	2500	617
covid	2	69	1624
mri	2	100	427
invasiveness	3	186	72
spine	7	14	120

was detected among DELIN and GRMNN, as well as between GRMNN, GNN and PLC. These results confirm the good performance of RRL, which can then be related with the theoretical results demonstrated in Section 3.4. Indeed, RRL had results comparable with those of GRMNN and DELIN, respectively an optimistic risk minimization and a (dimensionality reduced) instance-based method. Interestingly, however, the proposed RRL algorithm reported better performance, on average, than the other two methods. Also this difference could be explained by referring to the theoretical results shown in Section 3. For the case of GRMNN, the fact that solving the optimistic risk minimization problem is NP-hard may lead to premature convergence to either local minima or saddle points, and, consequently, to sub-optimal generalization error. For the case of DELIN, by contrast, even though this algorithm performs a data dimensionality pre-processing step to reduce the risk of over-fitting of GNN,



(a) (b)

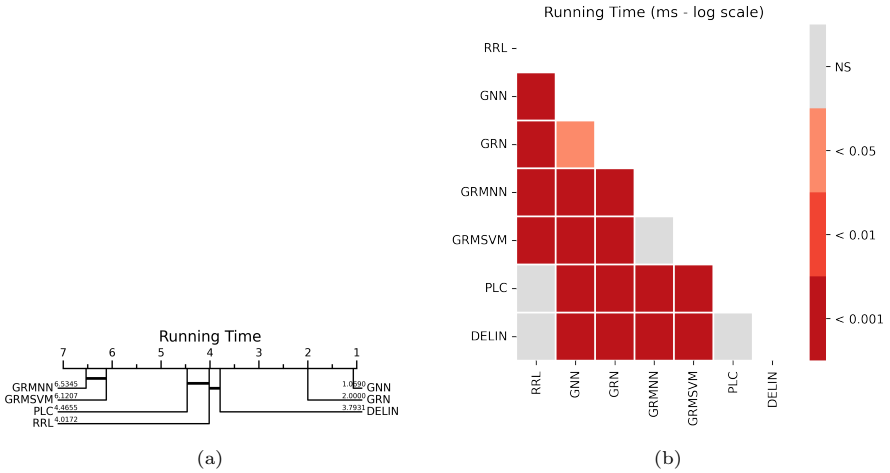
Figure 2: Results of the experiments. Left: mean balanced accuracy scores of the models under study (higher is better), Error bars denote 95% C.I. Mean running times (ms) of the models under study (lower is better). Error bars denote 95% C.I. Legend, okra: generalized risk minimization based, green: pseudo label-based learning based, blue: instance-based methods.



(a) (b)

Figure 3: Comparison of the the models under study in terms of balanced accuracy. Left: critical difference diagram of the mean ranks (lower is better), bars denote significance cliques at 95% confidence level. Right: heatmap of p-values obtained with the post-hoc Friedman-Nemenyi test, significance at different thresholds is denoted with shades of red. For each significant comparison in the right side, the best method in the corresponding pair of models can be assessed from the left side, by looking at which of the two models had a lower mean rank.

the number of reduced dimension may still be too large to avoid the curse of dimensionality. Indeed, in the experiments the number of reduced dimension was dynamically optimized during cross-validation, thus leading to a possible



(a) (b)

Figure 4: Comparison of the the models under study in terms of running time.Left: critical difference diagram of the mean ranks (lower is better), bars denote significance cliques at 95% confidence level. Right: heatmap of p-values obtained with the post-hoc Friedman-Nemenyi test, significance at different thresholds is denoted with shades of red. For each significant comparison in the right side, the best method in the corresponding pair of models can be assessed from the left side, by looking at which of the two models had a lower mean rank.

over-fitting of this hyper-parameter. By contrast, the worst performing algorithm was GRN, which reported significantly lower performance than all the other considered methods. Interestingly, GRMNN reported significantly better performance than GRMSVM, likely due to the fact that most of the considered datasets did not satisfy the linear separability assumption required for the good functioning of the linear SVM model underlying GRMSVM.

In terms of running time, the best performing algorithm was GNN, which was significantly more computationally efficient than all other considered algorithms, with the exception of GRN. This result is expected: indeed, the training time of lazy instance-based methods such as GNN and GRN is typically constant (or, at most, log-linear) in the size of the training set. By contrast, the two worst performing algorithms were both generalized risk minimization methods, namely GRMNN and GRMSVM: these two algorithms were significantly less computationally efficient than all the other considered algorithms. This result, on the one hand, confirms the general hardness of these learning algorithm (see Theorem 3.1); on the other hand, it can be remarked that memory transfer bottlenecks in the scikit-weak implementation of these algorithms (which offloads tensor processing operations to GPU execution) could also have a role in the observed performance gap. Further research should be devoted at decomposing these two computing costs, and possibly optimizing memory usage. The proposed RRL algorithm reported a running time which was intermediate between those of instance-based methods and generalized risk minimization ones: in particular, RRL had an average running time comparable with (i.e.,

not significantly different from) that of DELIN.

Thus, the experimental results show the effectiveness of the proposed RRL algorithm: indeed, the proposed approach reported a running time which was comparable, or better than, other state-of-the-art methods for LFL, while at the same time exhibiting the best generalization accuracy among the compared methods. These results, furthermore, are complemented by the generalization guarantees for RRL which were proved in Section 3.4.

5. Conclusion

The aim of this article was to study the problem of learnability in the LFL setting. To this aim, we first analyzed the generalization ability, as well as the computational complexity, of two of the main learning paradigms in this setting, namely: instance-based methods and generalized risk minimization. Furthermore, the second main contribution consists in the proposal of a novel pseudo label-based learning algorithm, called RRL, and the study of its statistical and generalization properties. To our knowledge, this is the first theoretical investigation of the pseudo label-based learning paradigm in the setting of learning from imprecise data. These theoretical contributions have then been complemented with a third, experimental, contribution through which we compared the performance (in terms of generalization accuracy and running time) of several state-of-the-art methods for LFL. In particular, our results show the effectiveness of the proposed RRL algorithm, and thus confirm and reinforce the presented theoretical analysis. We believe these results to be particularly interesting, as they show how the interaction between uncertainty representation theories and machine learning could lead to the development of novel and effective algorithmic approaches: indeed, we showed that the construction method for RRL (which directly relies on well-known results in possibility theory) allows to achieve performance and theoretical properties comparable with, and better than, the state-of-the-art. In light of these results and contributions, the following open problems could be worthy of further research.

- From an empirical perspective, the performance gap reported by generalized risk minimization algorithms, despite being consistent with the hardness of the associated optimization problems, could also be attributed to costs related to GPU usage. Further work should be devoted at optimizing resource usage to improve the efficiency of these algorithms.
- Several theoretical characterizations of LFL paradigms, namely generalized risk minimization, instance-based methods and pseudo label-based learning, have been considered, focusing on the establishment of upper bounds on the learnability of this setting: further work should be devoted at exploring tighter bounds, especially under constraining assumptions on the problem instances, as well as at proving matching lower bounds.
- In regard to GRM, we explicitly focused on the case of optimistic risk minimization. However, other alternative approaches have been proposed [48],

chiefly among them pessimistic risk minimization [43], which has been previously proposed as a way to enable less cautious predictions and improve robustness to noisy labels. Future work should explore the theoretical properties of these alternative instantiations of the GRM paradigm.

– Finally, in this article we focused on the LFL setting: despite it being a practically relevant and natural setting for weakly supervised learning, future research should investigate theoretical characterizations, as well as practical algorithms, for more general forms of imprecise data [36, 46]. To this aim, two particularly promising research directions regard the problem of learning from fuzzy data, as well as the problem of learning from comparative probabilities [21]. On the one hand, the study of the problem of learning from fuzzy data would extend the applicability of the proposed RRL algorithm, as well as of other state-of-the-art methods for LFL, to more general settings in which imprecision affects not only the target supervision, but also the feature values. On the other hand, the problem of learning from comparative probabilities represents a particularly interesting conceptual generalization of LFL, due to the relationship between comparative probabilities and the theory of credal sets [60].

More generally, we believe that further interaction between machine learning (specifically, learning theory) and uncertainty representation theories would enable the study of more realistic and complex learning problems, involving different forms of uncertainty that may affect the data, as well as enable the design of simple, yet effective, learning algorithms.

References

[1] Angluin, D., & Laird, P. (1988). Learning from noisy examples. *Machine Learning*, *2*, 343–370.

[2] Arratia, R., & Gordon, L. (1989). Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology*, *51*, 125–131.

[3] Balcan, M.-F., & Blum, A. (2010). A discriminative model for semi-supervised learning. *Journal of the ACM (JACM)*, *57*, 1–46.

[4] Bao, W.-X., Hang, J.-Y., & Zhang, M.-L. (2021). Partial label dimensionality reduction via confidence-based dependence maximization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 46–54). ACM.

[5] Bartlett, P. L., & Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, *3*, 463–482.

[6] Benavoli, A., Corani, G., & Mangili, F. (2016). Should we really use post-hoc tests based on mean-ranks? *The Journal of Machine Learning Research*, *17*, 152–161.

[7] Bezdek, J. C., Chuah, S. K., & Leep, D. (1986). Generalized k-nearest neighbor rules. *Fuzzy Sets and Systems*, *18*, 237–256.

[8] Biau, G., Devroye, L., & Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *The Journal of Machine Learning Research*, *9*, 2015–2033.

[9] Boucheron, S., Lugosi, G., & Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press.

[10] Bshouty, N. H., Eiron, N., & Kushilevitz, E. (2002). PAC learning with nasty noise. *Theoretical Computer Science*, *288*, 255–275.

[11] Cabannes, V., Bach, F., & Rudi, A. (2021). Disambiguation of weak supervision with exponential convergence rates. *arXiv preprint arXiv:2102.02789*, .

[12] Cabitza, F., Campagner, A., Ferrari, D., Di Resta, C., Ceriotti, D., Sabetta, E., Colombini, A., De Vecchi, E., Banfi, G., Locatelli, M. et al. (2021). Development, evaluation, and validation of machine learning models for COVID -19 detection based on routine blood tests. *Clinical Chemistry and Laboratory Medicine (CCLM)*, *59*, 421–431.

[13] Cabitza, F., Campagner, A., & Sconfienza, L. M. (2020). As if sand were stone. New concepts and metrics to probe the ground on which to build trustable AI. *BMC Medical Informatics and Decision Making*, *20*, 1–21.

[14] Cabitza, F., & Ciucci, D. (2018). Fuzzification of ordinal classes. The case of the HL7 severity grading. In *Scalable Uncertainty Management: 12th International Conference, SUM 2018, Milan, Italy, October 3-5, 2018, Proceedings 12* (pp. 64–77). Springer.

[15] Campagner, A. (2021). Learnability in "learning from fuzzy labels". In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–6). IEEE.

[16] Campagner, A., Berjano, P., Lamartina, C., Langella, F., Lombardi, G., & Cabitza, F. (2020). Assessment and prediction of spine surgery invasiveness with machine learning techniques. *Computers in Biology and Medicine*, *121*, 103796.

[17] Campagner, A., Cabitza, F., Berjano, P., & Ciucci, D. (2021). Three-way decision and conformal prediction: Isomorphisms, differences and theoretical properties of cautious learning approaches. *Information Sciences*, *579*, 347–367.

[18] Campagner, A., & Ciucci, D. (2022). Rough-set based genetic algorithms for weakly supervised feature selection. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2022)* (pp. 761–773). Springer.

[19] Campagner, A., Ciucci, D., Svensson, C.-M., Figge, M. T., & Cabitza, F. (2021). Ground truthing from multi-rater labeling with three-way decision and possibility theory. *Information Sciences*, *545*, 771–790.

[20] Campagner, A., Lienen, J., Hüllermeier, E., & Ciucci, D. (2022). Scikit-Weak: A Python library for weakly supervised machine learning. In *Rough Sets: International Joint Conference, IJCRS 2022, Suzhou, China, November 11–14, 2022, Proceedings* (pp. 57–70). Springer.

[21] Capotorti, A., & Formisano, A. (2008). Comparative uncertainty: theory and automation. *Mathematical Structures in Computer Science*, *18*, 57–79.

[22] Cour, T., Sapp, B., & Taskar, B. (2011). Learning from partial labels. *The Journal of Machine Learning Research*, *12*, 1501–1536.

[23] Couso, I., Borgelt, C., Hullermeier, E., & Kruse, R. (2019). Fuzzy sets in data analysis: From statistical foundations to machine learning. *IEEE Computational Intelligence Magazine*, *14*, 31–44.

[24] Couso, I., & Dubois, D. (2018). A general framework for maximizing likelihood under incomplete data. *International Journal of Approximate Reasoning*, *93*, 238–260.

[25] Couso, I., Dubois, D., & Hüllermeier, E. (2017). Maximum likelihood estimation and coarse data. In *Scalable Uncertainty Management: 11th International Conference, SUM 2017, Granada, Spain, October 4-6, 2017, Proceedings 11* (pp. 3–16). Springer.

[26] Couso, I., Dubois, D., & Sánchez, L. (2014). *Random sets and random fuzzy sets as ill-perceived random variables*. Springer.

[27] Daniely, A., Sabato, S., Ben-David, S., & Shalev-Shwartz, S. (2011). Multiclass learnability and the ERM principle. In *Proceedings of the 24th Annual Conference on Learning Theory* (pp. 207–232). JMLR Workshop and Conference Proceedings.

[28] Dempster, A. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, *38*, 325–339.

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

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

[31] Denoeux, T. (2021). Belief functions induced by random fuzzy sets: A general framework for representing uncertain and fuzzy evidence. *Fuzzy Sets and Systems*, *424*, 63–91.

[32] Denœux, T., Dubois, D., & Prade, H. (2020). Representations of uncertainty in artificial intelligence: Probability and possibility. In *A Guided Tour of Artificial Intelligence Research* (pp. 69–117). Springer.

[33] Denoeux, T., Kanjanatarakul, O., & Sriboonchitta, S. (2019). A new evidential k-nearest neighbor rule based on contextual discounting with partially supervised learning. *International Journal of Approximate Reasoning*, *113*, 287–302.

[34] Denœux, T., & Zouhal, L. M. (2001). Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, *122*, 409–424.

[35] Derrac, J., García, S., & Herrera, F. (2014). Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects. *Information Sciences*, *260*, 98–119.

[36] Destercke, S. (2022). Uncertain data in learning: challenges and opportunities. In *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications* (pp. 322–332). PMLR.

[37] Dua, D., & Graff, C. (2017). UCI machine learning repository. URL: `http://archive.ics.uci.edu/ml`.

[38] Dubois, D., & Prade, H. (1998). Possibility theory: qualitative and quantitative aspects. In *Quantified representation of uncertainty and imprecision* (pp. 169–226). Springer.

[39] Dubois, D., Prade, H., & Sandri, S. (1993). On possibility/probability transformations. In *Fuzzy Logic: State of the Art* (pp. 103–112). Springer.

[40] Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., & Sugiyama, M. (2020). Provably consistent partial-label learning. *Advances in Neural Information Processing Systems*, *33*, 10948–10960.

[41] Ferreira, J. A. (2022). Models under which random forests perform badly; consequences for applications. *Computational Statistics*, *37*, 1839–1854.

[42] Grabisch, M. (2016). *Set functions, games and capacities in decision making* volume 46 of *Theory and Decision Library C*. Springer.

[43] Guillaume, R., & Dubois, D. (2015). Robust parameter estimation of density functions under fuzzy interval observations. In *9th International Symposium on Imprecise Probability: Theories and Applications (ISIPTA'15)* (pp. 147–156). Aracne.

[44] Guillaume, R., & Dubois, D. (2018). A maximum likelihood approach to inference under coarse data based on minimax regret. In *Uncertainty Modelling in Data Science. SMPS 2018* (pp. 99–106). Springer volume 832 of *Advances in Intelligent Systems and Computing*.

[45] Hose, D., & Hanss, M. (2021). A universal approach to imprecise probabilities in possibility theory. *International Journal of Approximate Reasoning*, *133*, 133–158.

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

[47] Hüllermeier, E. (2015). Does machine learning need fuzzy logic? *Fuzzy Sets and Systems*, *281*, 292–299.

[48] Hüllermeier, E., Destercke, S., & Couso, I. (2019). Learning from imprecise data: Adjustments of optimistic and pessimistic variants. In *Scalable Uncertainty Management - 13th International Conference, SUM 2019, Compiègne, France, December 16-18, 2019, Proceedings* (pp. 266–279). Springer volume 11940 of *Lecture Notes in Computer Science*.

[49] Jin, R., & Ghahramani, Z. (2003). Learning with multiple labels. *Advances in Neural Information Processing Systems*, *15*, 921–928.

[50] Kornowski, G., & Shamir, O. (2021). Oracle complexity in nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, *34*, 324–334.

[51] Kuncheva, L. (2000). *Fuzzy classifier design*. Springer Science & Business Media.

[52] Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*.

[53] Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., & Li, L.-J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1910–1918). IEEE.

[54] Lienen, J., & Hüllermeier, E. (2021). From label smoothing to label relaxation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 8583–8591). AAAI.

[55] Lienen, J., & Hüllermeier, E. (2021). Instance weighting through data imprecisiation. *International Journal of Approximate Reasoning*, *134*, 1–14.

[56] Liu, L., & Dietterich, T. (2014). Learnability of the superset label learning problem. In *Proceedings of the 31st International Conference on Machine Learning* (pp. 1629–1637). PMLR.

[57] Liu, L., & Dietterich, T. G. (2012). A conditional multinomial mixture model for superset label learning. *Advances in Neural Information Processing Systems*, *25*, 548–556.

[58] Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., & Sugiyama, M. (2020). Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning* (pp. 6500–6510). PMLR.

[59] Ma, G., Liu, F., Zhang, G. et al. (2021). Learning from imprecise observations: An estimation error bound based on fuzzy random variables. In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–8). IEEE.

[60] Miranda, E., & Destercke, S. (2015). Extreme points of the credal sets generated by comparative probabilities. *Journal of Mathematical Psychology*, *64*, 44–57.

[61] Molchanov, I. (2005). *Theory of random sets*. Springer.

[62] Müller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help? *Advances in neural information processing systems*, *32*.

[63] Natarajan, B. K. (1989). On learning sets and functions. *Machine Learning*, *4*, 67–97.

[64] Natarajan, N., Dhillon, I. S., Ravikumar, P. K., & Tewari, A. (2013). Learning with noisy labels. *Advances in neural information processing systems*, *26*.

[65] Nguyen, V.-L. (2018). *Imprecision in Machine Learning Problems*. Ph.D. thesis Université de Technologie de Compiègne.

[66] Poyiadzi, R., Bacaicoa-Barber, D., Cid-Sueiro, J., Perello-Nieto, M., Flach, P., & Santos-Rodriguez, R. (2022). The weak supervision landscape. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)* (pp. 218–223). IEEE.

[67] Quost, B., Denoeux, T., & Li, S. (2017). Parametric classification with soft labels using the evidential em algorithm: linear discriminant analysis versus logistic regression. *Advances in Data Analysis and Classification*, *11*, 659–690.

[68] Rahimi, A., & Recht, B. (2008). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in neural information processing systems*, *21*.

[69] Rizve, M. N., Duarte, K., Rawat, Y. S., & Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, .

[70] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*, e1249.

[71] Sakai, H., Liu, C., Nakata, M., & Tsumoto, S. (2016). A proposal of a privacy-preserving questionnaire by non-deterministic information and its analysis. In *IEEE Big Data 2016* (pp. 1956–1965). IEEE.

[72] Schmarje, L., Brünger, J., Santarossa, M., Schröder, S.-M., Kiko, R., & Koch, R. (2020). Beyond cats and dogs: Semi-supervised classification of fuzzy labels with overclustering. *arXiv preprint arXiv:2012.01768*, .

[73] Seveso, A., Campagner, A., Ciucci, D. et al. (2020). Ordinal labels in machine learning: a user-centered approach to improve data validity in medical settings. *BMC Medical Informatics and Decision Making*, *20*, 1–14.

[74] Shafer, G. (1976). *A mathematical theory of evidence*. Princeton University Press.

[75] Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

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

[77] Svensson, C.-M., Krusekopf, S., Lücke, J., & Thilo Figge, M. (2014). Automated detection of circulating tumor cells with naive bayesian classifiers. *Cytometry Part A*, *85*, 501–511.

[78] Tolstikhin, I. O., & Seldin, Y. (2013). PAC-Bayes-empirical-Bernstein inequality. *Advances in Neural Information Processing Systems*, *26*.

[79] Wu, J.-H., & Zhang, M.-L. (2019). Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 416–424). ACM.

[80] Xu, N., Lv, J., Liu, B., Qiao, C., & Geng, X. (2022). Progressive purification for instance-dependent partial label learning. *arXiv preprint arXiv:2206.00830*, .

[81] Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, *1*, 3–28.

[82] Zhang, M.-L., Wu, J.-H., & Bao, W.-X. (2022). Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *16*, 1–18.

[83] Zheng, K., Fung, P. C., & Zhou, X. (2010). K-nearest neighbor search for fuzzy objects. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 699–710). ACM.

[84] Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review*, *5*, 44–53.

[85] Zhou, Z.-H., Sun, Y.-Y., & Li, Y.-F. (2009). Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 1249–1256). ACM.