Machine Learning (2025) 114:19
https://doi.org/10.1007/s10994-024-06703-y

Credal ensembling in multi-class classification

Vu-Linh Nguyen[1] · **Haifei Zhang**[1,2] · **Sébastien Destercke**[1]

Received: 4 March 2024 / Revised: 6 November 2024 / Accepted: 15 November 2024 /
Published online: 16 January 2025

Abstract
In this paper, we present a formal framework to (1) aggregate probabilistic ensemble members into either a representative classifier or a credal classifier, and (2) perform various decision tasks based on this uncertainty quantification. We first elaborate on the aggregation problem under a class of distances between distributions. We then propose generic methods to robustify uncertainty quantification and decisions, based on the obtained ensemble and representative probability. To facilitate the scalability of the proposed framework, for all the problems and applications covered, we elaborate on their computational complexities from the theoretical aspects and leverage theoretical results to derive efficient algorithmic solutions. Finally, relevant sets of experiments are conducted to assess the usefulness of the proposed framework in uncertainty sampling, classification with a reject option, and set-valued prediction-making.

Keywords Credal ensembling · Quantile-based approach · Uncertainty sampling · Classification with a reject option · Set-valued prediction

1 Introduction

Model ensembling has a long history during which it has shown advantages in different applications, including but not limited to robust/accurate decision-making under the presence of noisy and insufficient data (Breiman, 1996; Dietterich, 2000; Khoshgoftaar et al., 2010; Nguyen et al., 2020), enhancing Bayes optimal prediction-making under generalized

Editors: Rita P. Ribeiro, Ana Carolina Lorena, Albert Bifet.

✉ Vu-Linh Nguyen
vu-linh.nguyen@hds.utc.fr

Haifei Zhang
haifei.zhang@hds.utc.fr

Sébastien Destercke
sebastien.destercke@hds.utc.fr

[1] UMR CNRS 7253, Heudiasyc, Sorbonne Universite, Université de Technologie de Compiègne, Compiègne, France

[2] UMR CNRS 5516, Laboratoire Hubert Curien, Université Jean Monnet, Saint-Étienne, France

losses for cautious (set-valued) predictions (Nguyen & Hüllermeier, 2021), prediction-making under imprecise probability (IP) decision rules (Nguyen et al., 2023b; Zhang et al., 2023) and uncertainty quantification (Hüllermeier & Waegeman, 2021; Shaker & Hüllermeier, 2020).

In this paper, we present a model ensembling framework to (1) aggregate the probabilistic ensemble members into either a representative classifier, which provides reliable estimates of class probabilities, or a credal classifier, which provides reliable estimates of a predictive *credal sets* (Levi, 1983), and (2) do decision-related uncertainty quantification, where we aim to robustify traditional probabilistic uncertainty measures, such as confidence level, smallest margin and entropy (Nguyen et al., 2022). We also present its applications in uncertainty sampling (Nguyen et al., 2022) and classification with a reject option (Chow, 1970; Condessa et al., 2017), where we use the robustified uncertainty measures, and set-valued prediction-making (Jansen et al., 2022; Troffaes, 2007), where we use robustified decision rules and the obtained credal set estimate. To facilitate the scalability of the proposed framework, for all the problems and applications covered, we first elaborate on their computational complexities from the theoretical aspects and then leverage theoretical results to derive efficient algorithmic solutions.

Note that there are nowadays many credal classifiers extending classical classifiers such as pairwise classifiers (Quost & Destercke, 2018), discriminant analysis (Alarcon & Destercke, 2021), naive Bayes (Corani & Zaffalon, 2008) and its tree extensions (Corani & De Campos, 2010), and decision trees (Abellan & Masegosa, 2012). Some of them also extend ensembling techniques, but concern very specific ensemble techniques such as random forests (Zhang et al., 2023; Abellán et al., 2017) or Bayesian model averaging (Corani & Antonucci, 2014), and adopts what we will later call a distort-then-aggregate approach, meaning that they make imprecise parts of the aggregation procedures, be it the predicted probabilities (e.g., in the case of the credal random forest) or the weights associated to the different members (e.g., in the case of credal model averaging). In contrast, the approach we propose here can be termed as an aggregate-then-distort approach, where we first find a representative, aggregated probability distribution and build a credal set around it. This has the advantage of being computationally quite efficient (as we will discuss), as well as being a plug-in approach to any ensembling technique where each ensemble member produces a probability distribution. To our knowledge, we are the first to study the problem of estimating credal sets from the output of ensembles, from which the various IP decision rules can be directly used to make set-valued predictions.

After providing in Sect. 2 a minimal description of probabilistic classification and classification with sets of probabilities, Sect. 3 tackles the problem of aggregating the probabilistic ensemble members into a representative classifier. The representative classifier is defined as a probabilistic classifier that minimizes the expected average distance to the ensemble members. We show that this aggregation problem can be solved in an instance-wise manner, whose solutions can be derived either analytically or via solving (convex) optimization problems, depending on the chosen distance. The problem of aggregating the probabilistic ensemble members into a reliable and robust estimate in the form of a *credal set* (Levi, 1983), that is a (convex) set of probabilities, is formulated as finding a neighborhood of the representative classifier that is expected to be informative and at the same time not very large. This credal set can, in turn, be used with theoretically justified decision rules (Jansen et al., 2022; Troffaes, 2007) to produce set-valued predictions. In Sect. 4, we implement this idea under different viewpoints, including ϵ-contamination (Bock et al., 2014) and quantile-based distortion (Nguyen et al., 2023b), and elaborate on how and where each implementation may be (dis)advantageous.

Section 5 is devoted to decision-related uncertainty quantification. We propose a generic method in which probabilistic uncertainty measures are robustified by using their (empirical) expectation over the robustified admissible region of the most probable class. In practice, this is done by an averaging procedure over the consensual ensemble members that agree on the most probable class. However, the key step of finding the robustified admissible region requires one to track the growth of admissible sets (Jansen et al., 2022; Troffaes, 2007), whose naive computation requires solving (a possibly huge number of) linear programs (Jansen et al., 2022; Nguyen et al., 2023b). We shall therefore show that computing the empirical expectation over this region using ensemble members can be done in $O(M(K + \log(M))$, where M and K are respectively the ensemble size and the output size, without having to solve any linear program.

Section 6 presents relevant sets of experiments to assess the usefulness of the proposed framework in uncertainty sampling, classification with a reject option, and set-valued prediction making, followed by a summary and an outlook on future work in Sect. 7. For uncertainty sampling, empirical evidence confirms that the robustified uncertainty measures proposed in Sect. 5 provide informative stopping rules, allowing for a nice trade-off between the gained accuracy and the used budget. They also informatively reflect the uncertainty level when varying the budget. For classification with a reject option, the robustified measures are shown to be effective in reflecting the uncertainty level when sliding the acceptance rate, and when being used as the threshold. For set-valued prediction-making, the proposed aggregate-then-distort predictors, i.e., credal classifiers constructed in Sect. 4, and cautious random forest (CRF) (Zhang et al., 2023), which is a competitive distort-then-aggregate predictor, are compared using the u_{65} score (Zaffalon et al., 2012). The experimental results indicate that the aggregate-then-distort predictors may perform differently and (slightly) better than the distort-then-aggregate predictor. Section 6 also includes discussions on the potential advantages of the proposed framework in tackling imbalanced data sets.

This paper is a significant extension of an earlier conference version (Nguyen et al., 2023b), in which the aggregation problems have originally been introduced and studied for selected distances, and are then applied only to set-valued prediction-making. The current version is more comprehensive in multiple ways, especially regarding the class of distances covered, the algorithmic procedure, the experimental evaluation, and the range of applications. The part concerning uncertainty measures and their use in rejection or active learning procedures is also completely new. All the proofs of formal results stated in this paper (propositions and remarks) can be found in the appendices.

We can summarise our main contributions as follows:

- we provide a full study of how to compute representative probability distributions and estimate credal sets from the output of ensembles with many classical distances;
- we detail an efficient way to derive from it a quantile-based credal set;
- we show that making inferences or uncertainty quantification from these sets can be done efficiently;
- we show that our approach displays performant results for various machine-learning tasks where uncertainty plays a key role, such as active learning by uncertainty sampling, classification with a reject option, and set-valued classification.

2 Preliminary

This section recalls the basics of probabilistic classification, classification with sets of probabilities, and introduces notations. Notations and acronyms are also listed in "Appendix 1".

2.1 Probabilistic classification

Let $\mathcal{X}$ denote an instance space, and let $\mathcal{Y} = \{y^1, \dots, y^K\}$ be a finite set of classes. We assume that an instance $\boldsymbol{x} \in \mathcal{X}$ is (probabilistically) associated with members of $\mathcal{Y}$. We denote by $\boldsymbol{p}(Y \,|\, \boldsymbol{x})$ the conditional distribution of Y given $\mathbf{X} = \boldsymbol{x}$. Given training data $\mathcal{D} = \{(\boldsymbol{x}_n, y_n) | n = 1, \dots, N\}$ drawn independently from $\boldsymbol{p}(\mathbf{X}, Y)$, the goal in (multi-class) classification is to learn a classifier $\boldsymbol{h}$, which is a mapping $\mathcal{X} \longrightarrow \mathcal{Y}$ that assigns to each instance $\boldsymbol{x} \in \mathcal{X}$ a (most relevant) class $\hat{y} := \boldsymbol{h}(\boldsymbol{x}) \in \mathcal{Y}$.

To evaluate the performance of a classifier $\boldsymbol{h}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$ is needed, which compares a prediction $\hat{y}$ with a ground-truth y. Each classifier $\boldsymbol{h}$ is evaluated using its expected loss

$$R(\boldsymbol{h}) := \mathbf{E}\big[\ell(Y, \boldsymbol{h}(\mathbf{X}))\big] = \int \ell(y, \boldsymbol{h}(\boldsymbol{x}))\, d\,\mathbf{P}(\boldsymbol{x}, y)\,, \tag{1}$$

where $\mathbf{P}$ is the joint probability measure on $\mathcal{X} \times \mathcal{Y}$ characterizing the underlying data-generating process. Therefore, the Bayes-optimal classifier is given by

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} R(\boldsymbol{h})\,, \tag{2}$$

where $\mathcal{H}$ is some hypothesis space (Vapnik, 1999) from which we pick $\boldsymbol{h}$. When $\mathcal{H}$ is probabilistic, we can follow maximum likelihood estimation and define the Bayes-optimal classifier as the classifier that optimizes the conditional log likelihood (CLL) function:

$$\hat{\boldsymbol{h}} := \hat{\boldsymbol{p}} \in \operatorname{argmax}_{\boldsymbol{p} \in \mathcal{H}} \mathrm{CLL}(\boldsymbol{p} \,|\, \mathcal{D}) := \operatorname{argmax}_{\boldsymbol{p} \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \log \boldsymbol{p}(y_n \,|\, \boldsymbol{x}_n)\,. \tag{3}$$

The CLL is often augmented by a regularization term to avoid overfitting (Murphy, 2012; Nguyen et al., 2023a).

Once the classifier (3) is learned from $\mathcal{D}$, we can in principle find an optimal prediction of any loss function ℓ at the prediction time (Elkan, 2001; Mortier et al., 2021). More precisely, assume the classifier (3) is made available, and predicts for each query instance $\boldsymbol{x}$ a probability distribution $\boldsymbol{p}(\cdot \,|\, \boldsymbol{x})$ on the set of labelings $\mathcal{Y}$. The Bayes-optimal prediction (BOP) of any ℓ is then given by the expected loss minimizer

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}(\boldsymbol{x}) \in \operatorname*{argmin}_{\bar{y} \in \mathcal{Y}} \mathbf{E}\big(\ell(y, \bar{y})\big) = \operatorname*{argmin}_{\bar{y} \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} \ell(y, \bar{y}) \boldsymbol{p}(y \,|\, \boldsymbol{x}). \tag{4}$$

Yet, different losses may call for different BOPs. Knowledge about the conditional distribution $\boldsymbol{p}(\mathcal{Y} \,|\, \boldsymbol{x})$ is enough to find a BOP of any loss ℓ. Commonly used losses are the 0/1 loss

$$\ell_S(y, \bar{y}) = [\![y \neq \bar{y}]\!]\,, \tag{5}$$

where $[\![\cdot]\!]$ is the indicator function, i.e., $[\![A]\!] = 1$ if the predicate A is true and $= 0$ otherwise, and its cost-sensitive variants (Elkan, 2001). A BOP (4) of the ℓ_S (5) is simply a most probable class. Likewise, BOPs of cost-sensitive losses (Elkan, 2001) are often the top-ranked class where the rankings are constructed in the decreasing order of plausible scores, which are constructed based on the class probabilities.

In general, ensuring that the BOP (4) is an accurate prediction often requires reliable estimates of the class probabilities, which are hard to ensure when information is lacking. Threshold-based classifiers (Del Coz et al., 2009; Mortier et al., 2021) have been developed to mitigate the consequence of facing inaccurate estimates of class probabilities. These classifiers essentially return set-valued prediction consisting of the top (locally/globally) ranked classes. The Bayes-optimal set-valued prediction is the one which optimizes some loss function $\mathcal{L} : \mathcal{Y} \times 2^{\mathcal{Y}} \longrightarrow \mathbb{R}_+$ generalizing some conventional loss such as ℓ_S (5):

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}(\boldsymbol{x}) \in \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in 2^{\mathcal{Y}}} \mathbf{E}\big(\mathcal{L}(y, \bar{\boldsymbol{y}})\big) = \operatorname*{argmin}_{\bar{\boldsymbol{y}} \in 2^{\mathcal{Y}}} \sum_{y \in \mathcal{Y}} \mathcal{L}(y, \bar{\boldsymbol{y}}) \boldsymbol{p}(y \,|\, \boldsymbol{x}). \tag{6}$$

Commonly used generalized losses, including but not limited to (the loss version of) u_{65} and u_{80} (Zaffalon et al., 2012), and F-measure (Del Coz et al., 2009), are elaborated in Mortier et al. (2021). Their primary goal is to seek set-valued prediction with a correctness-precision trade-off. Compared to probabilistic classifiers which produce BOP of the ℓ_S (5), such threshold-based classifiers always gain in terms of correctness because the top-ranked class, hence the BOP for the ℓ_S, is always included in their set-valued prediction.

2.2 Classification with a set of probabilities

As just said, precise estimates of probabilities cannot always be expected to be reliable, and using threshold-based classifiers mitigates the possible bias but does not make this estimate more reliable. In contrast, using a set of probabilities increases this reliability. Under this setting, we assume that our uncertainty is described by a (not necessarily convex) set of probabilities $\mathcal{P}(\mathcal{Y} \,|\, \boldsymbol{x})$, i.e., a *credal set* (Levi, 1983). Clearly, the decision rule (4) and (6) are no longer well-defined. Therefore, it is necessary to use some generalized decision rules, some of them benefiting from strong theoretical justifications (Jansen et al., 2022; Troffaes, 2007).

Credal sets can arise in different ways, either as a native result of the learning method (Augustin et al., 2014), as the result of an agnostic (with respect to the missingness process) estimation in the presence of imprecise data, or as a neighborhood taken over an initial estimated distribution $\boldsymbol{p}(Y \,|\, \boldsymbol{x})$ (Montes et al., 2020; Rahimian & Mehrotra, 2019). These approaches are however not without issues: native credal classifiers can be computationally hard to learn and are unavailable for complex inputs such as mixed data and images. Approximating $\mathcal{P}(\mathcal{Y} \,|\, \boldsymbol{x})$ as a neighborhood taken over an initial estimated distribution $\boldsymbol{p}(Y \,|\, \boldsymbol{x})$ does not face this inconvenience but requires that the initial estimated distribution is well-estimated, hence circling back to the previously mentioned precise estimate reliability issue.

The quantile-based approach (Nguyen et al., 2023b) is an attempt to address the aforementioned challenges. It assumes that an ensemble (Dietterich, 2000; Sagi & Rokach, 2018) is made available. For each instance, it identifies a representative distribution from the output of the ensemble members and then distorts it using evidence from the closest distributions. In that approach, a representative distribution is the "median" of the set of distributions provided by the ensemble members under a specified statistical distance between distributions. When

the statistical distance is the squared Euclidean distance, the representative distribution is identical to the probabilistic prediction of the graded majority voting ensemble which averages the class probabilities in a class-wise manner. Sections 3 and 4 study in detail this approach, that we use as our backbone.

2.3 Set-valued prediction-making using credal sets

As previously said, when our uncertainty is described by a credal set $\mathcal{P}(\mathcal{Y} \mid \boldsymbol{x})$, instead of a single probability $\boldsymbol{p}(\mathcal{Y} \mid \boldsymbol{x})$, it is necessary to make predictions using some theoretically founded decision rule extending classical expectation (Jansen et al., 2022; Troffaes, 2007). For any $\boldsymbol{p} \in \mathcal{P}(\mathcal{Y} \mid \boldsymbol{x})$ and any loss function ℓ, we shall denote the BOP by

$$\hat{y}^{\boldsymbol{p}}_{\ell} \in \operatorname*{argmin}_{\bar{y} \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} \ell(y, \bar{y}) \boldsymbol{p}(y \mid \boldsymbol{x}). \tag{7}$$

Definition 1 (*maximality*) An optimal set-valued prediction under the Maximality rule is the set of the maximal, non-dominated elements of the partial order $\succ_{\ell, \mathcal{P}}$:

$$\bar{y} \succ_{\ell, \mathcal{P}} \bar{y}' \text{ if } \inf_{\boldsymbol{p} \in \mathcal{P}} \mathbf{E}_{\boldsymbol{p}}\big(\ell(y, \bar{y}') - \ell(y, \bar{y})\big) > 0. \tag{8}$$

In other words, we have

$$\hat{\mathbf{Y}}^{M}_{\ell, \mathcal{P}} = \{\bar{y} \in \mathcal{Y} \mid \nexists\, \bar{y}' \text{ s.t. } \bar{y}' \succ_{\ell, \mathcal{P}} \bar{y}\}. \tag{9}$$

Definition 2 (*E-admissibility*) An optimal set-valued prediction under the E-admissibility rule is

$$\hat{\mathbf{Y}}^{E}_{\ell, \mathcal{P}} = \{\bar{y} \in \mathcal{Y} \mid \exists \boldsymbol{p} \in \mathcal{P} \text{ s.t. } \bar{y} = \hat{y}^{\boldsymbol{p}}_{\ell}\}. \tag{10}$$

It is known that the set-valued prediction given by the E-admissibility rule is a subset of the one obtained by the Maximality rule (Troffaes, 2007).

3 Learn a representative classifier

We assume an ensemble $\mathbf{H} := \{\boldsymbol{h}^m \mid m \in [M] := \{1, \ldots, M\}\}$ of M probabilistic classifiers $\boldsymbol{h}^m \in \mathcal{H}$, $m \in [M]$ is made available and provides, for each instance $\boldsymbol{x}$, a set of M probabilistic predictions, denoted by

$$\mathbf{H}(\boldsymbol{x}) := \{\boldsymbol{h}^m(\boldsymbol{x}) \mid m \in [M]\} = \left\{\boldsymbol{p}^m := \left(p^m_1, p^m_2, \ldots, p^m_K\right) \mid m \in [M]\right\}. \tag{11}$$

We formulate the problem of learning a representative classifier $\boldsymbol{h}^*$ given the ensemble $\mathbf{H} \subset \mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ under a specified statistical distance d between distributions as a minimization problem

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{Y}^{\mathcal{X}}} \sum_{m=1}^{M} \mathbf{E}(d(\boldsymbol{h}, \boldsymbol{h}^m)) = \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{Y}^{\mathcal{X}}} \mathbf{E}\left(\sum_{m=1}^{M} d(\boldsymbol{h}, \boldsymbol{h}^m)\right) \tag{12}$$

$$= \underset{\boldsymbol{h}\in\mathcal{Y}^{\mathcal{X}}}{\operatorname{argmin}} \int_{\boldsymbol{x}\in\mathcal{X}} \left(\sum_{m=1}^{M} d(\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{h}^m(\boldsymbol{x})) \right) d\boldsymbol{x}\,, \tag{13}$$

which can be interpreted as finding a classifier $\boldsymbol{h}^* \in \mathcal{Y}^{\mathcal{X}}$ which minimizes the average expected distance to the members of $\mathbf{H}$.

The assumption $\boldsymbol{h}^* \in \mathcal{Y}^{\mathcal{X}}$ means that $\boldsymbol{h}^*$ is not restricted to any specific hypothesis space $\mathcal{H} \subsetneq \mathcal{Y}^{\mathcal{X}}$. Therefore, one minimizer $\boldsymbol{h}^*$ of (12) can be defined in a pointwise manner, i.e., for each $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{h}^*$ can be any classifier which produces

$$\boldsymbol{p}^* = \underset{\boldsymbol{p}:\sum_{k=1}^{K} p_k=1}{\operatorname{argmin}} \sum_{m=1}^{M} d(\boldsymbol{p}, \boldsymbol{p}^m). \tag{14}$$

As pointed out (Nguyen et al., 2023b) (and elsewhere), when d is the squared Euclidean distance, the representative distribution $\boldsymbol{p}^*$ (14) is the average of $\boldsymbol{p}^m$, $m \in [M]$, i.e., $\boldsymbol{h}^*$ is the graded majority voting ensemble. This also suggests that changing the distance d in (14) may lead to other graded majority voting ensembles.

Once the representative distribution $\boldsymbol{p}^*$ (14) is found, it can be used to produce either a singleton prediction (4) or a set-valued prediction (6) (Mortier et al., 2021).

The computational complexity of the problem of determining the representative distribution (14) can greatly depend on the nature of the distance d. In the next section, we discuss this computation for commonly used convex distances (Cha, 2007; Gibbs & Su, 2002; Lee, 1999; Sriperumbudur et al., 2010).

3.1 The case of convex distances

For completeness, we shall start with a few definitions and remarks, which are quite basic and can be found in textbooks/papers (see, e.g., Boyd et al. 2004; Datta 2010; Pugh 2015).

Definition 3 A function $f : \mathbb{R}^K \longmapsto \mathbb{R}$ is convex if for every $\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{R}^K$ and every $\lambda_1, \lambda_2 \in [0, 1]$ such that $\lambda_1 + \lambda_2 = 1$, we have the inequality

$$f(\lambda_1 \boldsymbol{p} + \lambda_2 \boldsymbol{p}') \leq \lambda_1 f(\boldsymbol{p}) + \lambda_2 f(\boldsymbol{p}')\,. \tag{15}$$

Remark 1 Let $z \in \mathbb{R}^K$. Let $\|\cdot\|$ be a norm on $\mathbb{R}^K$. $f(\boldsymbol{p}) := \|\boldsymbol{p} - z\|$ is convex.

Remark 2 Conical combinations of convex functions are also convex.

In the following, we show that if $f^m(\boldsymbol{p}) := d(\boldsymbol{p}, \boldsymbol{p}^m)$ is convex, $m \in [M]$, then the problem of finding a representative distribution (14) of $\mathbf{H}(\boldsymbol{x})$ can be straightforwardly formulated as a convex optimization problem. This is indeed computationally advantageous, as with recent advances convex programming is nearly as straightforward to solve as linear programming (Boyd et al., 2004; Rockafellar, 1993). Moreover, the set of optimal representative distributions inherits nice properties of the set of optimal solutions of convex optimization problems (Boyd et al., 2004; Rockafellar, 1993): every local minimum is a global minimum; the optimal set is convex; if the objective function is strictly convex, then the problem has at most one optimal point.

Definition 4 A standard convex optimization problem is of the form

$$\text{minimize}_{\boldsymbol{p}} \;\; f(\boldsymbol{p}) \qquad \text{subject to} \qquad g_i(\boldsymbol{p}) \leq 0\,, i \in [I]\,,\; h_j(\boldsymbol{p}) = 0\,, j \in [J] \tag{16}$$

where: $\boldsymbol{p} \in \mathbb{R}^K$ is the optimization variable; the objective function $f : \mathbb{R}^K \longmapsto \mathbb{R}$ is convex; the inequality constraint functions $g_i : \mathbb{R}^K \longmapsto \mathbb{R}, i \in [I]$ are convex; the equality constraint functions $h_j : \mathbb{R}^K \longmapsto \mathbb{R}$, $j \in [J]$, are of the form: $h_i(\boldsymbol{p}) = \boldsymbol{a}_j\boldsymbol{p} - b_j$, where $\boldsymbol{a}_j$ is a vector and b_j is a scalar.

We can encode the condition that the representative distribution must be a valid probability distribution by using K inequality constraint functions g_i and 1 equality constraint function h_1:

$$g_k(\boldsymbol{p}) := -p_k \leq 0\,, k \in [K]\,,\; h_1(\boldsymbol{p}) := \mathbf{1}_K\boldsymbol{p} - 1 = 0\,, \tag{17}$$

where $\mathbf{1}_K = (1, \ldots, 1)$. The constraints $p_k \leq 1$, $k \in [K]$, are implicitly enforced by the K constraints g_k (i.e., $p_k \geq 0$, $k \in [K]$) and h_1 (i.e., $\sum_{k=1}^K p_k = 1$, $k \in [K]$). Therefore, we can use any existing package to find $\boldsymbol{p}^*$ (14).

Moreover, using Remark 2, we can easily check that the weighted version of the problem of finding the representative distribution $\boldsymbol{p}^*$ (14):

$$\boldsymbol{p}^* = \underset{\boldsymbol{p}:\sum_{k=1}^K p_k = 1}{\operatorname{argmin}} \sum_{m=1}^M w_m d(\boldsymbol{p}, \boldsymbol{p}^m)\,, w_m \geq 0\,, m \in [M]\,, \tag{18}$$

where the weight w_m typically reflects how reliable $\boldsymbol{p}^m$ is, can also be formulated as a convex optimization problem. We, however, will not discuss the weighted version (18) in this paper because it would complicate the presentation significantly while most of the results that have been and shall be presented for the unweighted version (14) can be generalized for the weighted version (18) either straightforwardly or with little extra attention.

Using Remarks 1–2, we can verify that different distances (See (Cha, 2007; Gibbs & Su, 2002; Lee, 1999; Sriperumbudur et al., 2010) and elsewhere) are convex. Examples are members of the L_p Minkowski family (19) and Chebyshev distance (20):

$$f_p(\boldsymbol{p}) := L_p(\boldsymbol{p}, \boldsymbol{z}) := \sqrt[p]{\sum_{k=1}^K |p_k - z_k|^p}\,, p \geq 1\,, \tag{19}$$

$$f_{\text{cheb}}(\boldsymbol{p}) := L_\infty(\boldsymbol{p}, \boldsymbol{z}) := \max_{k \in [K]} |p_k - z_k|\,. \tag{20}$$

Moreover, a closer look at Definition 3 is enough to verify the convexity of some other distances (Cha, 2007; Gibbs & Su, 2002; Lee, 1999; Sriperumbudur et al., 2010). Examples are the Squared Euclidean distance (whose square function allows triangle inequality) and KL divergence, whose inequality (15) is verified using the log sum inequality (See, e.g., https://statproofbook.github.io/P/kl-conv with $\boldsymbol{z} = \boldsymbol{q}_1 = \boldsymbol{q}_2$):

$$f_{\text{sqe}}(\boldsymbol{p}) := d^{\text{sqe}}(\boldsymbol{p}, \boldsymbol{z}) := \sum_{k=1}^K (p_k - z_k)^2\,, \tag{21}$$

$$f_{\text{KL}}(\boldsymbol{p}) := d_{\text{KL}}(\boldsymbol{p}, \boldsymbol{z}) := \sum_{k=1}^K p_k \log\left(p_k / z_k\right)\,. \tag{22}$$

Carefully looking at the nature of distance may allow one to solve the problem (16) even more efficiently. For example, for any given K, closed-form solution for the f_{Sqe} (21) and Inner Product (49) can be derived (See Proposition 1 and 6 later on). These are also special cases where the additional constraints (i.e., $\sum_{k=1}^{K} p_k = 1$ and $p_k \geq 0$, $k \in [K]$) do not change the minimizer. However, it is not always the case. For example, these additional constraints can change the minimizer of f_1 (19) (See Proposition 2). Also, different distances may share the same minimizer. Examples of such distances are Topsør (23) and Jensen-Shannon (24):

$$f_{\text{Top}}(\boldsymbol{p}) := d_{\text{Top}}(\boldsymbol{p}, z) := d_{\text{Kdiv}}(\boldsymbol{p}, z) + \sum_{k=1}^{K} z_k \log\left(2p_k^m/(p_k + z_k)\right), \tag{23}$$

$$f_{\text{JS}}(\boldsymbol{p}) := d_{\text{JS}}(\boldsymbol{p}, z) := d_{\text{Top}}(\boldsymbol{p}, z)/2\,. \tag{24}$$

3.2 Algorithmic solutions

We will only mention the distances used in the experiments (Euclidean, L1, KL) and defer the cases of other distances to "Appendix 2".

Finding the $\boldsymbol{p}^*$ (14) under the Squared Euclidean distance can be done analytically.

Proposition 1 *The representative distribution* $\boldsymbol{p}^*$ (14) *under the Squared Euclidean distance* f_{sqe} (21) *is uniquely defined as*

$$p_k^* = \frac{1}{M} \sum_{m=1}^{M} p_k^m\,, k \in [K]\,. \tag{25}$$

Regarding the following distances, we do not know whether analytical solutions to the problem (14) exist or not. Until further results are made available, we will need some convex optimisation solver to find the distribution $\boldsymbol{p}^*$ (14) under these distances. Moreover, it is important to encode the probability axioms that the representative distribution $\boldsymbol{p}^*$ needs to obey as constraints when solving (14). Otherwise, the solver can give us incorrect solutions as indicated in the next two propositions, and illustrated in their proofs through examples (see "Appendix 2").

Proposition 2 *Except for* $K = 2$, *the representative distribution* $\boldsymbol{p}^*$ (14) *under* f_1 (19) *may not be the minimizer of the relaxed optimization problem*

$$\bar{\boldsymbol{p}} \in \operatorname*{argmin}_{\boldsymbol{p}} \sum_{m=1}^{M} L_1(\boldsymbol{p}, \boldsymbol{p}^m) = \operatorname*{argmin}_{\boldsymbol{p}} \sum_{k=1}^{K} \left(\sum_{m=1}^{M} |p_k - p_k^m| \right). \tag{26}$$

Proposition 3 *The representative distribution* $\boldsymbol{p}^*$ (14) *under Kullback–Leibler* f_{KL} (22) *may not be the minimizer of the relaxed optimization problem*

$$\bar{\boldsymbol{p}} \in \operatorname*{argmin}_{\boldsymbol{p}} \sum_{m=1}^{M} d_{\text{KL}}(\boldsymbol{p}, \boldsymbol{p}^m) = \operatorname*{argmin}_{\boldsymbol{p}} \sum_{m=1}^{M} \left(\sum_{k=1}^{K} p_k \log\left(p_k/p_k^m\right) \right). \tag{27}$$

Without encoding the aforesaid axiomatic constraints, we could have solutions $\bar{\boldsymbol{p}}$ that are not probabilities, i.e., $\sum_{k=1}^{K} \bar{p}_k \neq 1$.

4 Learning credal classifier

As discussed in Sect. 3, threshold-based classifiers (Del Coz et al., 2009; Mortier et al., 2021), which are defined using the representative distribution $\boldsymbol{p}^*$ (14) and Bayes-optimal prediction (6), can result in set-valued prediction, and could be called "credal" classifiers. This section presents other credal classifiers whose intermediate outputs are *credal sets* (Levi, 1983), in which imprecision is encoded in the uncertainty representation rather than the decision.

4.1 Credal sets approximation

One might wonder whether simply defining the credal set as the convex hull

$$\mathbf{CH}(\boldsymbol{x}) := \left\{ \boldsymbol{p} := \sum_{m=1}^{M} \gamma_m \boldsymbol{p}^m \mid \sum_{m=1}^{M} \gamma_m = 1 \right\} \tag{28}$$

is enough to have a reliable yet informative credal classifier. While it seems natural, we think it is not a very promising strategy. As mentioned in Nguyen et al. (2023b), as ensembles typically include noisy and extreme probabilistic estimations, the credal set estimated by (28) would lead to set-valued predictions under Maximality and E-admissibility rules including unreliable classes supported by those few extreme distributions. Moreover, as illustrated in Example 1 later on, the admissible set may be large even if the number of ensemble members is small/moderate.

Another possibility is to adopt the ϵ-contamination (Bock et al., 2014) to restrict the credal set as

$$\mathcal{P}_\epsilon(\mathcal{Y} \mid \boldsymbol{x}) := \{(1-\epsilon)\boldsymbol{p}^* + \epsilon \boldsymbol{p} \mid \boldsymbol{p} \in \mathbf{CH}(\boldsymbol{x})\}\,, \tag{29}$$

and choose the hyperparameter ϵ using either a nested cross-validation procedure or a validation set. In particular, the next proposition shows that one can only rely on members of the ensemble to compute it.

Proposition 4 *$\mathcal{P}_\epsilon(\mathcal{Y} \mid \boldsymbol{x})$ is the convex hull of $\{(1-\epsilon)\boldsymbol{p}^* + \epsilon \boldsymbol{p}^m \mid m \in [M]\}$.*

Yet, as detailed in "Appendix 4 and 5", its training and inference phases can be costly due to complex constraints. This set is also based on all ensemble members, therefore not getting rid of extreme distributions. Finally, how to robustify probabilistic uncertainty measures using $\mathcal{P}_\epsilon$ in a way similar to the one we propose in Sect. 5 remains unclear.

To eliminate the effect of outliers among elements of $\mathbf{H}(\boldsymbol{x})$ when forming the credal set, we adopt the quantile-based approach originally introduced in Nguyen et al. (2023b). Once the distribution $\boldsymbol{p}^*$ (14) is made available, it allows us to define a preference order, reflecting how common/weird each distribution in $\mathbf{H}(\boldsymbol{x})$ is:

$$\boldsymbol{p} \succ \boldsymbol{p}' \text{ if } d(\boldsymbol{p}^*, \boldsymbol{p}) < d(\boldsymbol{p}^*, \boldsymbol{p}')\,. \tag{30}$$

Such a preference order in turn allows us to "discard" a given percentage of outliers among elements of $\mathbf{H}(\boldsymbol{x})$.

Let $\alpha \in [0, 1]$ be some threshold. We define $\mathbf{H}_\alpha(\boldsymbol{x})$ as the set of $(1-\alpha) * 100\,\%$ of closest distributions in $\mathbf{H}(\boldsymbol{x})$ with respect to the preference order (30). We approximate the credal set $\boldsymbol{p}(\mathcal{Y} \,|\, \boldsymbol{x})$ of $\boldsymbol{x}$ by the convex hull of $\mathbf{H}_\alpha(\boldsymbol{x})$. Let $\mathbf{H}_\alpha(\boldsymbol{x}) := \{\boldsymbol{p}^m \,|\, m \in [M_\alpha]\}$. The convex hull is defined as

$$\mathbf{CH}_\alpha(\boldsymbol{x}) := \left\{ \boldsymbol{p} := \sum_{m=1}^{M_\alpha} \gamma_m \boldsymbol{p}^m \,|\, \gamma_m \geq 0, m \in [M_\alpha], \sum_{m=1}^{M_\alpha} \gamma_m = 1 \right\}. \tag{31}$$

The hyperparameter α controls the percentage of outliers that should be discarded and can be chosen using either a nested cross-validation procedure or a validation set. Figure 1 illustrates the notion of thresholded credal set $\mathbf{CH}_\alpha(\boldsymbol{x})$ for the case of a 3-dimensional output space $\mathcal{Y} = \{a, b, c\}$.

In the next section, we elaborate on the problem of set-valued prediction-making using the credal set (31). Discussions on the computational complexity of choosing α by adopting a nested cross-validation procedure are deferred to "Appendix 5".

4.2 Set-valued prediction-making using credal sets

In the following, we discuss the computational complexity of the inference problem when ℓ is the 0/1 loss (5).

Let us start with the Maximality rule. For any distribution $\boldsymbol{p} \in \mathbf{CH}_\alpha(\boldsymbol{x})$, we have

$$\mathbf{E}_{\boldsymbol{p}}\big(\ell(y, \bar{y}') - \ell(y, \bar{y})\big) = \boldsymbol{p}(\bar{y} \,|\, \boldsymbol{x}) - \boldsymbol{p}(\bar{y}' \,|\, \boldsymbol{x}). \tag{32}$$

Thus, the relation $\bar{y} \succ_{\ell, \mathcal{P}} \bar{y}'$ holds if the maximum of the linear program

$$\text{maximize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \boldsymbol{p}(\bar{y}' \,|\, \boldsymbol{x}) - \boldsymbol{p}(\bar{y} \,|\, \boldsymbol{x}) \tag{33}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M_\alpha} \gamma_m \boldsymbol{p}^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M_\alpha} \gamma_m = 1, \tag{34}$$

is negative. Note that if $f(\boldsymbol{p})$ has a maximum value on the feasible region, then it has this value on (at least) one of the extreme points, i.e., elements of $\mathbf{H}_\alpha(\boldsymbol{x})$ (Murty, 1983)

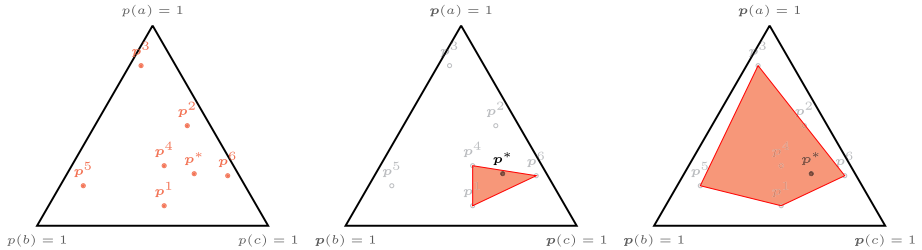


Fig. 1 Illustration of $\mathbf{CH}_\alpha(\boldsymbol{x})$ for $\mathcal{Y} = \{a, b, c\}$

[Theorem 3.3]. Thus, a naive algorithmic solution is to compute $f(\boldsymbol{p})$ for the extreme $\boldsymbol{p}$ and compare it with 0. This requires time $O(K^2 M_\alpha)$ because in the worst case, one needs to check all the $K(K-1)$ relation $\bar{y} \succ_{\ell,\mathcal{P}} \bar{y}'$, $\bar{y} \neq \bar{y}' \in \mathcal{Y}$.

We now tackle the E-admissibility rule. Reminding that, $\forall\ y \in \hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$, there must exist at least one $\boldsymbol{p} \in \mathbf{CH}_\alpha(\boldsymbol{x})$ such that $y = \hat{y}^{\boldsymbol{p}}_{\ell'}$. This is equivalent to having at least one $\boldsymbol{p} \in \mathbf{CH}_\alpha(\boldsymbol{x})$ such that $\boldsymbol{p}(y \,|\, \boldsymbol{x}) \geq \boldsymbol{p}(y' \,|\, \boldsymbol{x})$ for all $y' \neq y$. Thus, given any outer approximation $\mathcal{Y}^{\mathcal{O}}_{\ell,\mathcal{P}} \supseteq \hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ (e.g., the one given by maximality) we can follow the suggestion of Jansen et al. (2022) and formulate the problem of checking whether a given $y \in \mathcal{Y}^{\mathcal{O}}_{\ell,\mathcal{P}}$ satisfies the relation $y \in \hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ as checking whether a valid solution of a linear program in standard form exists.

$$\text{maximize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{35}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M_\alpha} \gamma_m \boldsymbol{p}^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M_\alpha} \gamma_m = 1\,, \tag{36}$$

$$\boldsymbol{p}(y \,|\, \boldsymbol{x}) - \boldsymbol{p}(y' \,|\, \boldsymbol{x}) \geq 0\,, y' \in \mathcal{Y} \setminus \{y\}\,. \tag{37}$$

Hence, finding $\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ requires solving $|\mathcal{Y}^{\mathcal{O}}_{\ell,\mathcal{P}}|$ linear programs, one per $y \in |\mathcal{Y}^{\mathcal{O}}_{\ell,\mathcal{P}}|$. The naive algorithmic solution, i.e., iterating over all the extreme points, can not be applied here because a class y may be optimal only for probabilities in the interior of $\mathbf{CH}_\alpha(\boldsymbol{x})$. This is illustrated in the next example.

Example 1 Let $\mathcal{Y} := \{y^1, y^2, y^3\}$ and $\mathbf{H}_\alpha(\boldsymbol{x}) := \{(0.6, 0.4, 0.0), (0.0, 0.4, 0.6)\}$. Then, the first extreme distribution and the second extreme distribution respectively support y^1 and y^3, while there is at least one interior distribution, which supports y^2, such as

$$(0.3, 0.4, 0.3) = 1/2 * (0.6, 0.4, 0.0) + 1/2 * (0.0, 0.4, 0.6)\,. \tag{38}$$

This fact has been known and discussed for a long time in the IP literature (Seidenfeld et al., 1995). In our case, this also raises the question of whether we should simply consider the elements of the ensemble as a probability set, or their convex hull. In the earlier case, one can simply compute the BOP for each element of the ensemble to get the corresponding E-admissible set. We will not explore this option in this paper and will consider the convex hull of the selected probabilities. Further details and consideration about computational aspects can be found in Nakharutai et al. (2019); Decadt et al. (2022). Note that our approach allows for efficient classification using these two rules, which may not be the case for other "credalised" classifiers (Antonucci & De Campos, 2011).

Finding set-valued predictions under these IP rules when they are coupled with cost sensitivity losses (Elkan, 2001; Lachiche & Flach, 2003; O'brien et al., 2008) is elaborated in "Appendix 4.1".

5 Decision-related uncertainty quantification

A commonly tackled problem in decision-related uncertainty quantification is to find a meaningful way to associate each instance with a real-valued uncertainty score, which can be used to compare and select the instances whose prediction is the most (or least)

uncertain. A notable example is uncertainty sampling (see (Nguyen et al., 2022) and references therein) where the unlabeled instances are ranked according the their uncertainty scores, such as confidence level, smallest margin, and entropy, and the most uncertain instances are labeled to enrich the current training data. Another example is classification with a reject option where one decides to reject a particular instance according to its uncertainty score (Chow, 1970; Condessa et al., 2017).

When a single (probabilistic) classifier is employed to make predictions, the probabilistic uncertainty measures are typically constructed based on its probabilistic predictions and essentially reflect how the classifier is uncertain about its predictions. In the ensemble learning setting, a common practice is to learn an ensemble from the training data and use its representative classifier (12) to make predictions as well as to define uncertainty measures. This may grossly oversimplify the available information provided by the ensemble members. We wish to construct uncertainty measures that take into account the consensus between trusted elements of $\mathbf{H}(\boldsymbol{x})$ and are well-normalized to [0, 1]. Commonly used probabilistic measures, such as smallest margin (SM), confidence level (CL), and entropy, do not fit this purpose.

To illustrate their inadequateness, let us take the case of SM

$$\mathrm{SM}(\mathbf{H}(\boldsymbol{x})) := \mathrm{SM}(\boldsymbol{p}^*) = p^*_{\mathrm{st}} - p^*_{\mathrm{nd}}\,, \tag{39}$$

where p^*_{st} and p^*_{nd} are respectively the most and the second most probable classes. While SM can take any value within [0, 1], its ability to reward the consensus of the ensemble members seems to be weak. The next example illustrates this behaviour.

Example 2 Let us extend the example 1 to the case of 100 members, where 50 members predict (0.6, 0.4, 0.0) and 50 other members predict (0.0, 0.4, 0.6) for the first instance, and all the 100 members predict (0.3, 0.4, 0.3) for the second instance. Yet, SM treats the two instances equally. It would be more reasonable to consider the first instance to be more uncertain as its uncertainty is due to both the uncertainty seen by the ensemble members and their consensus.

Clearly, CL and entropy, described in "Appendix 3", and any uncertainty measure, which is constructed merely based on the output of the representative classifier (12), can suffer from similar problems. Moreover, the range of CL and entropy are respectively $[1/K, 1]$ and $[0, \log_2(K)]$, and may be harder to normalize.

Let S be any probabilistic measure, whose range is a subset of [0, 1], and $\mathbf{H}(\boldsymbol{x})$ the ensemble predictions. To solve the mentioned issue, we propose to robustify and normalize S by using its robustified expectation

$$\mathrm{RS}(\mathbf{H}(\boldsymbol{x})) := \mathbf{E}(S(\boldsymbol{p}, \mathbf{H}(\boldsymbol{x}))|\boldsymbol{p} \in \Delta^{\mathrm{st}}_{\boldsymbol{x}}) \tag{40}$$

where Δ is the probability simplex, $\Delta^{\mathrm{st}}_{\boldsymbol{x}}$ is the admissible region of the most probable class on $\boldsymbol{p}^*$, that is the subset of probabilities for which the most probable class on $\boldsymbol{p}^*$ is the optimal prediction for $\boldsymbol{x}$, and $S(\boldsymbol{p}, \mathbf{H}(\boldsymbol{x}))$ is some uncertainty measure which simultaneously takes into account the uncertainty score $S(\boldsymbol{p})$ and the consensus among $\mathbf{H}(\boldsymbol{x})$. One intuitive notion of $S(\boldsymbol{p}, \mathbf{H}(\boldsymbol{x}))$ might be

$$S(\boldsymbol{p}, \mathbf{H}(\boldsymbol{x})) := [\![\boldsymbol{p} \in \mathbf{CH}(\boldsymbol{x})]\!] S(\boldsymbol{p})\,, \tag{41}$$

whose expectation (40) somehow reflects the consensus among $\mathbf{H}(\boldsymbol{x})$, as it would come down to condition (40) by $p \in \mathbf{CH}(\boldsymbol{x}) \cap \Delta^{\text{st}}_{\boldsymbol{x}}$. While it would be relatively simple to compute, this notion might be sensitive to noisy/extreme distributions, which would be hard to avoid when the ensemble $\mathbf{H}$ is constructed, e.g., from a random forest.

Therefore, we propose to further eliminate the effect of any $\boldsymbol{p} \in \Delta^{\text{st}}_{\boldsymbol{x}}$ which is "far" from $\boldsymbol{p}^*$, i.e., the potentially noisy/extreme distributions, by resorting to the quantile-based distortion approach (Nguyen et al., 2023b) and enlarging the convex hull (31) until $\mathbf{CH}_{\alpha}(\boldsymbol{x})$ no-longer supports a single class, that is until $\mathbf{CH}_{\alpha}(\boldsymbol{x}) \not\subseteq \Delta^{\text{st}}_{\boldsymbol{x}}$. Let α^* be the smallest value so that $\mathbf{CH}_{\alpha^*}(\boldsymbol{x})$ supports a single class. Figure 2 illustrates the notions of $\Delta^{\text{st}}_{\boldsymbol{x}}$ for 0/1 loss (5) and $\mathbf{CH}_{\alpha^*}$, using the same ensemble as in Fig. 1. We then define

$$S(\boldsymbol{p}, \mathbf{H}(\boldsymbol{x})) := [\![\boldsymbol{p} \in \mathbf{CH}_{\alpha^*}(\boldsymbol{x})]\!] S(\boldsymbol{p})\,. \tag{42}$$

The inclusion $\mathbf{CH}_{\alpha^*}(\boldsymbol{x}) \subset \Delta^{\text{st}}_{\boldsymbol{x}}$ simply implies

$$\text{RS}(\mathbf{H}(\boldsymbol{x})) = \int_{\boldsymbol{p} \in \Delta^{\text{st}}_{\boldsymbol{x}}} S(\boldsymbol{p}, \mathbf{H}(\boldsymbol{x})) d\boldsymbol{p} = \int_{\boldsymbol{p} \in \mathbf{CH}_{\alpha^*}(\boldsymbol{x})} S(\boldsymbol{p}) d\boldsymbol{p}\,. \tag{43}$$

In practice, we propose to use the empirical version of (43), i.e.,

$$\text{RS}(\mathbf{H}(\boldsymbol{x})) := \frac{1}{M+1} \left(\sum_{m=1}^{M} [\![\boldsymbol{p}^m \in \mathbf{CH}_{\alpha^*}(\boldsymbol{x})]\!] S(\boldsymbol{p}^m) + S(\boldsymbol{p}^*) \right). \tag{44}$$

Note that naively determining α^* by enlarging the convex hull (31) until $\mathbf{CH}_{\alpha}(\boldsymbol{x})$ no-longer supports a single class can be painfully expensive as it can require solving a possibly huge number of linear programs (35), especially in the context of uncertainty sampling. The following proposition ensures that the empirical expectation (44) can be computed in $O(M(K + \log(M))$ without having to explicitly determine $\mathbf{CH}_{\alpha^*}(\boldsymbol{x})$ by solving linear programs.

Proposition 5 *Assume the distance $d(\boldsymbol{p}, \boldsymbol{p}')$, for any $\boldsymbol{p}, \boldsymbol{p}' \in \Delta$, and the most probable class $\hat{y}$ on $\boldsymbol{p} \in \Delta$ can be computed in $O(K)$. Assume the admissible region $\Delta^{\text{st}}_{\boldsymbol{x}}$ is convex. Assume the representative distribution $\boldsymbol{p}^*$ (14) is already computed. The empirical expectation (44) can be computed in $O(M(K + \log(M))$ given $\mathbf{H}(\boldsymbol{x})$.*

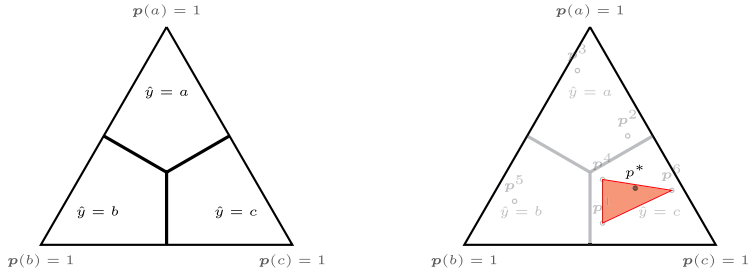


Fig. 2 Illustration of $\Delta^{\text{st}}_{\boldsymbol{x}}$ for 0/1 loss (5) and of $\mathbf{CH}_{\alpha^*}(\boldsymbol{x})$

The proof of Proposition 5 as well as a practical algorithm to compute the empirical expectation (44) are given in the "Appendix 5". We would like to emphasize that the assumptions on the complexity of computing the distance and the convexity of the admissible region $\Delta_{\boldsymbol{x}}^{\text{st}}$ are indeed weak and can be satisfied by all the distances and losses mentioned in this paper. Proposition 5 also indicates that this approach scales well both with respect to the number of classes (its complexity is linear with respect to this parameter) and with respect to the number of ensemble members (as it is mostly linear in this number).

6 Experiment

6.1 Experimental setting

We perform experiments on 12 tabular datasets from the UCI repository (see the first 4 columns of Table 1). Similar to what has been done in Nguyen et al. (2023b), we employ random forests (RFs) (Ho, 1995) (with the default setting of scikit-learn, except the minimum number of samples required to be at a leaf node is set to be 5) as the base learner in all the experiments. The source code has been made public at https://github.com/Haifei-ZHANG/Probability-Sets-Model.

Yet, the experiments on uncertainty sampling and classification with a reject option described in this section and the next section can be conducted with any distance and loss, which satisfy the conditions mentioned in Proposition 5. To keep the experimental part not too long, we will focus on the commonly used f_{sqe} (21) in the experiments.

6.1.1 Uncertainty sampling

To assess the robustified uncertainty scores (44), we use them to robustify the SM (39) and CL (58) in the uncertainty sampling setting. As discussed in Nguyen et al. (2022) (and references therein), the querying process typically ends if either the training data set reaches a desired size, a targeted performance level is achieved, or no informative samples are available anymore. While pre-defining a targeted performance level is difficult (even if a validation data set is given), the 2 other stopping criteria are more practical. The last criterion can be done by setting some predefined uncertainty threshold and stopping the querying process if the certainty score exceeds the threshold (Nguyen et al., 2022; Zhu et al., 2010). We can also use the robustified scores (44) to select unlabeled instances when increasing the budget. The SM (39) and CL (58) are used as baselines in the experiments.

We follow a 10×5 fold-cross validation and start the querying process with 3% of initial training data. To facilitate the running time, we use a batch size of 3 and 5 whenever the initial pool contains respectively < 500 instances and ≥ 500 instances. In the experiments where the uncertainty scores are used as the stopping rule, we test with $t \in \{0.05 + x * 0.05 \,|\, x = 0, 1, \ldots, 19\}$. Whenever the threshold is used as the stopping rule, we shall start considering whether to end the querying process after querying 1% of the pool. In the second set of experiments, we further challenge the classifier and the uncertainty scores by randomly flipping 25% of the labels of the instances in the initial training data set and the initial pool. This is specially designed to assess the usefulness of the uncertainty scores under settings where data are annotated by inexperienced experts or inaccurate automatic annotation tools.

6.1.2 Classification with a reject option

We follow a 20×5 fold-cross validation procedure where 20% of the data are used as training data and the rest is considered as a test data set. This setting is chosen to ensure that the training data set is not uninformative and at the same time provides room for the improvement given by the reject option.

We conduct two sets of experiments on clean data sets and noisy data sets, where 25% of the labels of the instances in the initial training data are randomly flipped. For each set of experiments, we make two tests. In the former, the robustified scores (44), and the SM (39) and CL (58) are employed to rank the test instances. Their performance is compared using the test accuracy when increasing the acceptance rate. The uncertainty scores are also reported. In the latter, we report the test accuracy and acceptance rate when the threshold is fixed and only instances, whose confidence scores (44), (39) and (58) are higher than the threshold, are predicted.

6.1.3 Set-valued prediction-making

We follow a 10-fold cross-validation procedure and conduct experiments under two scenarios: original data sets and noisy data sets, which are constructed by randomly flipping 25% of the labels of the training instances. We compare 4 cautious predictors constructed based on the discussions in Sect. 4 with cautious random forest (CRF) (Zhang et al., 2023) which is a competitive approach among those that seek set-valued predictions from a given ensemble of trees.

- NDC (Del Coz et al., 2009; Mortier et al., 2021): The distribution (14) under the squared Euclidean distance (21) is used to make set-value predictions (6) optimizing the $1 - u_{65}$ loss.
- SQE-Ead (Nguyen et al., 2023b): The representative distribution (14) under the squared Euclidean distance (21) is found and then distorted to construct the credal set (31). The optimal set-value prediction is then the E-admissible classes under 0/1 loss (as detailed in Sect. 4.2). For each train test split, we follow a 10-fold nested cross-validation procedure to choose α optimizing u_{65}. The RF is then retrained using the entire training dataset and the chosen α is used to construct $\mathbf{CH}_{\alpha}(\boldsymbol{x})$ during the inference phase.
- L1-Ead (KL-Ead): L1-Ead (KL-Ead) is similar to SQE-Ead, except the distribution (14) is defined under the f_1 (19) (f_{KL} (22)) distance.
- CRF (Zhang et al., 2023): This cautious classifier applies the interval dominance on probability intervals provided by each imprecise tree to contract a belief function about classes and selects the subset of classes that maximizes the lower expected utility associated with u_{65} score as the set-valued prediction.
- $\mathbf{CH}_0$: For each $\boldsymbol{x}$, the credal set is estimated by the convex hull $\mathbf{CH}(\boldsymbol{x})$ given by Equation (28).

We compare the predictors using the u_{65} score (Zaffalon et al., 2012) and the correctness (i.e., the percentage of times the true class is in $\hat{\mathbf{Y}}^{M}_{\ell,\mathcal{P}}$), given the prediction was imprecise, versus the accuracy of RF on those instances. This set of experiments is conducted to study a few questions: can predictors that take a credal set as input and produce set-valued predictions following IP rules, such as SQE-Ead, L1-Ead, and

KL-Ead bring any advantage, compared to NDC which directly optimizes u_{65} score given an estimate of class probabilities? Can the aggregate-then-distort strategies, such as SQE-Ead, L1-Ead, and KL-Ead bring any advantage, compared to the distort-then-aggregate strategies, such as CRF? Can cautious predictors which include the prediction of the ensemble itself, such as SQE-Ead, bring any advantage, compared to those that do not obey this requirement, such as L1-Ead and KL-Ead?

6.1.4 Classification with imbalanced data sets

Although classification with imbalanced data sets is not a primary goal of this study, such a situation is common in applications (Haixiang et al., 2017; Krawczyk, 2016; Sáez et al., 2016), and one may wonder how the current approaches behave in such cases. As a first investigation in this direction, we conduct a series of experiments on the highly imbalanced data sets mentioned in the previous sections with respective to the imbalance ratio (IR) (Krawczyk, 2016; Sáez et al., 2016). This ratio is defined as the frequency of the most minor class divided by the frequency of the most major class. More precisely, our experiments will include 5 data sets with imbalance ratios smaller than 0.2: glass (IR = 0.118), ecoli (IR = 0.014), dermatology (IR = 0.180), balance scale (IR = 0.170) and wine quality (IR = 0.015).

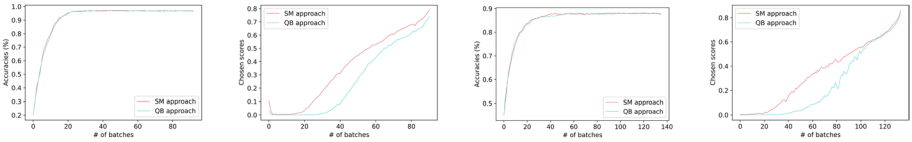
We follow a 10-fold cross-validation procedure on the clean data sets and conduct experiments under three scenarios (see https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html):

- RF is employed as the base learner.
- RF trained under class balanced weights (RF_BW) is employed as the base learner. The class-balanced weights are inversely proportional to the class frequencies in the training data set.
- RF trained under class-balanced subsample weights (RF_BSW) is employed as the base learner. For each tree, the class-balanced subsample weights are inversely proportional to the class frequencies in the corresponding bootstrap data set.

In each scenario, for each data set, we compared the accuracies given by the original RFs with the correctness, u_{65} and u_{80} given by the set-valued predictors (KL-Ead and SQE-Ead) on the classes separately. This set of experiments is conducted to study a few questions. Can our proposal on set-valued prediction-making bring any advantages in avoiding precisely wrong predictions on the minor classes? Can our proposal provide more significant impacts when coupled with ensembles, which explicitly consider the imbalances, such as RF_BW and RF_BSW? Our findings are given in Sect. 6.2.4

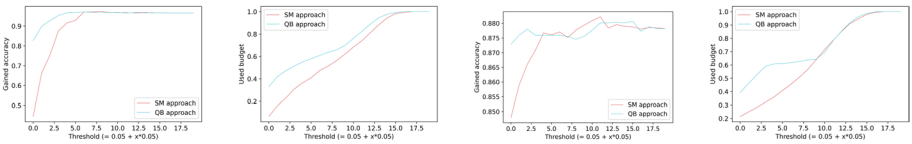
6.2 Results

Since the experimental results occupy several pages and follow consistent trends over data sets, we provide the full results to "Appendix 6", and only present the results for two data sets under the presence of noise (i.e., the more challenging setting) for uncertainty sampling and classification with a reject option.



(a) derma. + SM (b) derma. + SM (c) forest + SM (d) forest + SM

Fig. 3 Test accuracy and chosen score as the functions of the number of queries: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on noisy data sets



(a) derma. + SM (b) derma. + SM (e) forest + SM (f) forest + SM

Fig. 4 Test accuracy and used budget as the functions of the threshold: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on noisy data sets

6.2.1 Uncertainty sampling

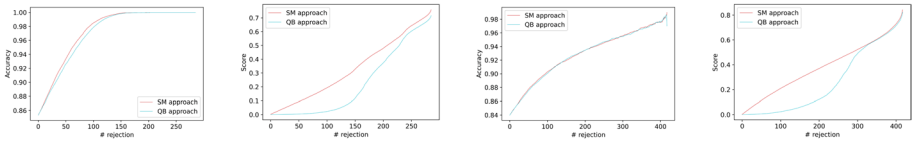
The results given in Fig. 3 indicate that the robustified uncertainty scores (44), marked as QB approach, and the SM (39) provide competitive test accuracy when increasing the budget, i.e., moving along the x-axis. An even more interesting remark is that the evolution of the uncertainty scores, in the case of robustified uncertainty scores (44), allows one to identify change points where accuracy will stabilize. More precisely, QB approach scores increase rate augment when asymptotic accuracy has been reached, which is not systematically the case for SM scores (this is particularly true for the forest data set in Fig. 3, where SM scores start to increase significantly after 20 batches, a state where the accuracy is not yet stabilized). This means that the robustified score, when used as a stopping sample criterion, tends to be more reliable, which was precisely the goal of this score.

Figure 4 provides another view, with the x-axis being a threshold cut rather than the number of queried points. One key interesting thing in this graph is that using the robustified scores (44) as a way to identify critical points to query, and the amount of such points is very effective. We can see that querying points with very low robust scores allow to already reach a quite high accuracy, close to the asymptotic one. Looking at "Appendix 6", one can see that this "initial" budget can vary between data sets, from less than 10 to more than 80%, and can be used as an assessment of the data set difficulty.

Similar results on other data sets as well as the cases of CL measure (58) are given in Fig. 9, 10, 11, 12, 13, 14, 15, 16 of "Appendix 6".

6.2.2 Classification with a reject option

The results given in Fig. 5 indicate that the robustified score (44) and the SM (39) provide competitive test accuracy when increasing the acceptance rate. More or less the same remarks can be done for the rejection case than for the active learning case. Again, the



(a) derma. + SM (b) derma. + SM (c) forest + SM (d) forest + SM

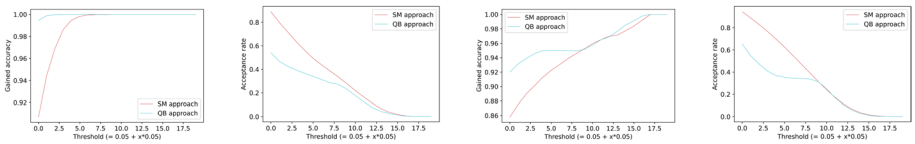
Fig. 5 Test accuracy and chosen score as the functions of the number of rejections: 20×5 cross-validation with (train, test) = (20%, 80%) on noisy data sets

evolution of the uncertainty scores provided by robustified uncertainty score (44) tends to better reflect the uncertainty level of the predicted set when the rejection process goes along, compared to the probabilistic score. When the uncertainty scores are used as the stopping rules, i.e., the rejection process ends as soon as the score exceeds the threshold, the robustified score (44) consistently provides better test accuracy, compared to the SM (39). Moreover, it tends to smartly trade-off between the test accuracy and the acceptance rate. Again, similar results on other data sets and the cases of CL (58) are given in Fig. 17, 18, 19, 20, 21, 22, 2324 of "Appendix 6" (Fig. 6).

6.2.3 Set-valued prediction-making

The u_{65} scores given in Table 1 suggest that aggregate-then-distort strategies, such as SQE-Ead, L1-Ead, and KL-Ead may be advantageous, compared to the distort-then-aggregate strategies, such as CRF. They also suggest that predictors that take a credal set as input and produce set-valued predictions following IP rules, such as SQE-Ead, L1-Ead, and KL-Ead may provide slightly better scores, compared to NDC which directly optimizes u_{65} score given an estimate of class probabilities, and the process by thresholding a precise estimate. In particular, KL-Ead consistently provides competitive and better results, compared to NDC, especially on the data sets with higher numbers of classes, i.e., ecoli, vowel, and libras with more than 7, and the data sets with noisy class labels. Besides providing promising empirical evidence supporting the potential advantages of our proposal, it also suggests that extending our proposal to cover other distances, which are specially designed to compare probability distributions, may further improve the predictive performance of the ensembles.

Another observation is that set-valued predictors which include the prediction of the ensemble itself, such as NDC and SQE-Ead, may provide worse scores, compared to those that do not obey this requirement, such as KL-Ead, especially on the data sets with larger



(a) derma. + SM (b) derma. + SM (c) forest + SM (d) forest + SM

Fig. 6 Test accuracy and acceptance rate as the functions of the threshold: 20×5 cross-validation with (train, test) = (20%, 80%) on noisy data sets

Table 1 U_{65} scores (in %) of different imprecise classifiers

Data set	N	P	K	NDC	SQE-Ead	L1-Ead	KL-Ead	CRF	$\mathbf{CH}_0$
The cases of training sets without label noise									
Wine (a)	178	13	3	96.35	97.57	97.38	**97.78**	95.10	58.12
Seeds (b)	210	7	3	91.07	91.45	91.26	**91.98**	90.33	75.28
Glass (c)	214	9	6	76.44	**76.56**	75.92	75.52	75.45	35.57
Ecoli (d)	336	7	8	85.51	86.07	85.81	**87.10**	84.46	43.60
Dermato. (e)	358	34	6	97.18	97.05	97.22	**98.34**	96.19	51.74
Libras (f)	360	90	15	76.58	73.35	75.24	**79.19**	73.45	14.60
Forest (g)	523	27	4	88.62	89.05	**89.09**	88.69	89.00	56.35
Balance (h)	625	4	3	85.94	**86.71**	86.70	85.32	85.46	61.45
Vehicle (i)	846	18	4	78.93	77.13	78.06	78.00	**79.05**	46.18
Vowel (j)	990	10	11	86.63	86.35	87.65	**92.12**	82.68	17.75
Wine qua. (k)	1599	11	6	**68.66**	68.32	68.39	68.30	67.35	36.53
Segment (l)	2300	19	7	97.17	97.12	96.99	**97.62**	96.73	71.00
The cases of training sets with 25% label noise									
Wine (a)	178	13	3	91.54	**95.15**	94.30	95.07	85.24	47.18
Seeds (b)	210	7	3	88.76	89.29	88.96	**89.46**	84.25	49.52
Glass (c)	214	9	6	**74.05**	72.16	73.40	73.01	70.96	26.02
Ecoli (d)	336	7	8	83.45	84.06	83.74	**84.71**	82.92	24.84
Dermato. (e)	358	34	6	95.82	96.68	**97.22**	96.91	95.52	32.72
Libras (f)	360	90	15	67.77	64.59	69.37	**72.86**	66.25	11.46
Forest (g)	523	27	4	87.76	87.70	87.61	**88.11**	85.97	36.94
Balance (h)	625	4	3	81.15	**83.64**	83.51	82.08	77.92	49.27
Vehicle (i)	846	18	4	**76.50**	75.64	76.41	75.05	74.50	36.55
Vowel (j)	990	10	11	81.44	79.81	83.27	**85.76**	79.45	14.53
Wine qua. (k)	1599	11	6	66.57	65.57	66.37	**66.87**	65.29	25.27
Segment (l)	2300	19	7	96.65	97.08	97.06	**97.13**	96.22	26.71

numbers of classes and noisy class labels. The consistently low scores provided by $\mathbf{CH}_0$ suggest that under such difficult data sets, the ensemble may provide more wrong predictions, and as a consequence, set-valued predictions that are forced to include them may need to be overly enlarged to cover the true class.

Ideally, a set-valued predictor should be more imprecise on difficult instances, on which the conventional precise prediction is likely to fail (Nguyen et al., 2018; Yang et al., 2014). To assess this ability of the set-valued predictors, we display in Fig. 7 (and Fig. 8 of "Appendix 6"), after fixing the set-valued predictor, for each data set, the correctness of the set-valued predictor, i.e., the percentage of times the true class is in the prediction of the set-valued predictor, given the prediction was imprecise, versus the accuracy of the RF on those instances.

In those graphs, an ideal point would be on the top left: this would mean that all set-valued predicted instances were wrongly predicted by RF (0 accuracy on the x-axis), but always covered the truth for the robust classifier (100% accuracy on the y-axis). They should also be put in perspective with Table 1, in particular their value for the x-axis. Let us consider for instance the segment data set (*l*) with clean labels. For SQE-Ead, we can see that for those instances having set-valued predictions, we have an accuracy around

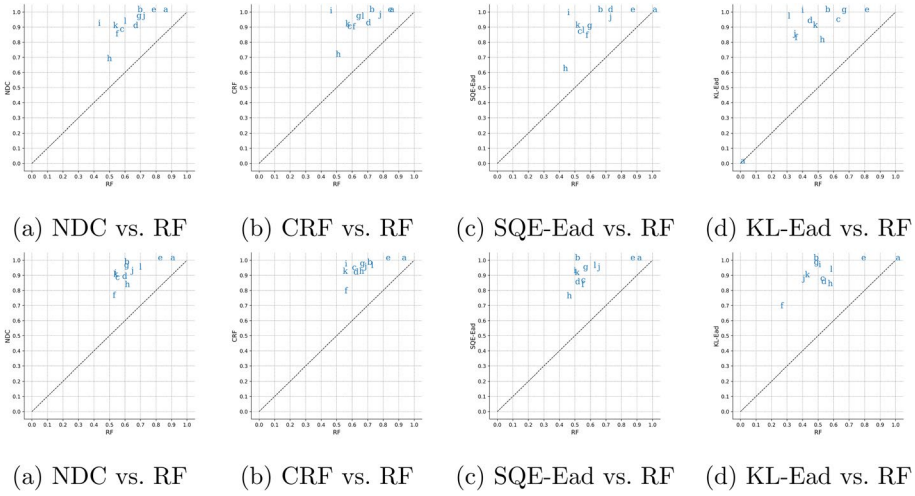


Fig. 7 Correctness of different imprecise classifiers in the case of abstention versus accuracy of the random forest on training sets without noise (first row) and with 25% noise (second row)

0.9, while the random forest (RF) predictions for the same instances have an accuracy of around 0.55, which is far below the average accuracies displayed in Table 1. For KL-Ead, this is even more pronounced: on those instances having set-valued predictions, KL-Ead accuracy is around 0.95, while RF predictions accuracy drops around 0.31. This indicates that for this data set, KL-Ead set-valued predictions tend to be more reliable, and focus on instances that are much harder to classify for the classical RF.

The results given in Figs. 7, 8 of "Appendix 6" suggest a few general interesting points. First, we can see that the correctness of the set-valued predictors is higher than the accuracy of the RF on both the clean data sets and noisy data sets. Moreover, the correctness tends to increase while the accuracy of RF tends to decrease when noisy classes are injected into training data. This suggests that all the set-valued predictors seem to do their job. On the clean data sets, there seem to be no clear trends in the difference in the correctness provided by the set-valued predictors. However, on the noisy data sets, the results would suggest that NDC, SQE-Ead, and L1-Ead do a better job, compared to KL-Ead and CRF. Moreover, L1-Ead seems to provide the most promising correctness which is usually higher than 80%, while the accuracy of the RF on the corresponding instances is at most 60%.

6.2.4 Classification with imbalanced data sets

In this subsection, we discuss the potential use of our methods to solve the common issue that is data set imbalance, using the experimental protocol of Sect. 6.1.4. In particular, we give here some results on two data sets that present some imbalance between classes, and complementary results on other data sets are given in Tables 5, 6, 7 of "Appendix 6.2". More precisely, we provide class-wise results for the balance data set (whose class B is imbalanced) and the Ecoli data set, which presents both very severe imbalance (two classes with only two samples) and more standard imbalance (classes omL and om).

The results given in Tables 2 and 3 indicate that set-valued prediction-making following either SED-Ead or KL-Ead often brings advantages in avoiding precisely wrong predictions on the minor classes. This is reflected by the promising correctness scores (compared to the RF accuracies). Moreover, as expected, SED-Ead, whose predictions always cover the predictions of RFs as part of its predictions consistently provides promising correctness, u_{65} and u_{65} scores, compared to the RF accuracies. KL-Ead, on the other hand, seems to be better to optimize average scores and can be worse in some minor classes occasionally.

However, the improvement when using standard RF remains limited, and sometimes non-existent (e.g., the class B in the case of SED-Ead). The empirical evidence suggests that our proposal may provide more significant impacts when coupled with ensembles of classifiers specifically designed to tackle classification with imbalanced data sets, such as RF_BW and RF_BSW. First, the empirical evidence suggests that ensembles that explicitly consider the imbalance, such as RF_BW and RF_BSW, clearly improve the predictive performance of the minor classes even in the precise case. Second, our proposal on set-valued prediction-making usually strengthens their ability to recognize these cases by either making precisely correct predictions or necessarily set-valued predictions. This is reflected by promising correctness, u_{65} and u_{65} scores, compared to the RF accuracies. Again, SED-Ead seems to be more promising in this regard, compared to KL-Ead. This is particularly clear, for instance, in the case of the Balance data set and class B.

To see how our method would perform in imbalance problems, we only performed some tests using rather basic techniques to deal with imbalance, and the results we discussed

Table 2 Results on balance scale data set (in %)

RF as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
B	49	0	0	0	0	22.45	10.48	13.47
L	288	94.79	96.88	94.5	95.49	97.92	92.37	93.89
R	288	94.1	96.18	93.69	94.72	97.22	91.01	92.78
RF_BW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
B	49	22.45	77.55	45.41	55.1	28.57	13.33	17.14
L	288	84.72	90.97	83.48	86.46	95.83	91.35	92.71
R	288	84.38	89.93	83.03	85.63	95.49	91.5	92.78
RF_BSW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
B	49	22.45	77.55	45.44	55.51	42.86	20.37	26.12
L	288	82.64	90.28	83.14	85.9	96.18	91.7	93.06
R	288	83.33	89.93	83.71	86.11	95.14	91.27	92.5

Table 3 Results on ecoli data set (in %)

RF as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
imL	2	0	0	0	0	0	0	0
imS	2	0	0	0	0	0	0	0
omL	5	0	60	29.58	37.5	60	22.5	29.34
om	20	85	90	75.83	80	90	83.31	85.38
imU	35	62.86	65.71	60.89	62.5	60	56.48	57.71
pp	52	84.62	84.62	83.27	83.85	90.38	87.46	88.22
im	77	88.31	88.31	86.04	87.01	89.61	89.16	89.35
cp	143	98.6	99.3	99.06	99.16	98.6	98.6	98.6
RF_BW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
imL	2	0	0	0	0	0	0	0
imS	2	0	0	0	0	0	0	0
omL	5	60	100	51.17	60.01	60	22.33	28.46
om	20	85	90	78.02	80.3	85	75.75	77.33
imU	35	77.14	82.86	75.04	77.93	62.86	56.85	58.83
pp	52	86.54	92.31	83.81	86.04	90.38	87.04	87.71
im	77	77.92	79.22	75.34	76.61	85.71	83.98	84.51
cp	143	97.9	98.6	95.77	96.64	98.6	97.87	98.18
RF_BSW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
imL	2	0	0	0	0	0	0	0
imS	2	0	0	0	0	0	0	0
omL	5	60	100	49.75	58.84	60	20.42	26.84
om	20	85	85	78.31	80.38	90	82.06	83.96
imU	35	77.14	85.71	77.89	80.79	60	56	57.71
pp	52	84.62	92.31	86.46	87.98	90.38	88.37	89.23
im	77	76.62	80.52	75.22	77.06	88.31	86.57	87.11
cp	143	97.9	99.3	96	97.07	98.6	98.11	98.32

should be considered encouraging clues rather than a firm conclusion. They however indicate that our approach is not sufficient by itself to deal with imbalance in a completely satisfactory way, and should be coupled with appropriate processing methods. As worthy follow-up work, one might think of considering other sophisticated classifiers for imbalanced classification, such as minimax classifiers (Gilet et al., 2024), as ensemble members in an extensive study.

7 Conclusion

We have elaborated on a model ensembling framework designed for multiple purposes, including set-valued prediction-making and decision-related uncertainty quantification, with applications in uncertainty sampling and classification with a reject option. To facilitate the scalability of the proposed framework, for all the problems and applications covered, we elaborate on their computational complexities from the theoretical aspects and leverage theoretical results to derive efficient algorithmic solutions. We think our effort is worthy, especially in the case of decision-related uncertainty quantification, where we are able to derive effective IP-based approaches whose computational complexities do not significantly exceed the ones of probabilistic approaches.

For set-valued prediction-making, empirical evidence suggests that our proposal may help to balance the trade-off between precision and recall, which is reflected via the u_{65} score. Moreover, the high correctness suggests that our proposal is capable of accurately detecting and covering the true classes in difficult instances, in which the ensemble is less accurate. Our proposal on decision-related uncertainty quantification might be interesting given the trade-off between its ability to reflect the uncertainty level introduced to the active learner during the querying process and to the decision maker during the rejection process, and its computational complexity.

We hope our theoretical results and algorithmic solutions (given in both the main text and appendices) facilitate follow-up works on the problems of distance, loss, and distortion mechanism selections for set-valued prediction-making. Moreover, we believe that our proposal on decision-related uncertainty quantification benefits and encourages follow-up works on robustifying other probabilistic measures and aggregating other types of ensembles, such as Bayesian neural networks (Jospin et al., 2022) and Monte Carlo dropout predictions (Lemay et al., 2022). Another worthy direction can be to explore the potential use of the robustified scores in online batch selection for faster training of neural networks (Mindermann et al., 2022).

Appendix 1: Notation and acronyms

Table 4 lists some frequently used notations and acronyms.

Appendix 2: Proofs of lemmas, propositions and remarks

We will first provide proofs for the remarks and propositions stated in Sect. 3.1 and then elaborate on the problem of finding the reference distribution (14) under other convex distances found in literature (Cha, 2007; Gibbs & Su, 2002; Lee, 1999; Sriperumbudur et al., 2010).

Table 4 Notation and acronyms

Symbol/acronym	Meaning
$\mathcal{X}, \boldsymbol{x}$	Instance space, instance
$\mathcal{Y} = \{y^1, \ldots, y^K\}, y$	Output space, outcome
X^p, Y^k	Feature, class variable
$\mathcal{D}$	Training data
K, P	Number of class variables, number of features
$[n]$	Set $\{1, \ldots, n\}$ of natural numbers
$[\![\cdot]\!]$	Indicator function
$\|\cdot\|$	A norm on $\mathbb{R}^K$
$\succ$	Preference order (30)
$\succ_{\ell,\mathcal{P}}$	Partial order (8)
$\mathbf{H} := \{\boldsymbol{h}^m \mid m \in [M]\}$	Ensemble of M probabilistic classifiers
$\mathbf{H}(\boldsymbol{x}) := \{\boldsymbol{h}^m(\boldsymbol{x}) \mid m \in [M]\}$	Set of probabilistic predictions on $\boldsymbol{x}$
$\mathbf{CH}_\alpha(\boldsymbol{x})$	Convex hull (31) of $\mathbf{H}_\alpha(\boldsymbol{x}) := \{\boldsymbol{p}^m \mid m \in [M_\alpha]\}$
$\mathcal{P}_\epsilon(\mathcal{Y} \mid \boldsymbol{x})$	Convex hull (29) of $\{(1-\epsilon)\boldsymbol{p}^* + \epsilon\boldsymbol{p}^m \mid m \in [M]\}$
$\hat{\mathbf{Y}}^M_{\ell,\mathcal{P}}$	Set-valued prediction under the Maximality rule (9)
$\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$	Set-valued prediction under the E-admissibility rule (10)
$\boldsymbol{p} := \boldsymbol{p}(\mathcal{Y} \mid \boldsymbol{x})$	Conditional probability distribution
$p_k := \boldsymbol{p}(y^k \mid \boldsymbol{x})$	Probability of outcome y^k given $\boldsymbol{x}$
$\boldsymbol{p}^* := \boldsymbol{p}^*(y \mid \boldsymbol{x})$	The representative distribution (14)
$\boldsymbol{p}^m := \boldsymbol{h}^m(\boldsymbol{x})$	Probabilistic prediction given by $\boldsymbol{h}^m$
Δ	The probability simplex
$\Delta^{\text{st}}_{\boldsymbol{x}}$	The admissible region of the most probable class on $\boldsymbol{p}^*$
$\ell_S(\cdot,\cdot)$	0/1 loss (5)
$d_\cdot(\cdot,\cdot)$ (or $L_\cdot(\cdot,\cdot)$)	Distance between two distributions
$f_\cdot(\boldsymbol{p},\cdot)$	Distance $d_\cdot(\boldsymbol{p},\cdot)$ (or $L_\cdot(\boldsymbol{p},\cdot)$) as the function of $\boldsymbol{p}$
$L_p(\cdot,\cdot)$	Minkowski distance (19)
$L_1(\cdot,\cdot)$	Minkowski distance with $p = 1$
$L_\infty(\cdot,\cdot)$	Chebyshev distance (20)
$d_{\text{Sqe}}(\cdot,\cdot)$	Squared Euclidean (21)
$d_{\text{KL}}(\cdot,\cdot)$	KL divergence (22)
$d_{\text{Kdiv}}(\cdot,\cdot)$	K divergence (52)
$d_{\text{IP}}(\cdot,\cdot)$	Inner Product (49)
$d_{\text{Top}}(\cdot,\cdot)$	Topsør (23)
$d_{\text{JS}}(\cdot,\cdot)$	Jensen-Shannon (24)
$d_{\text{P}}(\cdot,\cdot)$	Pearson (55)
SM$(\cdot)$	Smallest margin (39)
CL$(\cdot)$	Confidence level (58)
EN$(\cdot)$	Entropy (59)
RS$(\cdot)$	Robustified version (40) of the uncertainty measure S
BOP	Bayes-optimal prediction
RF	Random forest (Ho, 1995)
NDC	Nondeterministic classifier (Del Coz et al., 2009; Mortier et al., 2021)
CRF	Cautious random forest (Zhang et al., 2023)
SQE-Ead, L1-Ead, KL-Ead	Credal classifiers constructed in Sect. 4 and named in Sect. 6.1.3

2.1 Section 3.1 (The cases of convex distances)

2.1.1. Proof of remark 1

The convexity of $f(\boldsymbol{p})$ follows consequently from the triangle inequality of norms:

$$\begin{aligned} f(\lambda_1\boldsymbol{p}+\lambda_2\boldsymbol{p}') &= \|\lambda_1\boldsymbol{p}+\lambda_2\boldsymbol{p}'-z\| = \|\lambda_1(\boldsymbol{p}-z)+\lambda_2(\boldsymbol{p}'-z)\| \\ &\le \|\lambda_1(\boldsymbol{p}-z)\|+\|\lambda_2(\boldsymbol{p}'-z)\| = \lambda_1\|\boldsymbol{p}-z\|+\lambda_2\|\boldsymbol{p}'-z\| \\ &= \lambda_1 f(\boldsymbol{p})+\lambda_2 f(\boldsymbol{p}')\,. \end{aligned}$$

2.1.2. Proof of remark 2

The proof is trivial. It is enough to multiply the inequalities, one per convex function, by non-negative scalars and sum them up.

2.1.3. Proof of Proposition 1 (Squared Euclidean (21))

The proof is trivial and is given for completeness. For any $k \in [K]$, the partial derivative of

$$f(\boldsymbol{p}) = \sum_{m=1}^{M} f^m_{\mathrm{Sqe}}(\boldsymbol{p}) = \sum_{k=1}^{K}\left(\sum_{m=1}^{M}(p_k - p_k^m)\right)^2 \tag{45}$$

with respect to the variable p_k is

$$\frac{\partial f}{\partial p_k}(\boldsymbol{p}) = 2\sum_{m=1}^{M}(p_k - p_k^m) = 2\left(Mp_k - \sum_{m=1}^{M} p_k^m\right). \tag{46}$$

Since $f_{\mathrm{sqe}}(\boldsymbol{p})$ (21) is strictly convex, its unique minimizer is attained when the partial derivatives are zeros, i.e., $\boldsymbol{p}^*$ is defined in (25) (See (Steinley, 2006) and references therein). $\boldsymbol{p}^*$ is a valid distribution because the set of possible distributions is a convex set.

2.1.4. Proof of Proposition 2 (L_1 (19))

Without enforcing the probability axioms (i.e., $\sum_{k=1}^{K} p_k = 1$ and $p_k \ge 0$, $k \in [K]$), a minimizer $\bar{\boldsymbol{p}}$ of the relaxed optimization problem (26) is

$$\bar{p}_k := \mathrm{median}(p_k^1, \ldots, p_k^M)\,, k \in [K]\,. \tag{47}$$

This closed-form solution has been mentioned in Steinley (2006) and references therein. This can be verified by showing that, for any $\boldsymbol{p} \neq \bar{\boldsymbol{p}}$, we have

$$f_1(p_k) := \sum_{m=1}^{M} |p_k - p_k^m| \ge \sum_{m=1}^{M} |\bar{p}_k - p_k^m| := f_1(p'_k)\,, k \in [K]\,, \tag{48}$$

which implies the relation $f_1(\boldsymbol{p}) \ge f_1(\bar{\boldsymbol{p}})$.

Let L_k be the number of p_k^m which is larger than $\bar{p}_k$. Let S_k be the number of p_k^m which is smaller than $\bar{p}_k$. By definition of "median", we have $L_k = S_k$.

- $p_k > \bar{p}_k$: We have the following relations

$$|p_k - p_k^m| = |\bar{p}_k - p_k^m| + |p_k - \bar{p}_k| \text{ if } p_k^m \le \bar{p}_k\,,$$
$$|p_k - p_k^m| \ge |\bar{p}_k - p_k^m| - |\bar{p}_k - p_k| \text{ if } p_k^m \ge \bar{p}_k\,.$$

Therefore, we have

$$\begin{aligned} f_1(p_k) &= \sum_{m=1}^{M} |p_k - p_k^m| \\ &\ge \sum_{m=1}^{M} |\bar{p}_k - p_k^m| + |p_k - \bar{p}_k| S_k - |p_k - \bar{p}_k| L_k \\ &= \sum_{m=1}^{M} |\bar{p}_k - p_k^m| + |p_k - \bar{p}_k| (S_k - L_k) \\ &= \sum_{m=1}^{M} |\bar{p}_k - p_k^m| = f_1(p_k')\,. \end{aligned}$$

- $p_k < \bar{p}_k$: We have the following relations

$$|p_k - p_k^m| = |\bar{p}_k - p_k^m| + |p_k - \bar{p}_k| \text{ if } p_k^m \ge \bar{p}_k\,,$$
$$|p_k - p_k^m| \ge |\bar{p}_k - p_k^m| - |\bar{p}_k - p_k| \text{ if } p_k^m \le \bar{p}_k\,.$$

Therefore, we have

$$\begin{aligned} f_1(p_k) &= \sum_{m=1}^{M} |p_k - p_k^m| \\ &\ge \sum_{m=1}^{M} |\bar{p}_k - p_k^m| + |p_k - \bar{p}_k| L_K - |p_k - \bar{p}_k| S_k \\ &= \sum_{m=1}^{M} |\bar{p}_k - p_k^m| + |p_k - \bar{p}_k| (L_k - S_k) \\ &= \sum_{m=1}^{M} |\bar{p}_k - p_k^m| = f_1(p_k')\,. \end{aligned}$$

For $K > 2$, $\bar{\boldsymbol{p}}$ may not satisfy the probability axioms (see next Table).

$K = 3$				$K > 3$					
$\boldsymbol{p}^1$	0.8	0.1	0.1	$\boldsymbol{p}^1$	0.4	0.2	0.4/(K-3)	⋯	0.4/(K-3)
$\boldsymbol{p}^2$	0.2	0.5	0.3	$\boldsymbol{p}^2$	0.2	0.7	0.1/(K-3)	⋯	0.1/(K-3)
$\boldsymbol{p}^3$	0.1	0.4	0.5	$\boldsymbol{p}^3$	0.1	0.6	0.3/(K-3)	⋯	0.3/(K-3)
$\bar{\boldsymbol{p}}$	0.2	0.4	0.3	$\bar{\boldsymbol{p}}$	0.2	0.6	0.3/(K-3)	⋯	0.3/(K-3)

When $K = 2$, the probability axioms of $\bar{\boldsymbol{p}}$ are ensured by the fact that the total rank of each distribution $\boldsymbol{p}^m$, $m \in [M]$, on the first and the second classes is always $M + 1$ (as the masses should sum up to 1). Thus, $\bar{\boldsymbol{p}}$ is either one element of $\mathbf{H}(\boldsymbol{x})$ or the average of two elements of $\mathbf{H}(\boldsymbol{x})$. Let us illustrate this property using an example where $M = 9$:

		$\boldsymbol{p}^1$	$\boldsymbol{p}^2$	$\boldsymbol{p}^3$	$\boldsymbol{p}^4$	$\boldsymbol{p}^5$	$\boldsymbol{p}^6$	$\boldsymbol{p}^7$	$\boldsymbol{p}^8$	$\boldsymbol{p}^9$
p_1	Value	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
	Rank	1	2	3	4	5	6	7	8	9
p_2	Value	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	Rank	9	8	7	6	5	4	3	2	1

In this example, the total rank is 10 and $\bar{\boldsymbol{p}}$ is $\boldsymbol{p}^5$.

2.1.5. Proof of Proposition 3 (Kullback–Leibler divergence (22))

For any $k \in [K]$, the partial derivative of

$$\begin{aligned} f(\boldsymbol{p}) &= \sum_{m=1}^{M} d_{\mathrm{KL}}(\boldsymbol{p}, \boldsymbol{p}^m) = \sum_{m=1}^{M} \sum_{k=1}^{K} p_k \log_a \left(p_k / p_k^m\right) \\ &= \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_k \log_a \left(p_k\right) - p_k \log_a \left(p_k^m\right)\right) \\ &= \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_k \frac{\ln \left(p_k\right)}{\ln a} - p_k \frac{\ln \left(p_k^m\right)}{\ln a}\right) \\ &= \frac{1}{\ln a} \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_k \ln \left(p_k\right) - p_k \ln \left(p_k^m\right)\right) \end{aligned}$$

with respect to the variable p_k is

$$\begin{aligned} \frac{\partial f}{\partial p_k}(\boldsymbol{p}) &= \frac{1}{\ln a} \sum_{m=1}^{M} \left(1 + \ln \left(p_k\right) - \ln \left(p_k^m\right)\right) \\ &= \frac{1}{\ln a} \left(M + M \ln \left(p_k\right) - \sum_{m=1}^{M} \ln \left(p_k^m\right)\right) \\ &= \frac{1}{\ln a} \left(M \ln \left(p_k\right) + M - \ln \left(\prod_{m=1}^{M} p_k^m\right)\right). \end{aligned}$$

Since $f(\boldsymbol{p})$ is convex, one of its minimizer is attained when the partial derivatives are zeros, i.e.,

$$\ln\left(p_k^*\right) = \frac{1}{M}\ln\left(\prod_{m=1}^{M} p_k^m\right) - 1 = \ln\left(\left(\prod_{m=1}^{M} p_k^m\right)^{\frac{1}{M}}\right) - 1$$
$$= \ln\left(\frac{1}{e}\left(\prod_{m=1}^{M} p_k^m\right)^{\frac{1}{M}}\right), \forall k \in [K],$$

which implies that

$$p_k^* = \frac{1}{e}\left(\prod_{m=1}^{M} p_k^m\right)^{\frac{1}{M}}, \forall k \in [K].$$

The possible minimizer(s) may not satisfy the probabilities axioms. This is because if one wishes to have $\sum_{k=1}^{K} p_k^* = 1$, one must ensure

$$\sum_{k=1}^{K} \frac{1}{e}\left(\prod_{m=1}^{M} p_k^m\right)^{\frac{1}{M}} = 1 \Leftrightarrow \sum_{k=1}^{K}\left(\prod_{m=1}^{M} p_k^m\right)^{\frac{1}{M}} = e.$$

One may find different examples of $\mathbf{H}(\boldsymbol{x})$ which result in $\sum_{k=1}^{K} p_k^* \neq 1$. An example is $\mathbf{H}(\boldsymbol{x}) = \{(0.1, 0.9), (0.9, 0.1)\}$, which gives us $\boldsymbol{p}^* := \frac{1}{e}(0.3, 0.3)$.

2.2. Other convex distances

In complement to the algorithmic solutions for finding the representative distribution (14) under the convex distances studied in Sect. 3.2, we discuss this problem for other convex distances found in literature (Cha, 2007; Gibbs & Su, 2002; Lee, 1999; Sriperumbudur et al., 2010).

2.2.1 Convex distances with analytical solutions

Using Remark 2, we can verify the convexity of the Inner Product f_{IP}

$$f_{\text{IP}}(\boldsymbol{p}) := d_{\text{IP}}(\boldsymbol{p}, \boldsymbol{z}) := \sum_{k=1}^{K} p_k z_k. \tag{49}$$

Proposition 6 *The representative distribution* $\boldsymbol{p}^*$ (14) *under* f_{IP} (49) *is* (*uniquely*) *defined as and* $p_k^* = 0$, *for any* $k \neq k^*$, *and* $p_{k^*}^* = 1$ *for*

$$k^* = \underset{k}{\text{argmin}} \sum_{m=1}^{M} p_k^m. \tag{50}$$

Proof Finding a representative distribution $\boldsymbol{p}^*$ (14) is translated into finding a probabilistic minimizer of

$$f_{\mathrm{IP}}(\boldsymbol{p}) = \sum_{m=1}^{M} d_{\mathrm{IP}}(\boldsymbol{p}, \boldsymbol{p}^m) = \sum_{m=1}^{M} \sum_{k=1}^{K} p_k p_k^m$$
$$= \sum_{k=1}^{K} \sum_{m=1}^{M} p_k p_k^m = \sum_{k=1}^{K} p_k \left(\sum_{m=1}^{M} p_k^m \right).$$

Therefore, to minimize $f(\boldsymbol{p})$, it is enough to maximize p_{k*} with

$$k^* = \underset{k}{\operatorname{argmin}} \sum_{m=1}^{M} p_k^m \,, \tag{51}$$

by setting $p_{k*}^* = 1$ and $p_k^* = 0$ for any $k \neq k^*$. □

As a side comment, we would say that d_{IP} might not be an interesting distance within our framework because its representative distribution $\boldsymbol{p}^*$ (14) seems too extreme.

2.2.2. Convex distances without known analytical solutions

Regarding the following distances, we do not know whether analytical solutions to the problem (14) exist or not. Until further results are made available, we will need some solver to find the distribution $\boldsymbol{p}^*$ (14) under these distances.

Lemma 1 *The K divergence* f_{Kdiv} (52) *is convex.*

$$f_{\mathrm{Kdiv}}(\boldsymbol{p}) := d_{\mathrm{Kdiv}}(\boldsymbol{p}, \boldsymbol{z}) := \sum_{k=1}^{K} p_k \log \left(2p_k / (p_k + z_k) \right) \tag{52}$$

Proof It is sufficient to prove that for any $\alpha \in [0, 1]$, and pair of distribution $(\boldsymbol{p}, \boldsymbol{p}')$ we have

$$\begin{aligned} f_{\mathrm{Kdiv}}(\alpha \boldsymbol{p} + (1-\alpha)\boldsymbol{p}') &= d_{\mathrm{Kdiv}}(\alpha \boldsymbol{p} + (1-\alpha)\boldsymbol{p}', \boldsymbol{z}) \\ &\leq \alpha\, d_{\mathrm{Kdiv}}(\boldsymbol{p}, \boldsymbol{z}) + (1-\alpha)\, d_{\mathrm{Kdiv}}(\boldsymbol{p}', \boldsymbol{z}) \\ &= \alpha f_{\mathrm{Kdiv}}(\boldsymbol{p}) + (1-\alpha) f_{\mathrm{Kdiv}}(\boldsymbol{p}') \,. \end{aligned}$$

Reminding that the log sum inequality states that

$$\left(\sum_{i=1}^{I} a_i \right) \log \frac{\sum_{i=1}^{I} a_i}{\sum_{i=1}^{I} b_i} \leq \sum_{i=1}^{I} a_i \log \frac{a_i}{b_i} \,, \tag{53}$$

where a_i, $i \in [I]$, and b_i, $i \in [I]$, are non-negative real numbers.

For any $k \in [K]$, let $I = 2$, $(a_1, a_2) = (\alpha\, p_1^k, (1-\alpha)\, p_2^k)$ and $(b_1, b_2) = (\alpha\, (p_1^k + z_k), (1-\alpha)\, (p_2^k + z_k))$. We have

$$
\begin{aligned}
&\left(\alpha p_1^k + (1-\alpha) p_2^k\right) \log \frac{\alpha p_1^k + (1-\alpha) p_2^k}{\alpha p_1^k + (1-\alpha) p_2^k + z^k} \\
&= \left(\alpha p_1^k + (1-\alpha) p_2^k\right) \log \frac{\alpha p_1^k + (1-\alpha) p_2^k}{\alpha (p_1^k + z^k) + (1-\alpha) (p_2^k + z^k)} \\
&\leq \alpha p_1^k \log \frac{\alpha p_1^k}{\alpha (p_1^k + z^k)} + (1-\alpha) p_2^k \log \frac{(1-\alpha) p_2^k}{(1-\alpha)(p_2^k + z^k)} \\
&\leq \alpha p_1^k \log \frac{p_1^k}{p_1^k + z^k} + (1-\alpha) p_2^k \log \frac{p_2^k}{p_2^k + z^k} .
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
d_{\mathrm{Kdiv}}(\alpha \boldsymbol{p} + (1-\alpha)\boldsymbol{p}', \boldsymbol{z}) &= \sum_{k=1}^{K} \left(\alpha p_1^k + (1-\alpha) p_2^k\right) \log \frac{\alpha p_1^k + (1-\alpha) p_2^k}{\alpha p_1^k + (1-\alpha) p_2^k + z^k} \\
&\leq \sum_{k=1}^{K} \left(\alpha p_1^k \log \frac{p_1^k}{p_1^k + z^k} + (1-\alpha) p_2^k \log \frac{p_2^k}{p_2^k + z^k} \right) \\
&= \alpha \sum_{k=1}^{K} p_1^k \log \frac{p_1^k}{p_1^k + z^k} + (1-\alpha) \sum_{k=1}^{K} p_2^k \log \frac{p_2^k}{p_2^k + z^k} \\
&= \alpha\, d_{\mathrm{Kdiv}}(\boldsymbol{p}, \boldsymbol{z}) + (1-\alpha)\, d_{\mathrm{Kdiv}}(\boldsymbol{p}', \boldsymbol{z}) .
\end{aligned}
$$

This completes the proof. □

Proposition 7 *The representative distribution* $\boldsymbol{p}^*$ (14) *under K divergence* d_{Kdiv} (52) *may not be the minimizer of the relaxed optimization problem*

$$
\bar{\boldsymbol{p}} \in \operatorname*{argmin}_{\boldsymbol{p}} \sum_{m=1}^{M} d_{\mathrm{Kdiv}}(\boldsymbol{p}, \boldsymbol{p}^m) = \operatorname*{argmin}_{\boldsymbol{p}} \sum_{m=1}^{M} \sum_{k=1}^{K} p_k \log \left(2p_k / (p_k + p_k^m)\right) . \tag{54}
$$

Proof For any $k \in [K]$, the partial derivative of

$$
\begin{aligned}
f_{\mathrm{Kdiv}}(\boldsymbol{p}) &= \sum_{m=1}^{M} d_{\mathrm{Kdiv}}(\boldsymbol{p}, \boldsymbol{p}^m) = \sum_{m=1}^{M} \sum_{k=1}^{K} p_k \log \left(2p_k / (p_k + p_k^m)\right) \\
&= \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_k \log_a \left(p_k\right) - p_k \log_a \left(p_k + p_k^m\right)\right) + K * M \log_a 2 \\
&= \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_k \frac{\ln \left(p_k\right)}{\ln a} - p_k \frac{\ln \left(p_k + p_k^m\right)}{\ln a} \right) + K * M \log_a 2 \\
&= \frac{1}{\ln a} \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_k \ln \left(p_k\right) - p_k \ln \left(p_k + p_k^m\right)\right) + K * M \log_a 2
\end{aligned}
$$

with respect to the variable p_k is

$$\begin{aligned}\frac{\partial f_{\mathrm{Kdiv}}}{\partial p_k}(\boldsymbol{p}) &= \frac{1}{\ln a}\sum_{m=1}^{M}\left(1+\ln\left(p_k\right)-\frac{p_k}{p_k+p_k^m}-\ln\left(p_k+p_k^m\right)\right)\\ &= \frac{1}{\ln a}\sum_{m=1}^{M}\left(\ln\left(\frac{p_k}{p_k+p_k^m}\right)+\frac{p_k^m}{p_k+p_k^m}\right)\\ &= \frac{1}{\ln a}\sum_{m=1}^{M}\left(\ln\left(1-\frac{p_k^m}{p_k+p_k^m}\right)+\frac{p_k^m}{p_k+p_k^m}\right).\end{aligned}$$

Since $f_{\mathrm{Kdiv}}(\boldsymbol{p})$ is convex, its minimizer is attained when the partial derivatives are zeros, i.e.,

$$\frac{p_k^m}{p_k^*+p_k^m}=0\,,\forall k\in[K]\,,\forall m\in[M]\,,$$

which can be attained at either $p_k^*=+\infty$ or $p_k^m=0$, $k\in[K]$, where its (non-positive and decreasing) partial derivatives are maximized. These properties of the partial derivatives can be verified by the fact that $\ln(1-x)+x$ is a decreasing function (with a non-positive derivative, which is zero if $x=0$). □

Remark 2 ensures the convexity of the Topsøe f_{Top} (23) and Pearson f_{P}

$$f_{\mathrm{P}}(\boldsymbol{p}):=d^{\mathrm{P}}(\boldsymbol{p},\boldsymbol{z}):=\sum_{k=1}^{K}\frac{(p_k-z_k)^2}{z_k}\,. \tag{55}$$

Proposition 8 *The representative distribution* $\boldsymbol{p}^*$ (14) *under Pearson* f_{P} (55) *may not be the minimizer of the relaxed optimization problem*

$$\bar{\boldsymbol{p}}\in\operatorname*{argmin}_{\boldsymbol{p}}\sum_{m=1}^{M}d_{\mathrm{P}}(\boldsymbol{p},\boldsymbol{p}^m)=\operatorname*{argmin}_{\boldsymbol{p}}\sum_{m=1}^{M}\left(\sum_{k=1}^{K}\frac{(p_k-p_k^m)^2}{p_k^m}\right). \tag{56}$$

which is well-defined when $p_k^m>0$, $\forall m\in[M]$, $\forall k\in[K]$.

Proof For any $k\in[K]$, the partial derivative of

$$f_{\mathrm{P}}(\boldsymbol{p})=\sum_{m=1}^{M}d_{\mathrm{P}}(\boldsymbol{p},\boldsymbol{p}^m)=\sum_{m=1}^{M}\sum_{k=1}^{K}\frac{(p_k-p_k^m)^2}{p_k^m}=\sum_{k=1}^{K}\left(\sum_{m=1}^{M}\frac{(p_k-p_k^m)^2}{p_k^m}\right)$$

with respect to the variable p_k is

$$\frac{\partial f_{\mathrm{P}}}{\partial p_k}(\boldsymbol{p})=2\sum_{m=1}^{M}\frac{p_k-p_k^m}{p_k^m}=2\sum_{m=1}^{M}\left(\frac{p_k}{p_k^m}-1\right)=2\left(p_k\sum_{m=1}^{M}\frac{1}{p_k^m}-M\right).$$

Since $f_{\mathrm{P}}(\boldsymbol{p})$ is convex, one of its minimizers is attained when the partial derivatives are zeros, i.e.,

$$p_k^* = \frac{M}{\sum_{m=1}^{M} \frac{1}{p_k^m}}, \forall k \in [K].$$

One may find different examples of $\mathbf{H}(\boldsymbol{x})$ which result in $\sum_{k=1}^{K} p_k^* \neq 1$. An example is $\mathbf{H}(\boldsymbol{x}) = \{(2/7, 5/7), (5/7, 2/7)\}$, which gives us $\boldsymbol{p}^* := (20/49, 20/49)$. □

Proposition 9 *The representative distribution* $\boldsymbol{p}^*$ (14) *under Topsøe* f_{Top} (23) *may not be the minimizer of the relaxed optimization problem*

$$\bar{\boldsymbol{p}} \in \operatorname*{argmin}_{\boldsymbol{p}} \sum_{m=1}^{M} d_{\text{Top}}(\boldsymbol{p}, \boldsymbol{p}^m). \tag{57}$$

Proof For any $k \in [K]$, the partial derivative of

$$\begin{aligned}
f_{\text{Top}}(\boldsymbol{p}) &= \sum_{m=1}^{M} \left(d_{\text{Kdiv}}(\boldsymbol{p}, \boldsymbol{p}^m) + \sum_{k=1}^{K} p_k^m \log\left(2p_k^m/(p_k + p_k^m)\right) \right) \\
&= \sum_{m=1}^{M} d_{\text{Kdiv}}(\boldsymbol{p}, \boldsymbol{p}^m) + \sum_{m=1}^{M} \left(\sum_{k=1}^{K} p_k^m \log\left(2p_k^m/(p_k + p_k^m)\right) \right) \\
&= \sum_{m=1}^{M} d_{\text{Kdiv}}(\boldsymbol{p}, \boldsymbol{p}^m) + \sum_{k=1}^{K} \left(\sum_{m=1}^{M} p_k^m \log\left(2p_k^m/(p_k + p_k^m)\right) \right) \\
&= \sum_{m=1}^{M} d_{\text{Kdiv}}(\boldsymbol{p}, \boldsymbol{p}^m) + \sum_{k=1}^{K} \left(\sum_{m=1}^{M} \left(p_k^m \frac{\ln\left(p_k^m\right)}{\ln a} - p_k^m \frac{\ln\left(p_k + p_k^m\right)}{\ln a} \right) \right) \\
&\quad + K * M \log_a 2
\end{aligned}$$

with respect to the variable p_k is

$$\begin{aligned}
\frac{\partial f_{\text{Top}}}{\partial p_k}(\boldsymbol{p}) &= \frac{1}{\ln a} \sum_{m=1}^{M} \left(1 + \ln\left(p_k\right) - \frac{p_k}{p_k + p_k^m} - \ln\left(p_k + p_k^m\right) - \frac{p_k^m}{p_k + p_k^m} \right) \\
&= \frac{1}{\ln a} \sum_{m=1}^{M} \left(1 + \ln\left(p_k\right) - \frac{p_k + p_k^m}{p_k + p_k^m} - \ln\left(p_k + p_k^m\right) \right) \\
&= \frac{1}{\ln a} \sum_{m=1}^{M} \left(1 + \ln\left(p_k\right) - 1 - \ln\left(p_k + p_k^m\right) \right) \\
&= \frac{1}{\ln a} \sum_{m=1}^{M} \left(\ln\left(p_k\right) - \ln\left(p_k + p_k^m\right) \right) = \frac{1}{\ln a} \sum_{m=1}^{M} \left(\ln\left(\frac{p_k}{p_k + p_k^m} \right) \right).
\end{aligned}$$

Since $f_{\text{Top}}(\boldsymbol{p})$ is convex, its minimizer is attained when the partial derivatives are zeros, i.e.,

$$\ln\left(\frac{p_k}{p_k + p_k^m} \right) = 0 \Leftrightarrow \frac{p_k}{p_k + p_k^m} = 1, \forall k \in [K], \forall m \in [M],$$

which can be attained at either $p_k^* = +\infty$ or $p_k^m = 0, k \in [K]$. □

Appendix 3: Decision-related uncertainty quantification

In complement to our discussion on robustifying and normalizing probabilistic uncertainty measures given in Sect. 5, we shall provide additional discussions on the case of CL (58) and entropy, which are respectively defined as

$$\mathrm{CL}(\mathbf{H}(\boldsymbol{x})) := \mathrm{CL}(\boldsymbol{p}^*) = p^*_{\mathrm{st}}\,, \tag{58}$$

$$\mathrm{EN}(\mathbf{H}(\boldsymbol{x})) := \mathrm{EN}(\boldsymbol{p}^*) = -\sum_{k=1}^{K} p^*_k \log_2(p^*_k)\,. \tag{59}$$

Clearly, CL cannot take any value that is smaller than $1/K$, where K is the number of classes and is typically changed when the querying process goes along. This fact may lead to difficulties in tracking and leveraging the trusted level of uncertainty when the querying process goes along, especially when *CL* is used as a stopping criterion. Our proposal on normalizing and robustifying probabilistic measures in Sect. 5 appears to be very effective in this case. As can be observed in a range of experiments with uncertainty sampling and classification with a reject option presented in "Appendix 6", its robustified score (44) tends to be better normalized to the range [0, 1].

Yet, one can use our proposal to robustify the entropy (59) and other probabilistic measures whose range are not subsets of [0, 1]. We think it is reasonable to defer an intensive study on such probabilistic measures to further work. So far, we only use the most visible information, i.e., the probabilistic measure itself and the outputs of ensemble members when defining the robustified scores (44). However, it is not clear to us how to normalize the entropy (59) to the range [0, 1] without taking into account the number of classes seen at the end of the querying process. Yet, existing works on uncertainty sampling typically assume that this number is known from the beginning of the querying process. It is not always the case. In practice, emerging classes may appear during the querying process as well as at the test time.

Appendix 4: Set-valued prediction-making using credal sets: Complexities and algorithmic solutions

4.1 Quantile-based approach under cost sensitivity losses

In the following, we show that the algorithmic solutions for finding optimal set-valued predictions under the Maximality rule (9) and E-admissibility rule (10) coupled with the 0/1 loss can be (easily) adapted for finding optimal set-valued predictions under these IP rules when they are coupled with cost sensitivity losses (Elkan, 2001; Lachiche & Flach, 2003; O'brien et al., 2008).

Different cost sensitivity losses (Elkan, 2001; Lachiche & Flach, 2003; O'brien et al., 2008) have been proposed to relax the assumption that $\ell(y, \bar{y}) = \ell(y', \bar{y}')$ for any $(y, \bar{y}) \neq (y', \bar{y}')$. Such a cost sensitivity loss is typically constructed based on a cost matrix $\mathbf{C}$ whose element $c(y, \bar{y})$ informs the cost of predicting $\bar{y}$ when the true class is y. The cost sensitivity loss is then

$$\ell(y, \bar{y}) = c(y, \bar{y})[\![y \neq \bar{y}]\!]\,, \tag{60}$$

Again, its BOP is given by the expected loss minimizer

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}(\boldsymbol{x}) \in \operatorname*{argmin}_{\bar{y}\in\mathcal{Y}} \mathbf{E}\big(c(y,\bar{y})[\![y \neq \bar{y}]\!]\big) \tag{61}$$

$$= \operatorname*{argmin}_{\bar{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} c(y,\bar{y})[\![y \neq \bar{y}]\!]\boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{62}$$

$$= \operatorname*{argmin}_{\bar{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} [\![y \neq \bar{y}]\!](c(y,\bar{y})\boldsymbol{p}(y \,|\, \boldsymbol{x})) \tag{63}$$

$$= \operatorname*{argmin}_{\bar{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}\setminus\{\bar{y}\}} c(y,\bar{y})\boldsymbol{p}(y \,|\, \boldsymbol{x})\,. \tag{64}$$

The relation $\bar{y} \succ_{\ell,\mathcal{P}} \bar{y}'$ can be translated into

$$\inf_{\boldsymbol{p}\in\mathcal{P}} \mathbf{E}\boldsymbol{p}\big(\ell(y,\bar{y}') - \ell(y,\bar{y})\big) > 0 \tag{65}$$

$$\Leftrightarrow \inf_{\boldsymbol{p}\in\mathcal{P}} \sum_{y\in\mathcal{Y}} \big(c(y,\bar{y}') - c(y,\bar{y})\big)\boldsymbol{p}(y \,|\, \boldsymbol{x}) > 0\,. \tag{66}$$

Therefore, this relation holds if the minimum of the linear program

$$\text{minimize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \sum_{y\in\mathcal{Y}} \big(c(y,\bar{y}') - c(y,\bar{y})\big)\boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{67}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M_\alpha} \gamma_m \boldsymbol{p}^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M_\alpha} \gamma_m = 1\,, \tag{68}$$

is positive. Again, a naive algorithmic solution for (67) is to compute $f(\boldsymbol{p})$ for the extreme distribution $\boldsymbol{p}^m$, $m \in [M_\alpha]$ and compare it with 0. This simply implies that finding $\hat{\mathbf{Y}}^M_{\ell,\mathcal{P}}$ under any cost sensitivity loss of the form (60) can also be done by solving a set of linear programs.

Analogously, the problem of checking whether a given $y \in \mathcal{Y}^O_{\ell,\mathcal{P}}$ satisfies the relation $y \in \hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ as checking whether a valid solution of a linear program in standard form exists.

$$\text{maximize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{69}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M_\alpha} \gamma_m \boldsymbol{p}^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M_\alpha} \gamma_m = 1\,, \tag{70}$$

$$\sum_{y\in\mathcal{Y}} \big(c(y,\bar{y}') - c(y,\bar{y})\big)\boldsymbol{p}(y \,|\, \boldsymbol{x}) \geq 0\,, y' \in \mathcal{Y} \setminus \{y\}\,. \tag{71}$$

Again, the naive algorithmic solution, i.e., iterating over all the extreme points, can not be applied. Moreover, one may expect that solving the linear program (69) can be harder than solving (35) due to more complicated constraints (71).

4.2 ϵ-contamination approach under 0/1 loss

We will first provide a proof for the proposition 4, which is needed to determine the extreme distribution for finding the set-valued predictions under the E-admissibility and Maximality rules presented in this and the next section. Finding the set-valued predictions under the E-admissibility and Maximality rules then can be done by modifying the linear constraints of (33)–(35) and (67)–(69) to accommodate the new sets of extreme distributions.

Proof (for the proposition 4) The proof is quite intuitive. For any fixed ϵ, we have

$$\mathcal{P}_\epsilon(\mathcal{Y} \,|\, \boldsymbol{x}) := \{(1-\epsilon)\boldsymbol{p}^* + \epsilon \boldsymbol{p} \,|\, \boldsymbol{p} \in \mathbf{CH}(\boldsymbol{x})\} \tag{72}$$

$$= \left\{ (1-\epsilon)\boldsymbol{p}^* + \epsilon \left(\sum_{m=1}^{M} \gamma_m \boldsymbol{p}^m \right) \,|\, \sum_{m=1}^{M} \gamma_m = 1 \right\} \tag{73}$$

$$= \left\{ (1-\epsilon) \left(\sum_{m=1}^{M} \gamma_m \boldsymbol{p}^* \right) + \epsilon \left(\sum_{m=1}^{M} \gamma_m \boldsymbol{p}^m \right) \,|\, \sum_{m=1}^{M} \gamma_m = 1 \right\} \tag{74}$$

$$= \left\{ \sum_{m=1}^{M} \gamma_m (1-\epsilon)\boldsymbol{p}^* + \sum_{m=1}^{M} \gamma_m \epsilon \boldsymbol{p}^m \,|\, \sum_{m=1}^{M} \gamma_m = 1 \right\} \tag{75}$$

$$= \left\{ \sum_{m=1}^{M} \gamma_m ((1-\epsilon)\boldsymbol{p}^* + \epsilon \boldsymbol{p}^m) \,|\, \sum_{m=1}^{M} \gamma_m = 1 \right\}. \tag{76}$$

In other words, $\mathcal{P}_\epsilon(\mathcal{Y} \,|\, \boldsymbol{x})$ is the convex hull of $\{(1-\epsilon)\boldsymbol{p}^* + \epsilon \boldsymbol{p}^m \,|\, m \in [M]\}$. □

Let us denote by $\boldsymbol{p}_\epsilon^m := (1-\epsilon)\boldsymbol{p}^* + \epsilon, m \in [M]$. For a fixed ϵ, the problem of checking the relation $\bar{y} \succ_{\ell,\mathcal{P}} \bar{y}'$ can be translated into checking if the minimum of the linear program

$$\text{maximize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \boldsymbol{p}(\bar{y}' \,|\, \boldsymbol{x}) - \boldsymbol{p}(\bar{y} \,|\, \boldsymbol{x}) \tag{77}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M} \gamma_m \boldsymbol{p}_\epsilon^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M} \gamma_m = 1\,, \tag{78}$$

is negative. Again, a naive algorithmic solution for (77) is to compute $f(\boldsymbol{p})$ for the extreme distribution $\boldsymbol{p}_\epsilon^m$, $m \in [M]$ and compare it with 0.

The problem of checking whether a given $y \in \mathcal{Y}^O_{\ell,\mathcal{P}}$ satisfies the relation $y \in \hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ as checking whether a valid solution of a linear program in standard form exists.

$$\text{maximize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{79}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M} \gamma_m \boldsymbol{p}_\epsilon^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M} \gamma_m = 1\,, \tag{80}$$

$$\boldsymbol{p}(y \,|\, \boldsymbol{x}) - \boldsymbol{p}(y' \,|\, \boldsymbol{x}) \geq 0 \,, y' \in \mathcal{Y} \setminus \{y\} \,. \tag{81}$$

Similar to the case of constructing the credal set using the quantile-based approach, the naive algorithmic solution, i.e., iterating over all the extreme points, can not be applied.

4.3 ϵ-contamination approach under cost sensitivity losses

Let ℓ be a cost sensitivity loss of the form (60). For a fixed ϵ, the problem of checking the relation $\bar{y} \succ_{\ell,\mathcal{P}} \bar{y}'$ can be translated into checking if the minimum of the linear program

$$\text{minimize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \sum_{y \in \mathcal{Y}} \big(c(y, \bar{y}') - c(y, \bar{y})\big)\, \boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{82}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M} \gamma_m \boldsymbol{p}_\epsilon^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M} \gamma_m = 1 \,, \tag{83}$$

is positive. A naive algorithmic solution for (82) is to compute $f(\boldsymbol{p})$ for the extreme distribution $\boldsymbol{p}_\epsilon^m$, $m \in [M]$ and compare it with 0.

The problem of checking whether a given $y \in \mathcal{Y}^{\mathcal{O}}_{\ell,\mathcal{P}}$ satisfies the relation $y \in \hat{\mathbf{Y}}^{E}_{\ell,\mathcal{P}}$ as checking whether a valid solution of a linear program in standard form exists.

$$\text{maximize}_{\boldsymbol{p}} \quad f(\boldsymbol{p}) := \boldsymbol{p}(y \,|\, \boldsymbol{x}) \tag{84}$$

$$\text{subject to} \quad \boldsymbol{p} - \sum_{m=1}^{M} \gamma_m \boldsymbol{p}_\epsilon^m = 0, \gamma_m \geq 0, \sum_{m=1}^{M} \gamma_m = 1 \,, \tag{85}$$

$$\sum_{y \in \mathcal{Y}} \big(c(y, \bar{y}') - c(y, \bar{y})\big)\, \boldsymbol{p}(y \,|\, \boldsymbol{x}) \geq 0 \,, y' \in \mathcal{Y} \setminus \{y\} \,. \tag{86}$$

The naive algorithmic solution, i.e., iterating over all the extreme points, can not be applied.

Appendix 5: Nested cross-validation and uncertainty quantification: complexities and algorithmic solutions

5.1 Nested cross-validation

In this section, we leverage the nested structure of $\mathbf{CH}_\alpha(\boldsymbol{x})$ to reduce the number of the linear programs, which are needed to be solved, when doing nested cross-validation as described in Sect. 6.1. We also examine the computational complexity in the case the Maximality rule is employed as an approximate of the E-admissibility rule.

What shall be described can be easily adapted to do nested cross-validation when $\mathcal{P}_\epsilon(\mathcal{Y} \,|\, \boldsymbol{x})$ (29) is employed as an approximation of the credal set. The main difference would be the computational complexities can (significantly) increase due to more complex linear constraints in the linear programs as well as larger numbers of extreme points.

5.1.1 The case of E-admissibility rule

In practice, we can use the following heuristics to (hopefully) reduce the number of linear programs (35) that need to be solved.

- H_1:$\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ should not include any $y \in \mathcal{Y} \backslash \hat{\mathbf{Y}}^M_{\ell,\mathcal{P}}$ and any y such that

$$\max_{\boldsymbol{p} \in \mathbf{H}_\alpha(\boldsymbol{x})} \boldsymbol{p}(y \mid \boldsymbol{x}) < \frac{1}{K}. \tag{87}$$

- H_2: $\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}}$ should include all the optimal classes on $\{\boldsymbol{p}^*\} \cup \mathbf{H}_\alpha(\boldsymbol{x})$ and any class y such that

$$\max_{\boldsymbol{p} \in \mathbf{H}_\alpha(\boldsymbol{x})} \boldsymbol{p}(y \mid \boldsymbol{x}) \geq \frac{1}{2}. \tag{88}$$

We now further reduce the number of linear programs that need to be solved by leveraging the nested structure of $\mathbf{CH}_\alpha(\boldsymbol{x})$. By definition we have $\mathbf{CH}_{\alpha_2}(\boldsymbol{x}) \subset \mathbf{CH}_{\alpha_1}(\boldsymbol{x})$ whenever $\alpha_2 \geq \alpha_1$ because we discard more extreme distributions when using α_2. Therefore, we have the nested structure

$$\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}_{\alpha_2}} \subset \hat{\mathbf{Y}}^E_{\ell,\mathcal{P}_{\alpha_1}}. \tag{89}$$

One might want to switch between the backward strategy, which only considers the members of $\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}_{\alpha_1}}$ when computing $\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}_{\alpha_2}}$, and forward strategy, which ignores the members of $\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}_{\alpha_2}}$ when computing $\hat{\mathbf{Y}}^E_{\ell,\mathcal{P}_{\alpha_1}}$, to further reduce the total number of the number of the linear programs, which are needed to be solved.

5.1.2 The case of Maximality rule

Choosing an optimal α following a nested cross-validation when the Maximality rule is employed as an approximate of the E-admissibility rule does not require solving any linear program and can be done relatively fast by relying on the nested structure

$$\hat{\mathbf{Y}}^M_{\ell,\mathcal{P}_{\alpha_2}} \subset \hat{\mathbf{Y}}^M_{\ell,\mathcal{P}_{\alpha_1}}, \forall \alpha_2 \geq \alpha_1. \tag{90}$$

In practice, one can first sort the extreme distribution $\boldsymbol{p}^m$, $m \in [M]$, using the preference order (30), and compute their most probable classes. One can then go backward starting from the smallest value of α and its optimal set-valued prediction under the Maximality rule, and only focus on potential classes. At each step, one can first form a set of the most probable classes on the covered extreme distributions and check whether the other classes that have not been discarded in the previous step should be included in the current set.

5.2 Decision-related uncertainty quantification

We will first provide a proof for the proposition 5 and then provide a practical algorithm for computing the empirical expectation (44).

Proof (for the proposition 5) Assume the distance $d(\boldsymbol{p},\boldsymbol{p}')$, for any $\boldsymbol{p},\boldsymbol{p}' \in \Delta$, and the most probable class $\hat{y}$ on any distribution $\boldsymbol{p} \in \Delta$ can be computed in $O(K)$. Assume the admissible region $\Delta_{\boldsymbol{x}}^{\text{st}}$ is convex. Assume the distribution $\boldsymbol{p}^*$ (14) is already computed.

Since $\mathbf{CH}_{\alpha}(\boldsymbol{x})$ (31) is convex by definition, we have $\mathbf{CH}_{\alpha}(\boldsymbol{x}) \subset \Delta_{\boldsymbol{x}}^{\text{st}}$ whenever the set of extreme distributions $\mathbf{H}_{\alpha}(\boldsymbol{x}) \subset \Delta_{\boldsymbol{x}}^{\text{st}}$. Moreover, $\mathbf{H}_{\alpha}(\boldsymbol{x}) \backslash \Delta_{\boldsymbol{x}}^{\text{st}} \neq \emptyset$ as soon as there is at least one member of $\mathbf{H}_{\alpha}(\boldsymbol{x})$ has another most probable class than the one which is most probable on all the member of $\Delta_{\boldsymbol{x}}^{\text{st}}$. Therefore, whatever the value of α^* is, $\mathbf{CH}_{\alpha^*}(\boldsymbol{x})$ should be the convex hull of the longest consecutive subsequence of $\boldsymbol{p}^* \succ \boldsymbol{p}^{(1)} \succ \boldsymbol{p}^{(2)} \succ \boldsymbol{p}^{(M)}$ covering $\boldsymbol{p}^*$, whose elements share the most probable class with $\boldsymbol{p}^*$, where $\boldsymbol{p}^* \succ \boldsymbol{p}^{(1)} \succ \boldsymbol{p}^{(2)} \succ \boldsymbol{p}^{(M)}$ be the permutation on $\mathbf{H}(\boldsymbol{x})$ formed by the preference order $\succ$ (30).

Once this consecutive subsequence is determined, the empirical expectation (44) can be computed in $O(M)$ given $S(\boldsymbol{p}^m)$ and $S(\boldsymbol{p}^*)$ since this subsequence can contain at most $M+1$ distributions. Given the set of distances $\{d(\boldsymbol{p}^*,\boldsymbol{p}^m) \,|\, m \in [M]\}$, which can be computed in $O(MK)$, the major computation effort should be distributed to construct the permutation $\boldsymbol{p}^* \succ \boldsymbol{p}^{(1)} \succ \boldsymbol{p}^{(2)} \succ \boldsymbol{p}^{(M)}$ which takes time $O(M \log(M))$.

Altogether, computing the empirical expectation (44) can be done in $O(M(K + \log(M))$ given $\mathbf{H}(\boldsymbol{x})$ without having to solve any linear program. □

A practical algorithm for computing the expectation (44) is given in Algorithm 1.

Algorithm 1 Compute the empirical expectation $\text{RS}(\mathbf{H}(\boldsymbol{x}))$ (44)

Input: $\mathbf{H}(\boldsymbol{x}) := \{\boldsymbol{p}^m \,|\, m \in [M]\}$, a distance d, the representative distribution $\boldsymbol{p}^*$ (14)
Compute the set of distances $\{d(\boldsymbol{p}^*,\boldsymbol{p}^m) \,|\, m \in [M]\}$
Construct the permutation $\boldsymbol{p}^* \succ \boldsymbol{p}^{(1)} \succ \boldsymbol{p}^{(2)} \succ \boldsymbol{p}^{(M)}$
Initialize $\text{RS}(\mathbf{H}(\boldsymbol{x})) \longleftarrow \frac{1}{M+1}\left(\sum_{m=1}^{M} S(\boldsymbol{p}^m) + S(\boldsymbol{p}^*)\right)$
Initialize $\text{RS}_{\text{temp}} \longleftarrow S(\boldsymbol{p}^*)$
for $m = 1$ **to** M **do**
  if $\boldsymbol{p}^{(m)}$ and $\boldsymbol{p}^*$ have different most probable classes **then**
    Update $\text{RS}(\mathbf{H}(\boldsymbol{x})) \longleftarrow \frac{1}{M+1}\text{RS}_{\text{temp}}$
    break
  end if
  Update $\text{RS}_{\text{temp}} \longleftarrow \text{RS}_{\text{temp}} + S(\boldsymbol{p}^{(m)})$
end for
Output: $\mathbf{RS}(\mathbf{H}(\boldsymbol{x}))$

We close this appendix with discussions on the convexity of the admissible region $\Delta_{\boldsymbol{x}}^{\mathrm{st}}$. It can be easily verified that this assumption is satisfied by any cost sensitivity loss[1] (60) (and of course the 0/1 loss (5)). Assume the most probable class on $\Delta_{\boldsymbol{x}}^{\mathrm{st}}$ is $\hat{y}$. For any $\boldsymbol{p}, \boldsymbol{p}' \in \Delta_{\boldsymbol{x}}^{\mathrm{st}}$ and any $\lambda_1, \lambda_2 \in [0,1]$ such that $\lambda_1 + \lambda_2 = 1$, we have

$$
\begin{aligned}
&\sum_{y \in \mathcal{Y} \setminus \{\hat{y}\}} c(y, \hat{y}) \, (\lambda_1 \boldsymbol{p} + \lambda_2 \boldsymbol{p}')(y \,|\, \boldsymbol{x}) \\
=& \lambda_1 \sum_{y \in \mathcal{Y} \setminus \{\hat{y}\}} c(y, \hat{y}) \boldsymbol{p}(y \,|\, \boldsymbol{x}) + \lambda_2 \sum_{y \in \mathcal{Y} \setminus \{\hat{y}\}} c(y, \hat{y}) \boldsymbol{p}'(y \,|\, \boldsymbol{x}) \\
\leq& \lambda_1 \sum_{y \in \mathcal{Y} \setminus \{y'\}} c(y, y') \boldsymbol{p}(y \,|\, \boldsymbol{x}) + \lambda_2 \sum_{y \in \mathcal{Y} \setminus \{y'\}} c(y, y') \boldsymbol{p}'(y \,|\, \boldsymbol{x}) \\
=& \sum_{y \in \mathcal{Y} \setminus \{y'\}} c(y, y') \, (\lambda_1 \boldsymbol{p} + \lambda_2 \boldsymbol{p}')(y \,|\, \boldsymbol{x}) \,,
\end{aligned}
$$

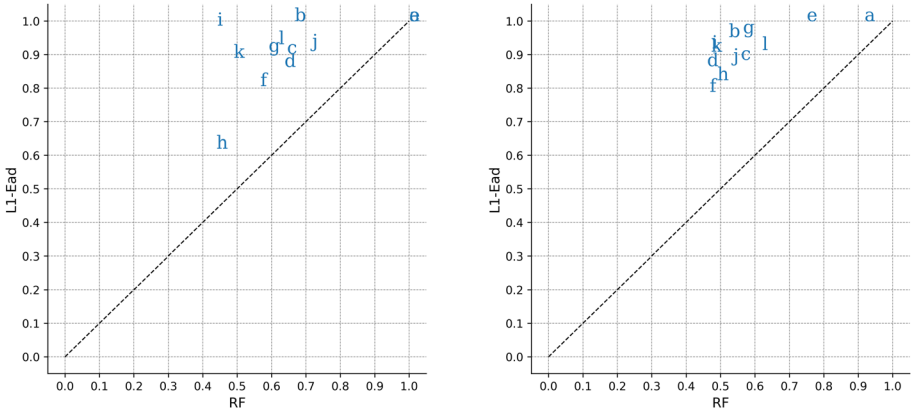
for any $y' \neq \hat{y}$. In other words, $\hat{y}$ is a most probable class on $\lambda_1 \boldsymbol{p} + \lambda_2 \boldsymbol{p}'$.

Appendix 6: Experimental results

6.1 Additional results

6.2 Additional results on imbalanced data sets

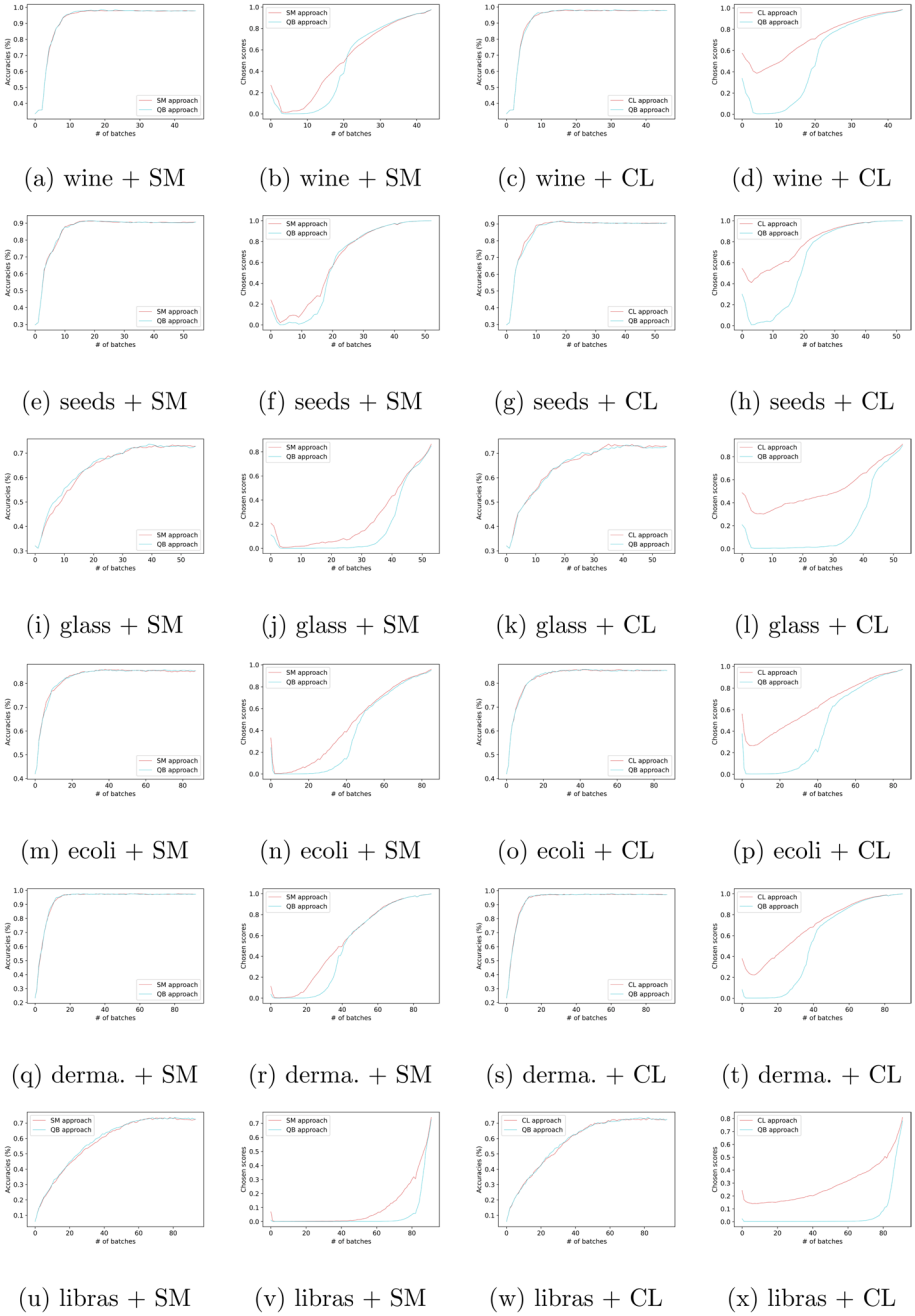
This appendix provides additional experimental results for the section 6.2.4 (Figs. 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, Tables 5, 6, 7).



(e.1) L1-Ead vs. RF (e.2) L1-Ead vs. RF

Fig. 8 Correctness of different imprecise classifiers in the case of abstention versus accuracy of the random forest on training sets without noise (left) and with 25% noise (right)

[1] This may be dissatisfied by cost sensitivity losses whose cost matrix depends on the concerned distribution $\boldsymbol{p}$, i.e., the cost matrix is defined locally.



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

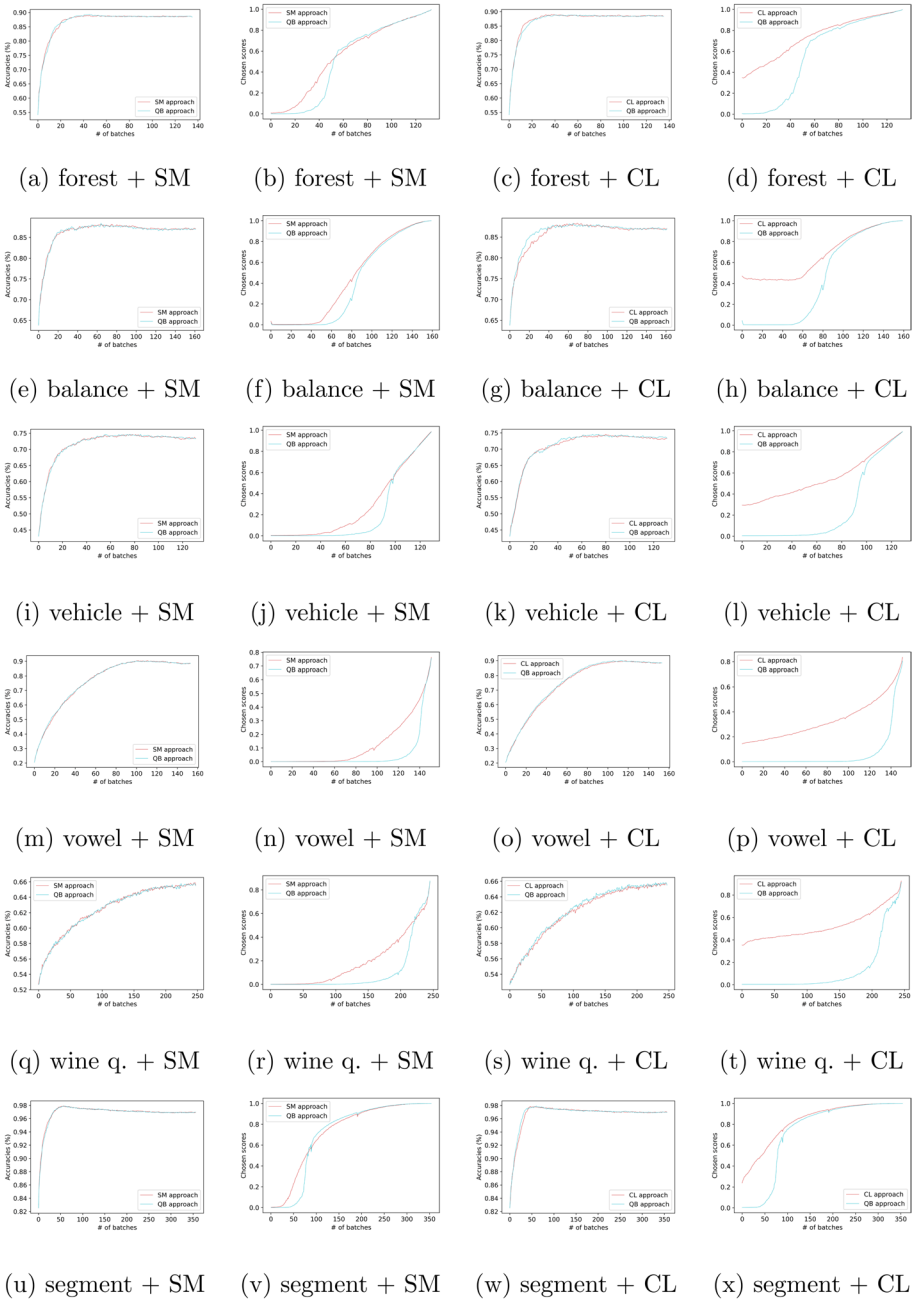
(i) glass + SM (j) glass + SM (k) glass + CL (l) glass + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. + SM (r) derma. + SM (s) derma. + CL (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 9 Test accuracy and chosen score as the functions of the number of queries: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on clean data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

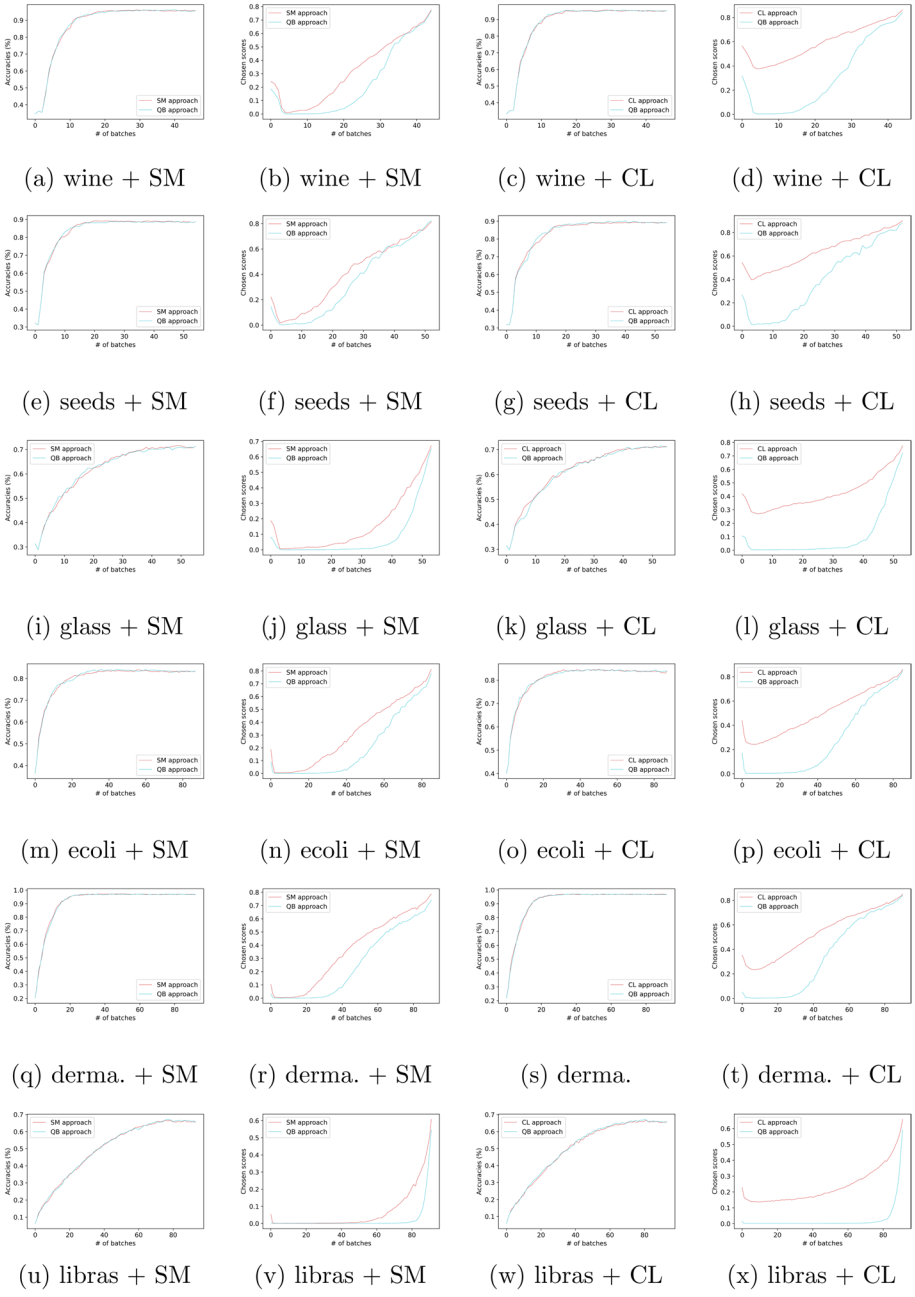
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(u) segment + SM (v) segment + SM (w) segment + CL (x) segment + CL

Fig. 10 Test accuracy and chosen score as the functions of the number of queries: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on clean data sets



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

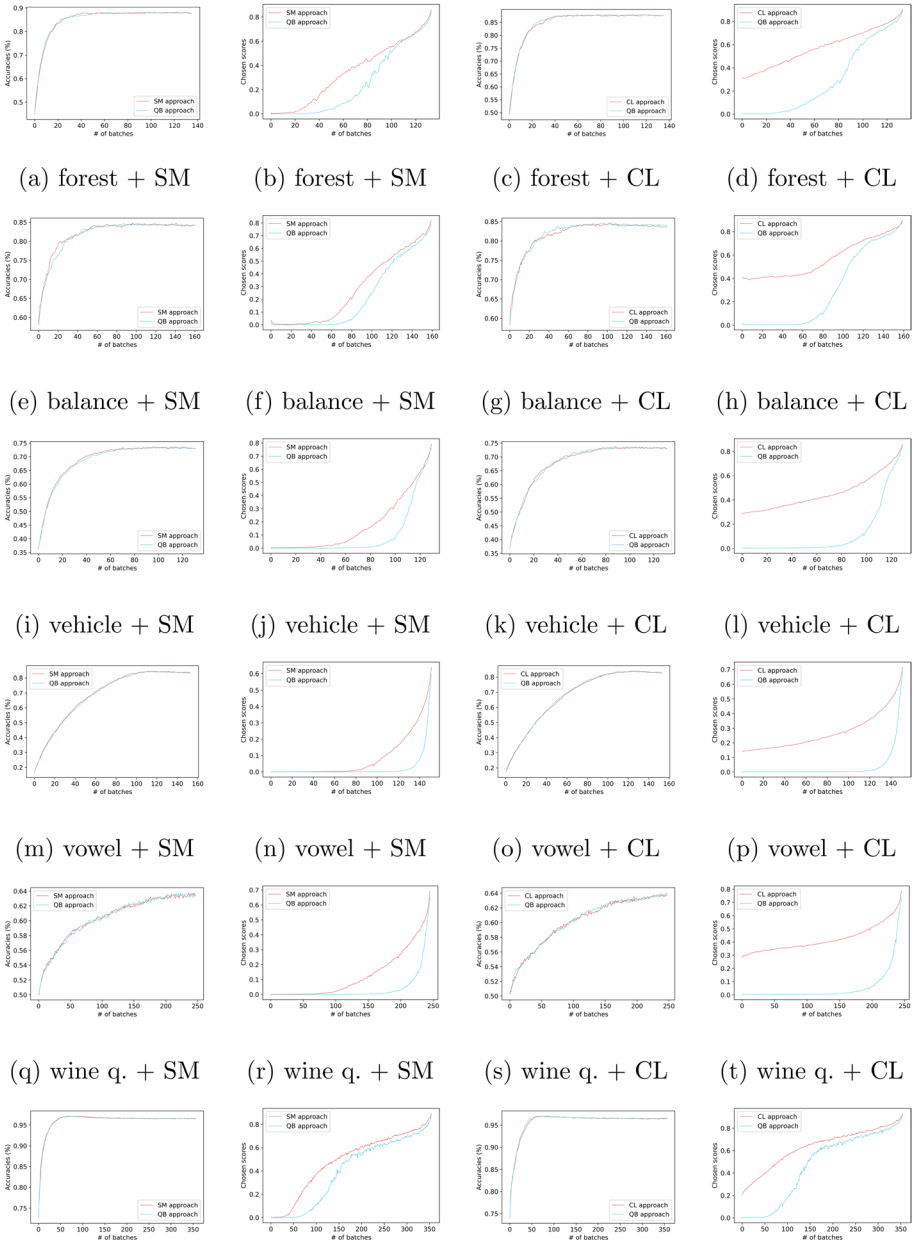
(i) glass + SM (j) glass + SM (k) glass + CL (l) glass + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. + SM (r) derma. + SM (s) derma. (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 11 Test accuracy and chosen score as the functions of the number of queries: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on noisy data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

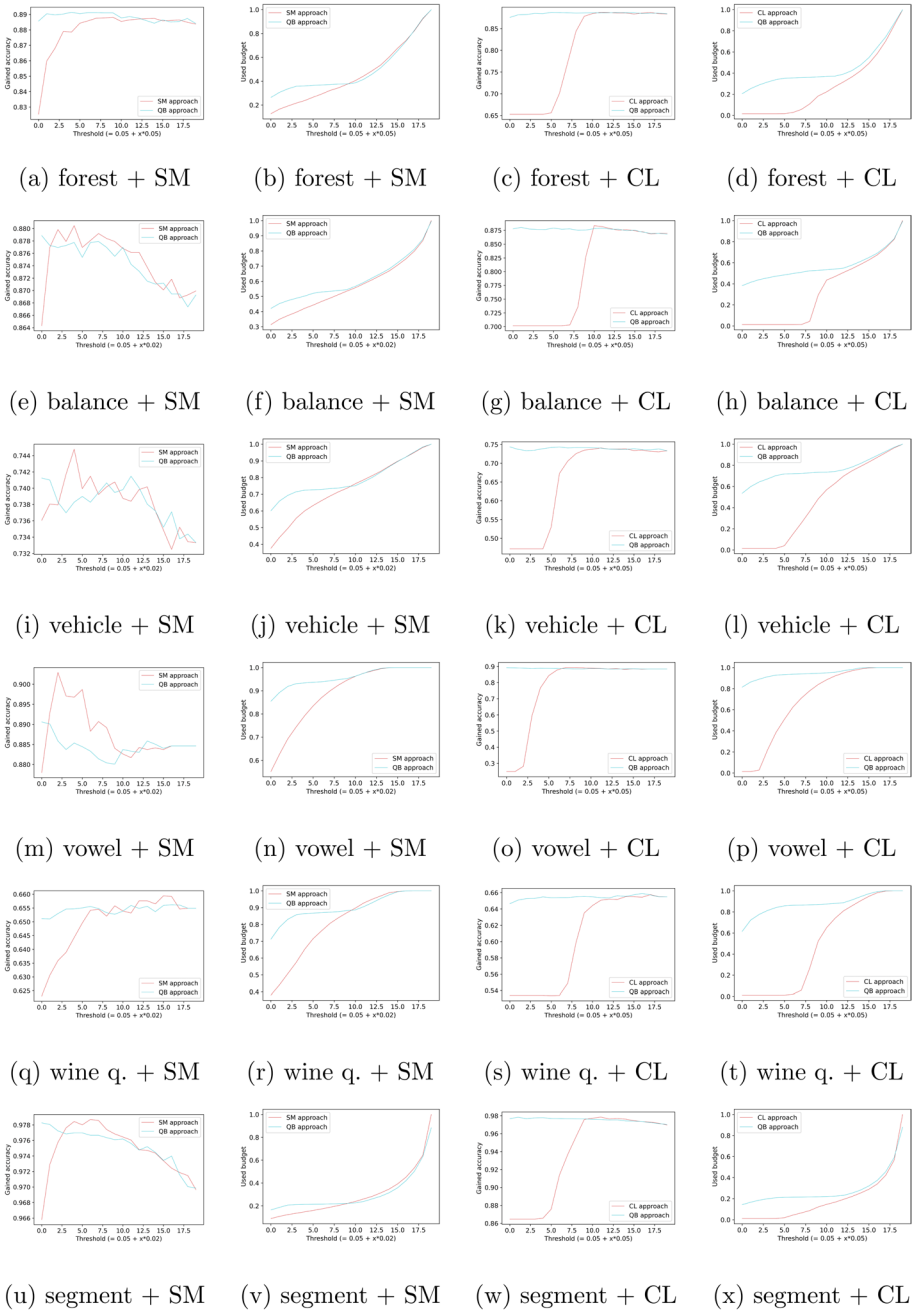
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(u) segment + SM (v) segment + SM (w) segment + CL (x) segment + CL

Fig. 12 Test accuracy and chosen score as the functions of the number of queries: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on noisy data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

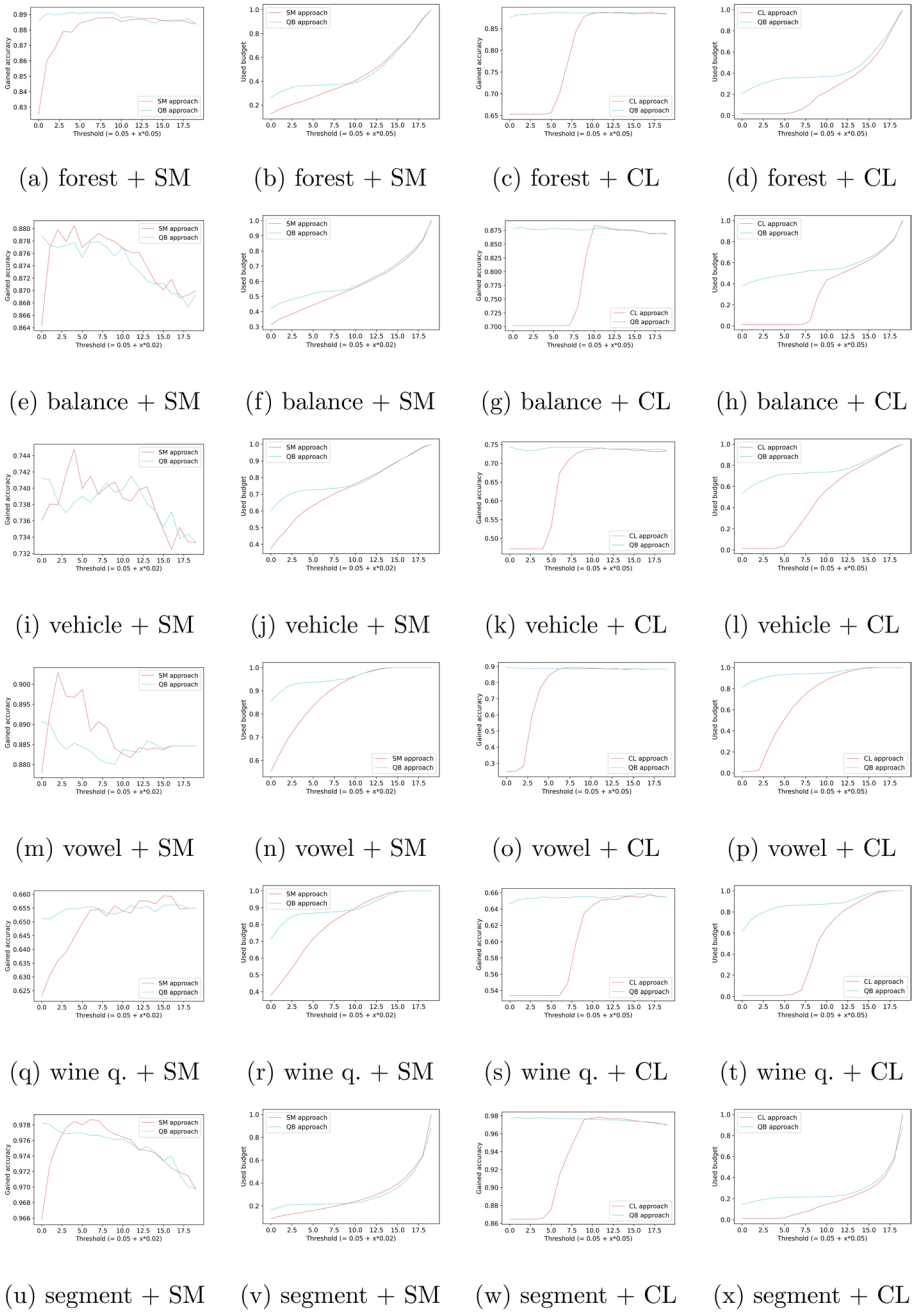
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(u) segment + SM (v) segment + SM (w) segment + CL (x) segment + CL

Fig. 13 Test accuracy and used budget as the functions of the threshold: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on clean data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

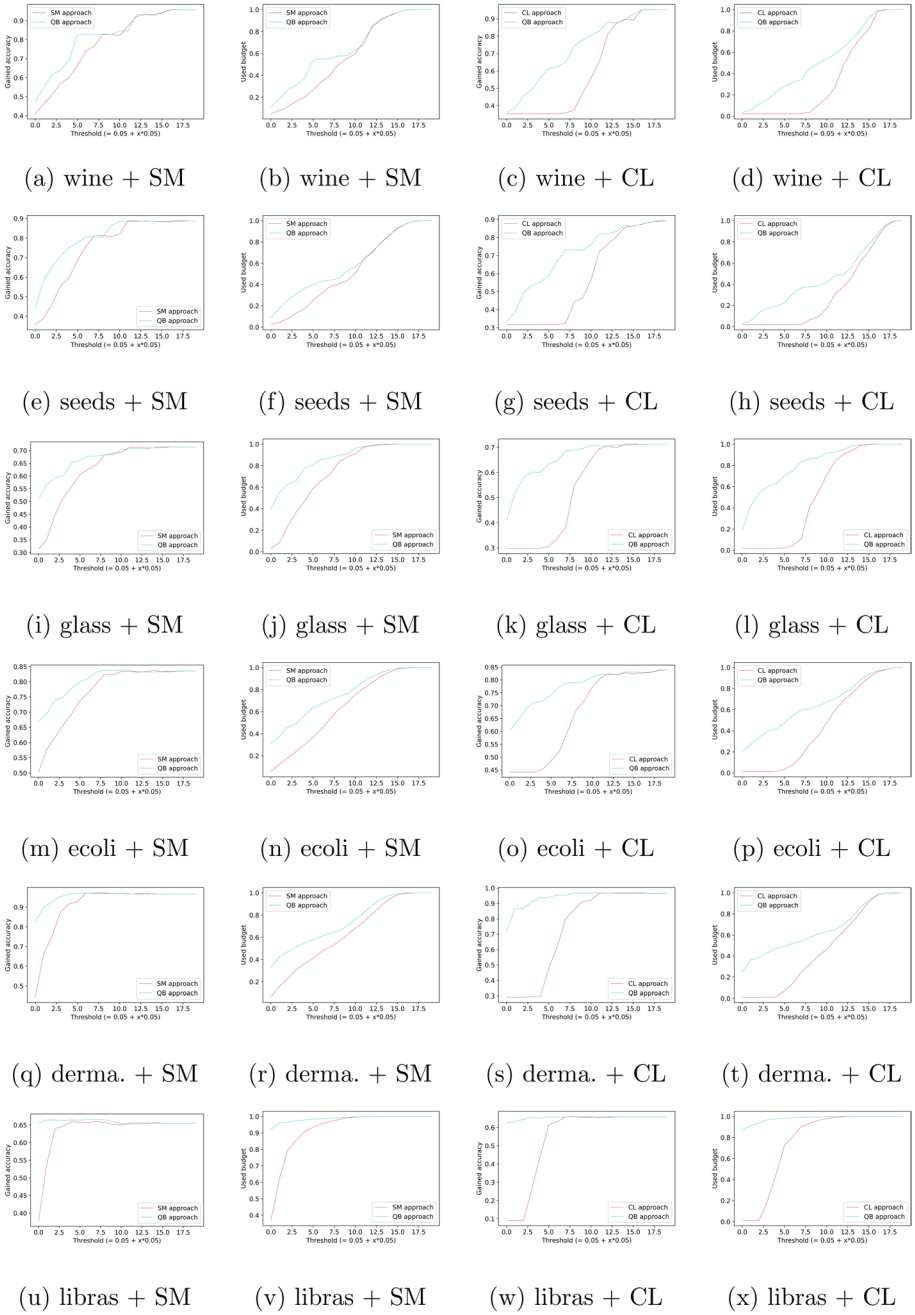
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(u) segment + SM (v) segment + SM (w) segment + CL (x) segment + CL

Fig. 14 Test accuracy and used budget as the functions of the threshold: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on clean data sets



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

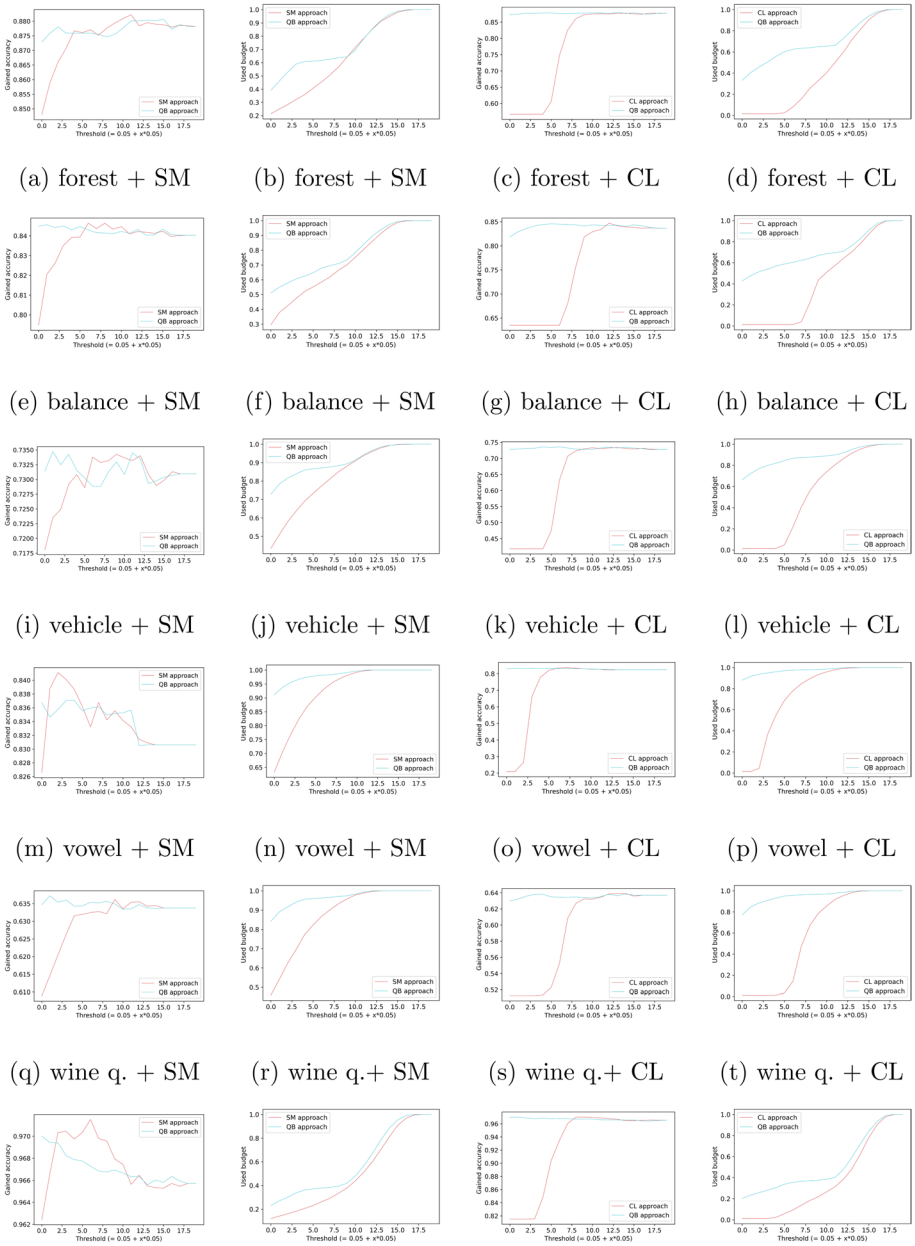
(i) glass + SM (j) glass + SM (k) glass + CL (l) glass + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. + SM (r) derma. + SM (s) derma. + CL (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 15 Test accuracy and used budget as the functions of the threshold: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on noisy data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

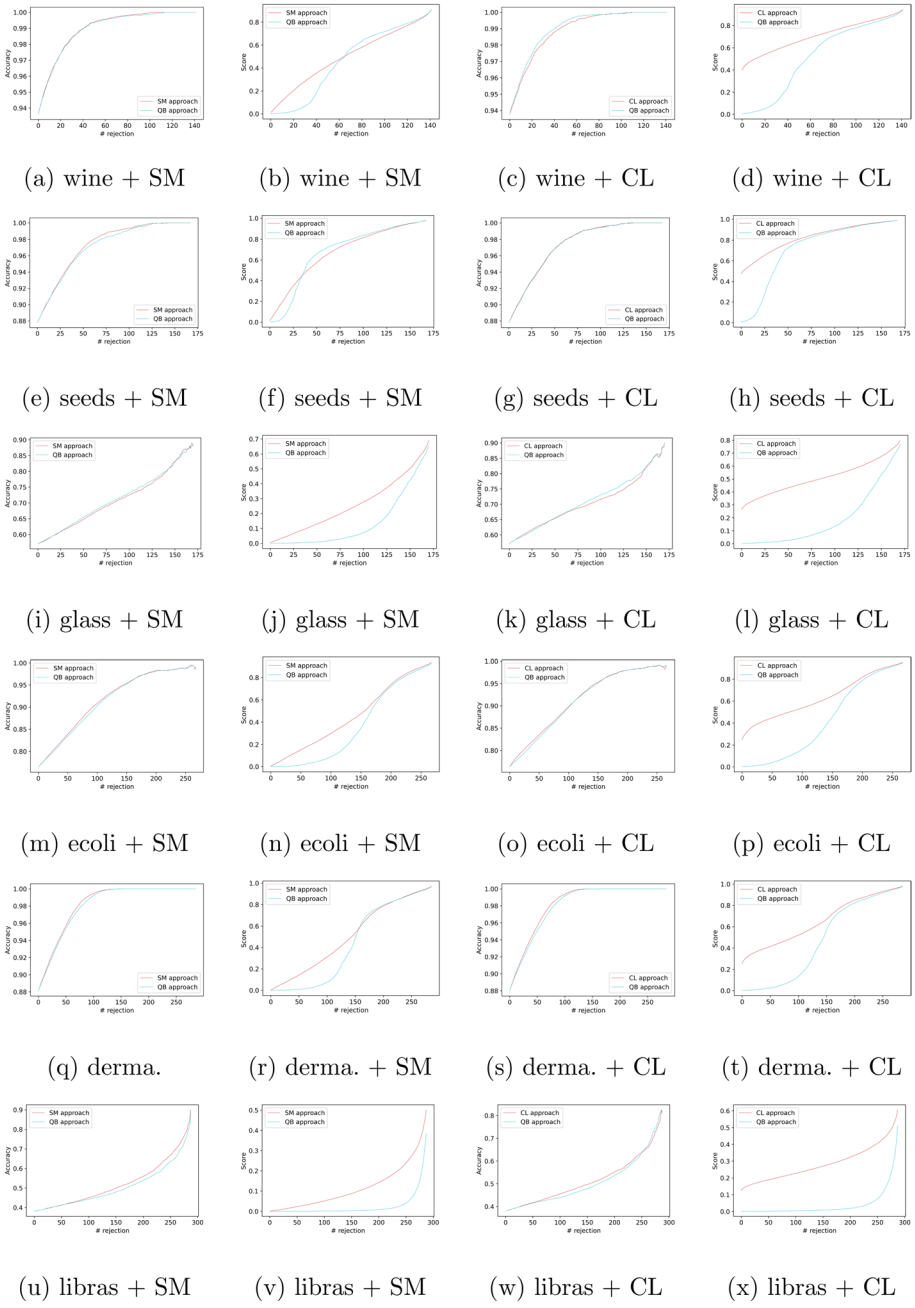
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q.+ SM (s) wine q.+ CL (t) wine q. + CL

(e) segment + SM (f) segment + SM (g) segment + CL (h) segment + CL

Fig. 16 Test accuracy and used budget as the functions of thresholds: 10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%) on noisy data sets



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

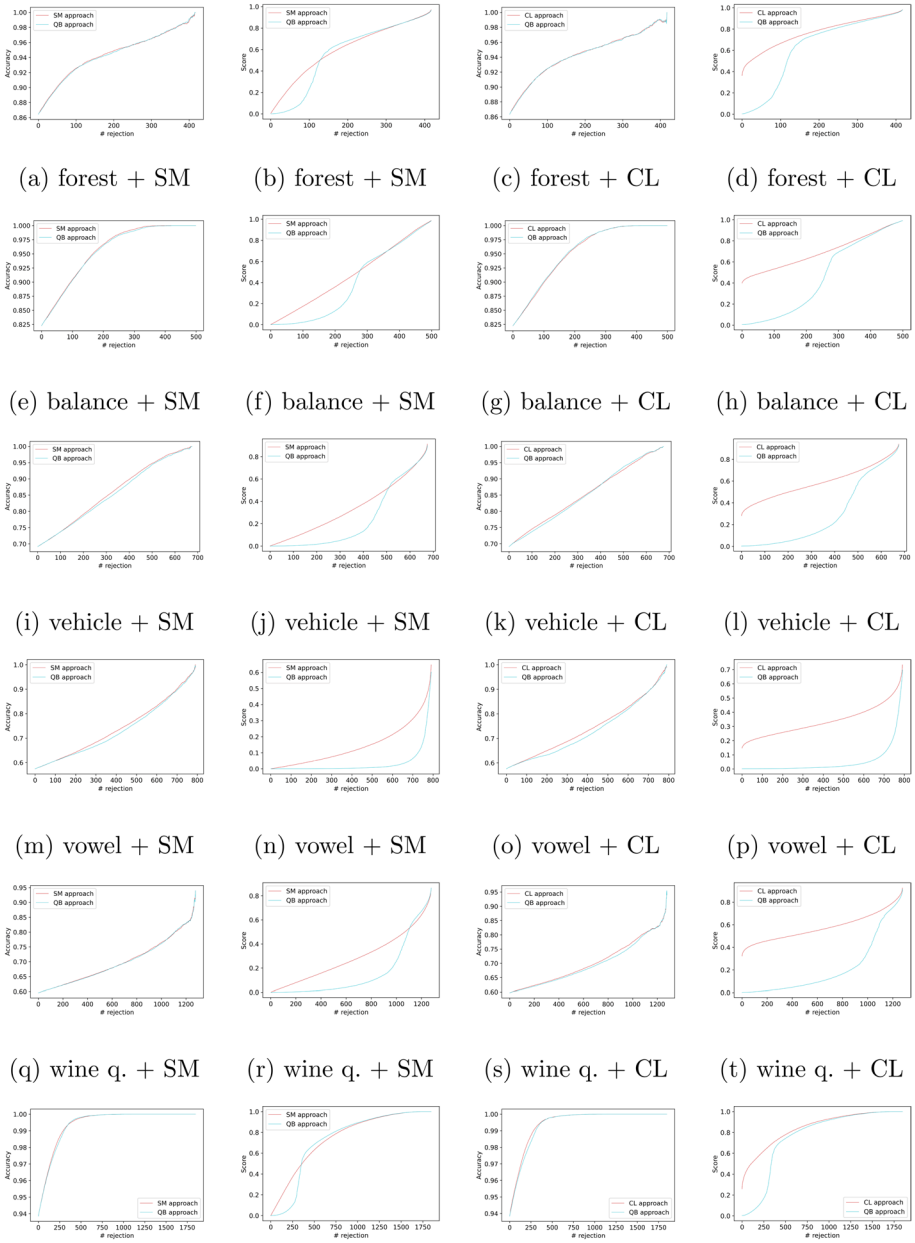
(i) glass + SM (j) glass + SM (k) glass + CL (l) glass + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. (r) derma. + SM (s) derma. + CL (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 17 Test accuracy and chosen score as the functions of the number of rejections: 20×5 cross-validation with (train, test) = (20%, 80%) on clean data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

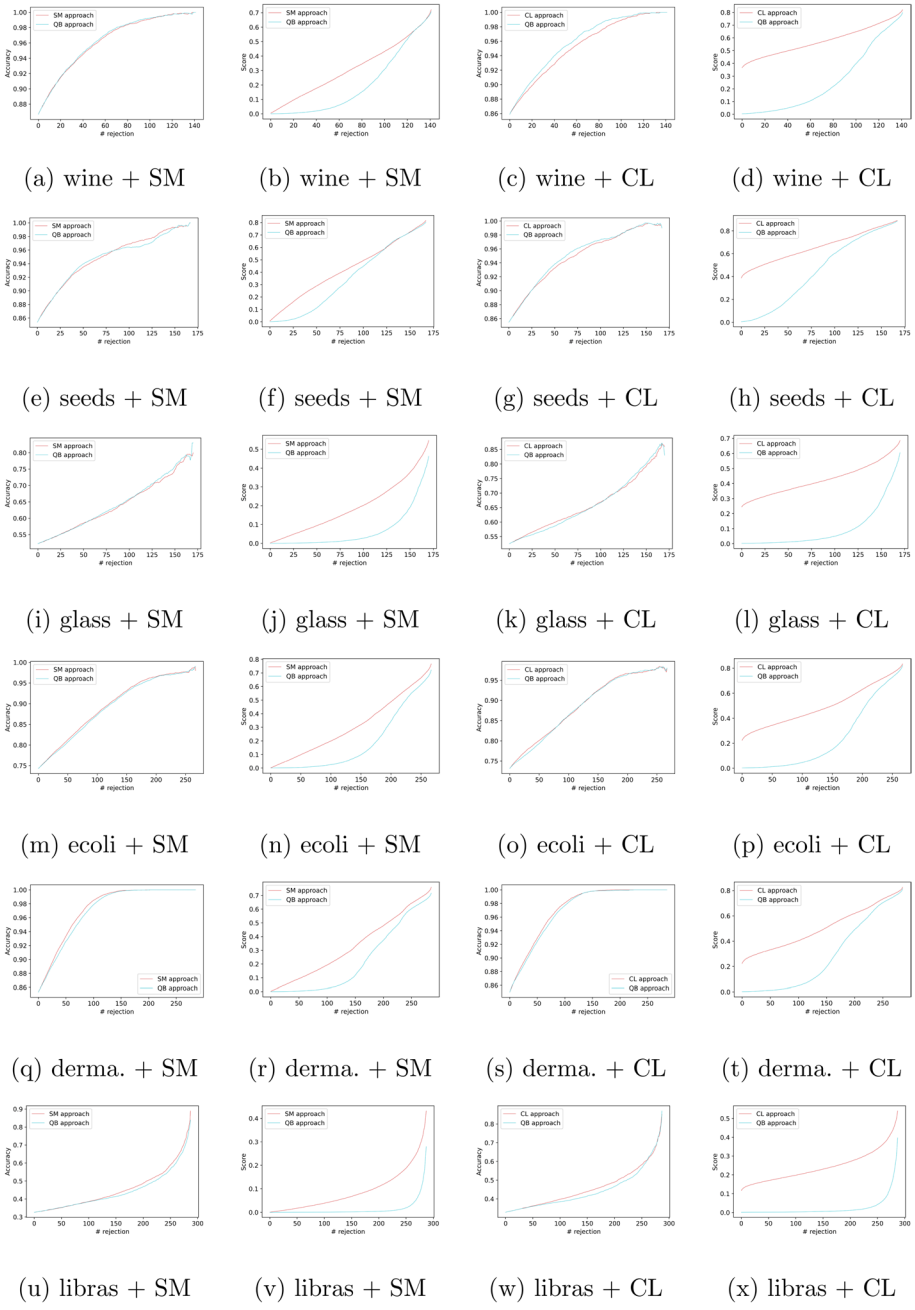
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(u) segment + SM (v) segment + SM (w) segment + CL (x) segment + CL

Fig. 18 Test accuracy and chosen score as the functions of the number of rejections: 20×5 cross-validation with (train, test) = (20%, 80%) on clean data sets



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

(i) glass + SM (j) glass + SM (k) glass + CL (l) glass + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. + SM (r) derma. + SM (s) derma. + CL (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 19 Test accuracy and chosen score as the functions of the number of rejections: 20×5 cross-validation with (train, test) = (20%, 80%) on noisy data sets

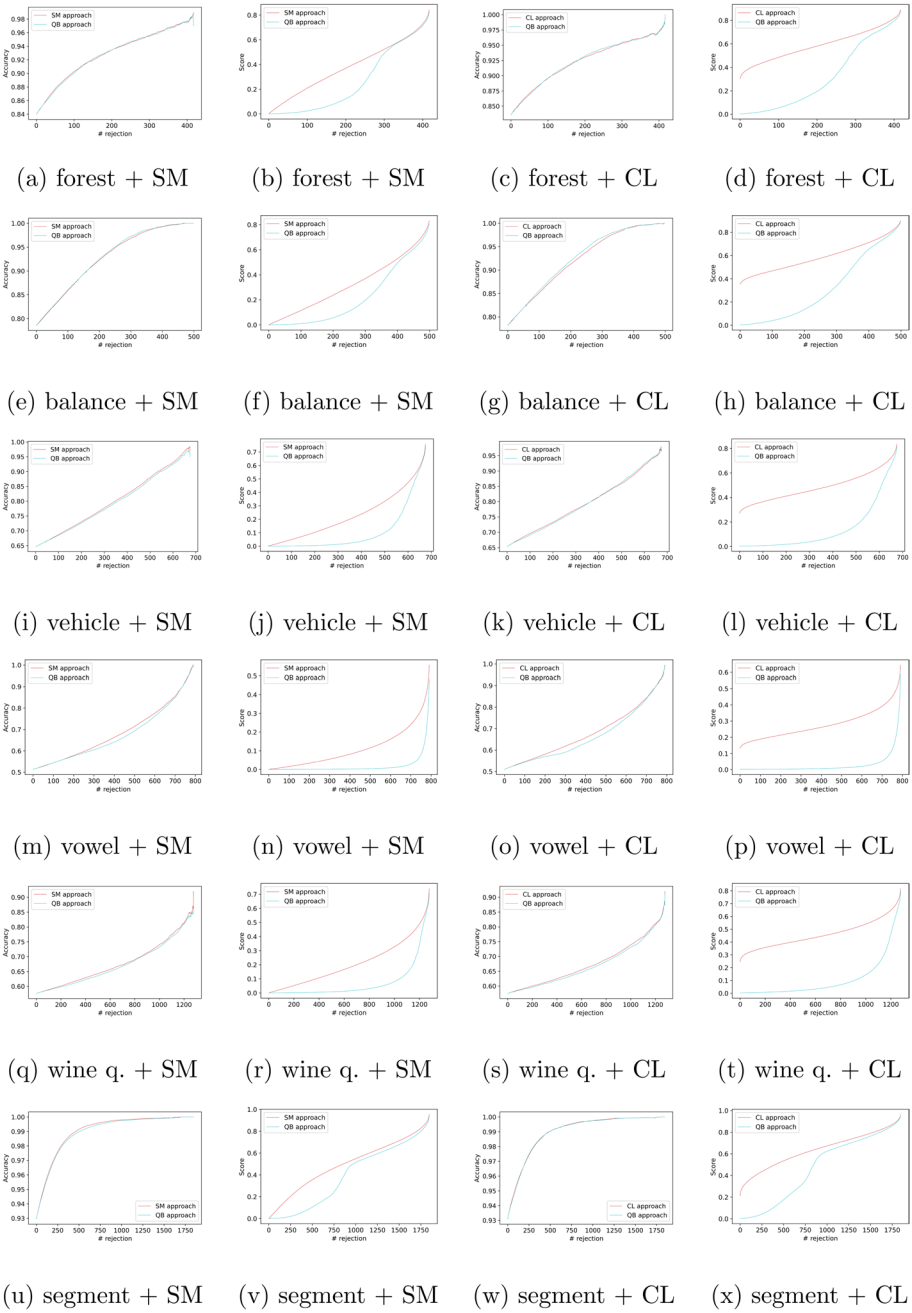
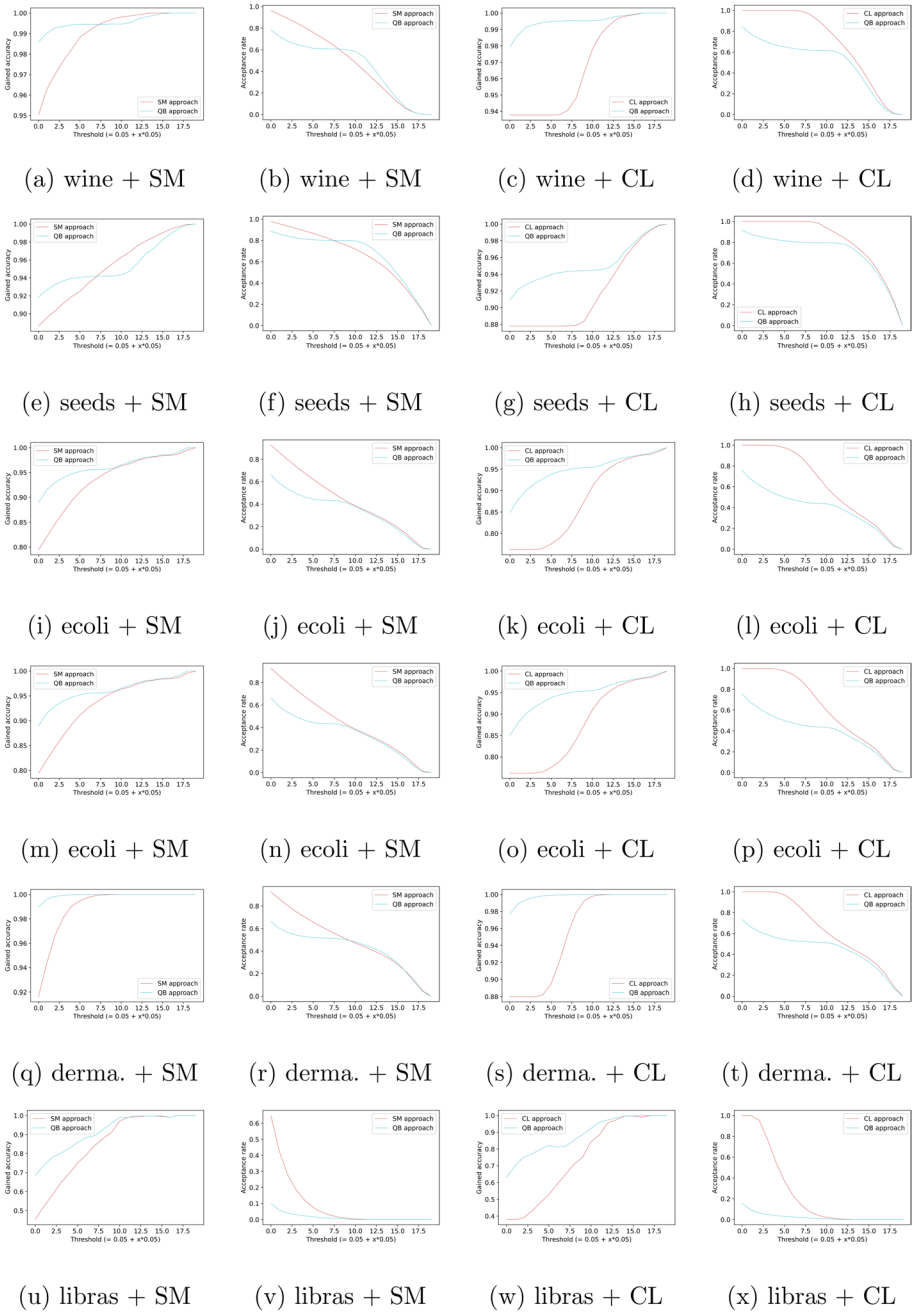


Fig. 20 Test accuracy and chosen score as the functions of the number of rejections: 20×5 cross-validation with (train, test) = (20%, 80%) on noisy data sets



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

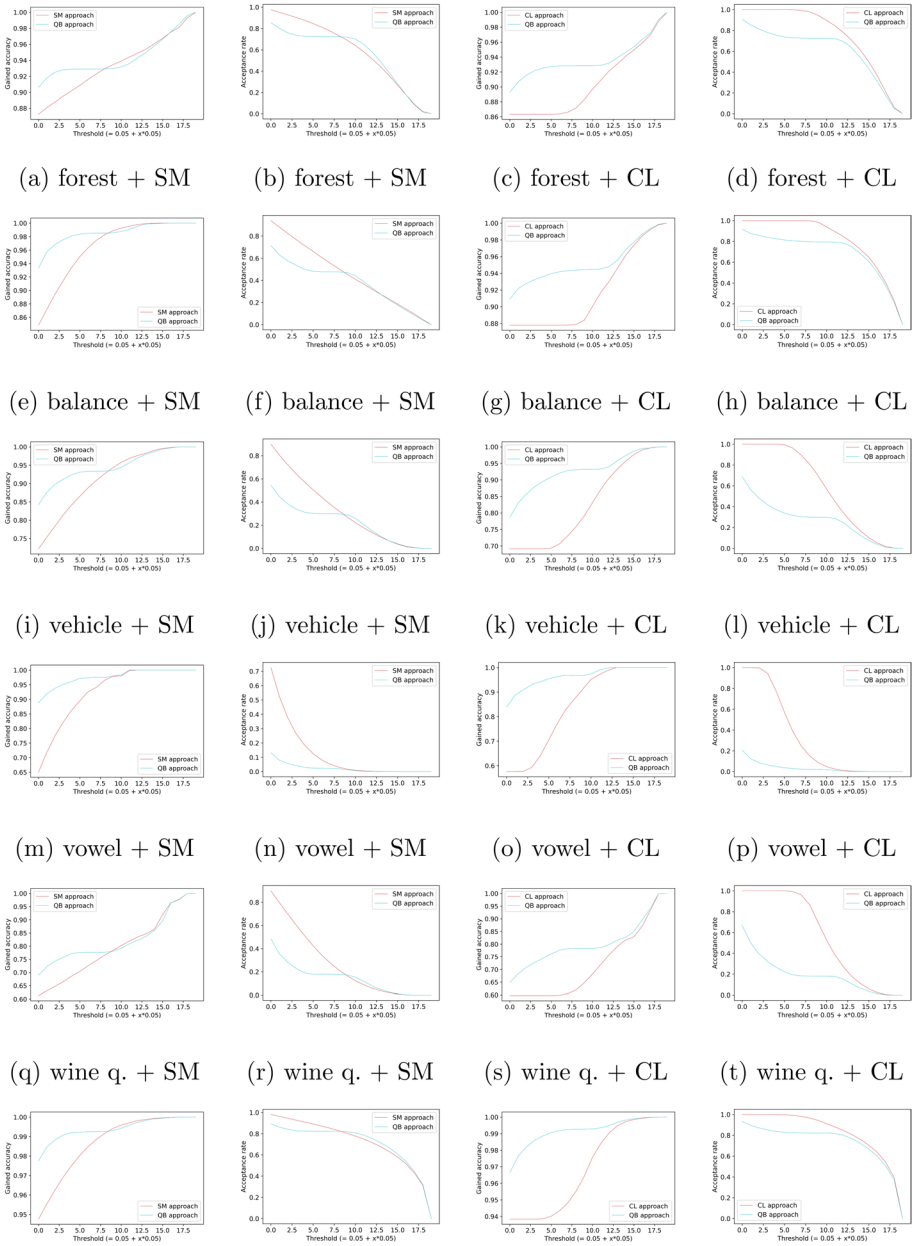
(i) ecoli + SM (j) ecoli + SM (k) ecoli + CL (l) ecoli + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. + SM (r) derma. + SM (s) derma. + CL (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 21 Test accuracy and acceptance rate as the functions of the threshold: 20×5 cross-validation with (train, test) = (20%, 80%) on clean data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

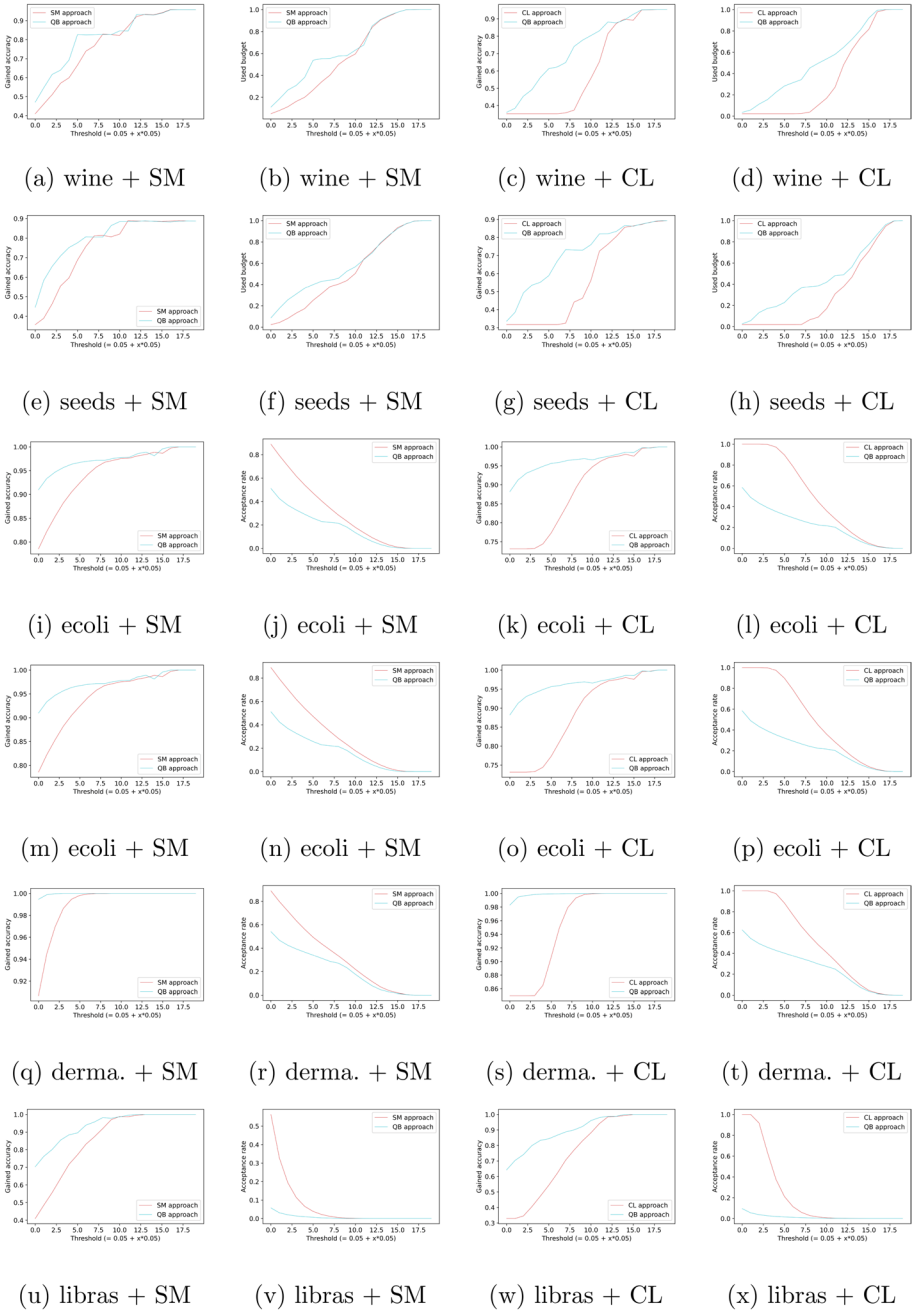
(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(u) segment + SM (v) segment + SM (w) segment + CL (x) segment + CL

Fig. 22 Test accuracy and acceptance rate as the functions of the threshold: 20×5 cross-validation with (train, test) = (20%, 80%) on clean data sets



(a) wine + SM (b) wine + SM (c) wine + CL (d) wine + CL

(e) seeds + SM (f) seeds + SM (g) seeds + CL (h) seeds + CL

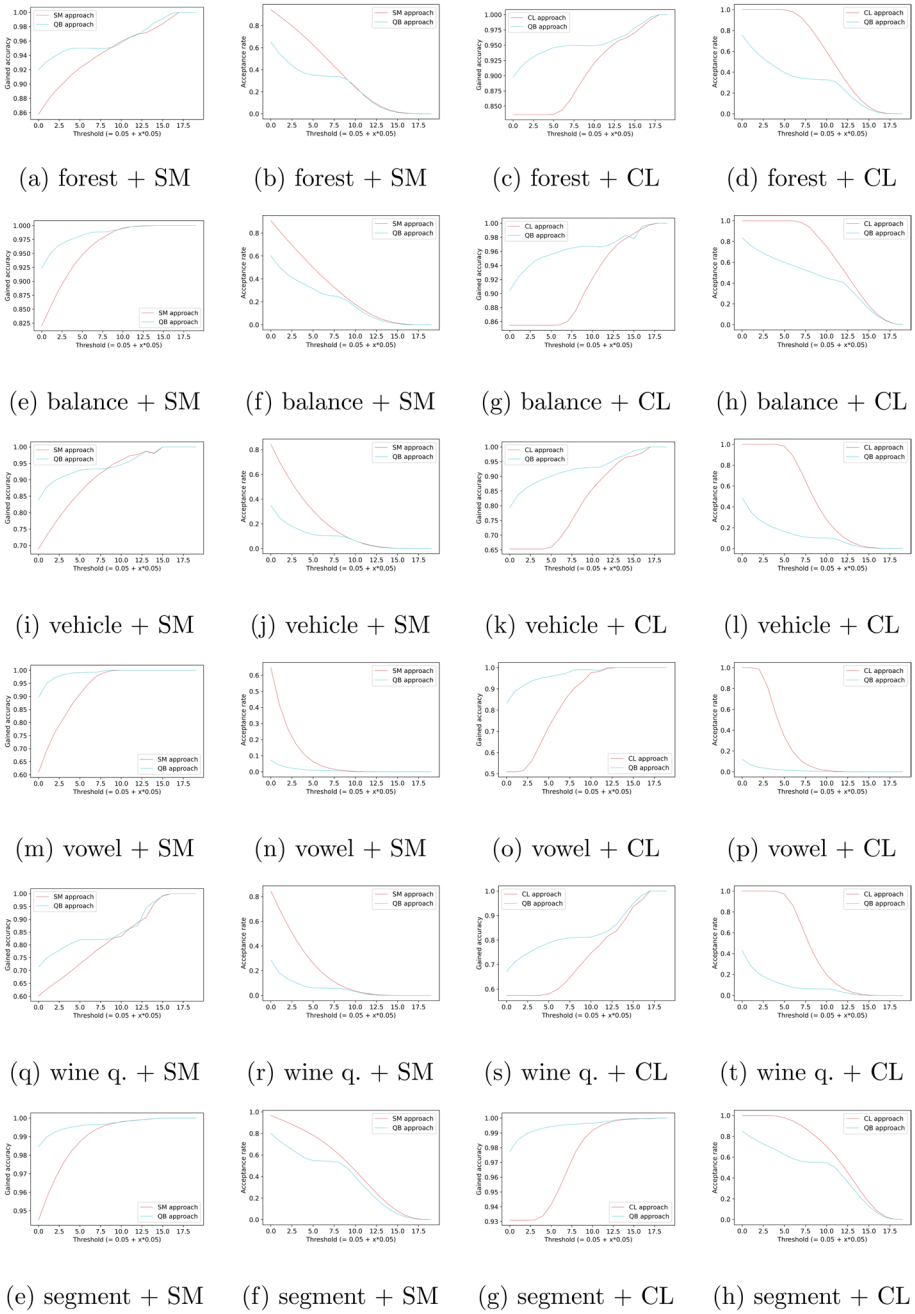
(i) ecoli + SM (j) ecoli + SM (k) ecoli + CL (l) ecoli + CL

(m) ecoli + SM (n) ecoli + SM (o) ecoli + CL (p) ecoli + CL

(q) derma. + SM (r) derma. + SM (s) derma. + CL (t) derma. + CL

(u) libras + SM (v) libras + SM (w) libras + CL (x) libras + CL

Fig. 23 Test accuracy and acceptance rate as the functions of the threshold: 20×5 cross-validation with (train, test) = (20%, 80%) on noisy data sets



(a) forest + SM (b) forest + SM (c) forest + CL (d) forest + CL

(e) balance + SM (f) balance + SM (g) balance + CL (h) balance + CL

(i) vehicle + SM (j) vehicle + SM (k) vehicle + CL (l) vehicle + CL

(m) vowel + SM (n) vowel + SM (o) vowel + CL (p) vowel + CL

(q) wine q. + SM (r) wine q. + SM (s) wine q. + CL (t) wine q. + CL

(e) segment + SM (f) segment + SM (g) segment + CL (h) segment + CL

Fig. 24 Test accuracy and acceptance rate as the functions of the threshold: 20×5 cross-validation with (train, test) = (20%, 80%) on noisy data sets

Table 5 Results on dermato. data set (in %)

RF as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
6	20	95	100	92.08	95	100	95.58	97
4	48	89.58	91.67	87.42	88.91	93.75	92.29	92.92
5	48	100	100	100	100	100	100	100
2	60	98.33	98.33	96.58	97.33	98.33	98.33	98.33
3	71	100	100	98.76	99.15	100	99.51	99.72
1	111	100	100	100	100	100	100	100
RF_BW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
6	20	100	100	98.25	99	100	100	100
4	48	93.75	93.75	91.91	92.5	91.67	91.67	91.67
5	48	100	100	100	100	100	100	100
2	60	93.33	98.33	94.53	96	96.67	93.55	94.46
3	71	100	100	99.1	99.26	100	99.25	99.44
1	111	100	100	99.68	99.82	100	100	100
RF_BSW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
6	20	100	100	100	100	100	100	100
4	48	91.67	97.92	92.43	94.58	93.75	91.18	92.08
5	48	100	100	100	100	100	100	100
2	60	93.33	96.67	92.58	94.33	98.33	95.97	96.67
3	71	100	100	99.01	99.44	100	99.51	99.72
1	111	100	100	100	100	100	100	100

Table 6 Results on glass data set (in %)

RF as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
Tableware	9	66.67	66.67	49.77	54.17	66.67	49.03	53.24
Containers	13	61.54	69.23	61.15	64.62	69.23	54.36	60
v. w. f	17	23.53	41.18	26.67	31.76	35.29	22.75	25.88
Headlamps	29	86.21	89.66	88.45	88.97	86.21	83.78	84.11
b. w. f	70	85.71	90	86.24	87.71	88.57	86.07	87.14
b. w. non-f	76	80.26	85.53	80.2	82.11	86.84	81.29	83.42
RF_BW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
Tableware	9	100	100	81.25	87.5	77.78	71.85	73.33
Containers	13	92.31	92.31	89.62	90.77	92.31	78.72	83.08
v. w. f	17	23.53	70.59	53.04	60	23.53	16.27	18.82
Headlamps	29	86.21	86.21	85	85.52	86.21	85	85.52
b. w. f	70	81.43	90	76.52	81.25	90	89	89.43
b. w. non-f	76	64.47	75	66.16	68.95	78.95	75.48	76.57
RF_BSW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
Tableware	9	100	100	100	100	88.89	81.11	84.44
Containers	13	92.31	92.31	92.31	92.31	92.31	84.23	87.69
v. w. f	17	29.41	82.35	51.64	61.62	17.65	11.84	13.38
Headlamps	29	86.21	93.1	87.64	89.66	86.21	85	85.52
b. w. f	70	81.43	91.43	77.33	82.29	88.57	88.57	88.57
b. w. non-f	76	61.84	71.05	61.62	64.57	77.63	74.47	75.36

Table 7 Results on wine qua. data set (in %)

RF as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
3	10	0	0	0	0	0	0	0
8	18	0	5.56	2.59	3.33	5.56	2.59	3.33
4	53	0	1.89	0.68	0.9	1.89	0.68	0.9
7	199	39.7	60.8	46.46	51.63	52.76	41.85	45.68
6	638	69.91	84.01	70.09	75.74	80.72	72.06	75.46
5	681	78.71	89.28	81.05	84.39	86.05	80.51	82.72
RF_BW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
3	10	0	10	6.5	8	0	0	0
8	18	11.11	27.78	20.93	23.33	16.67	6.62	8.61
4	53	18.87	33.96	23.06	26.46	15.09	5.54	7.25
7	199	66.33	84.92	69.7	75.72	60.8	47.75	52.5
6	638	58.62	73.04	58.15	63.61	77.27	68.57	71.87
5	681	74.89	82.82	74.02	77.11	85.61	79.32	81.62
RF_BSW as the base learner								
Class	# instances	RF acc.	SED-Ead			KL-Ead		
			Correctness	u_{65}	u_{80}	Correctness	u_{65}	u_{80}
3	10	0	10	6.5	8	0	0	0
8	18	11.11	16.67	14.72	15.56	11.11	11.11	11.11
4	53	18.87	41.51	26.7	30.75	9.43	6.13	7.55
7	199	69.35	85.43	69.27	75.48	67.84	58.17	62.31
6	638	58.31	74.45	57.22	63.56	86.52	72.47	78.46
5	681	75.18	83.55	74.41	77.71	90.16	82.91	85.99

Acknowledgements This work was mostly conducted when Haifei Zhang was at the Université de Technologie de Compiègne. Vu-Linh Nguyen has been funded by the Junior Professor Chair in Trustworthy AI (Ref. ANR-R311CHD).

Funding Open access funding provided by Université de Technologie de Compiègne. This work was mostly conducted when Haifei Zhang was at the Université de Technologie de Compiègne.

Vu-Linh Nguyen has been funded by the Junior Professor Chair in Trustworthy AI (Ref. ANR-R311CHD).

Data availability The data sets used in this submission are downloaded from the UCI Machine Learning Repository.

The sourcode has been made publicly available on https://github.com/Haifei-ZHANG/Probability-Sets-Model.

Declarations

Conflict of interest We declare that we have no conflict of interest.

Ethical approval We declare that this research did not require Ethical approval.

Consent for publication All the authors of this manuscript consent to its publication.

References

Abellán, J., Mantas, C. J., & Castellano, J. G. (2017). A random forest approach using imprecise probabilities. *Knowledge-Based Systems, 134*, 72–84.

Abellan, J., & Masegosa, A. R. (2012). Imprecise classification with credal decision trees. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 20*(05), 763–787.

Alarcon, Y. C. C., & Destercke, S. (2021). Imprecise gaussian discriminant classification. *Pattern Recognition, 112*, 107739.

Antonucci, A., & De Campos, C. P. (2011). Decision making by credal nets. In *2011 Third international conference on intelligent human-machine systems and cybernetics* (pp. 201–204). IEEE.

Augustin, T., Coolen, F. P., De Cooman, G., et al. (2014). *Introduction to imprecise probabilities*. John Wiley & Sons.

Bock, J. D., Campos, C. P. d., & Antonucci, A. (2014). Global sensitivity analysis for map inference in graphical models. In *Proceedings of the 27th international conference on neural information processing systems (NIPS)* (pp. 2690–2698).

Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Cha, S. H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical models and Methods in Applied Sciences, 1*(4), 300–307.

Chow, C. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory, 16*(1), 41–46.

Condessa, F., Bioucas-Dias, J., & Kovačević, J. (2017). Performance measures for classification systems with rejection. *Pattern Recognition, 63*, 437–450.

Corani, G., & Antonucci, A. (2014). Credal ensembles of classifiers. *Computational Statistics & Data Analysis, 71*, 818–831.

Corani, G., & De Campos, C. P. (2010). A tree augmented classifier based on extreme imprecise dirichlet model. *International Journal of Approximate Reasoning, 51*(9), 1053–1068.

Corani, G., & Zaffalon, M. (2008). Learning reliable classifiers from small or incomplete data sets: The naive credal classifier 2. *Journal of Machine Learning Research, 9*(4), 581.

Datta, B. N. (2010). Numerical linear algebra and applications, vol 116. Siam

Decadt, A., Erreygers, A., & De Bock, J., et al. (2022). Decision-making with e-admissibility given a finite assessment of choices. In *International conference on soft methods in probability and statistics* (pp. 96–103). Springer.

Del Coz, J. J., Díez, J., & Bahamonde, A. (2009). Learning nondeterministic classifiers. *Journal of Machine Learning Research, 10*(10), 2273.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the First International workshop on multiple classifier systems (MCS)* (pp. 1–15). Springer.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the 17th international conference on artificial intelligence (IJCAI)* (pp. 973–978).

Gibbs, A. L., & Su, F. E. (2002). On choosing and bounding probability metrics. *International Statistical Review, 70*(3), 419–435.
Gilet, C., Guyomard, M., Destercke, S., et al. (2024). Softmin discrete minimax classifier for imbalanced classes and prior probability shifts. *Machine Learning, 113*(2), 605–645.
Haixiang, G., Yijing, L., Shang, J., et al. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220–239.
Ho, T. K. (1995). Random decision forests. In textitProceedings of 3rd international conference on document analysis and recognition (ICDAR) (pp. 278–282). IEEE.
Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning, 110*, 457–506.
Jansen, C., Schollmeyer, G., & Augustin, T. (2022). Quantifying degrees of e-admissibility in decision making with imprecise probabilities. In *Reflections on the Foundations of Probability and Statistics: Essays in Honor of Teddy Seidenfeld* (pp. 319–346). Springer.
Jospin, L. V., Laga, H., Boussaid, F., et al. (2022). Hands-on bayesian neural networks–A tutorial for deep learning users. *IEEE Computational Intelligence Magazine, 17*(2), 29–48.
Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). Comparing boosting and bagging techniques with noisy and imbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41*(3), 552–568.
Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence, 5*(4), 221–232.
Lachiche, N., & Flach, P. A. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using roc curves. In *Proceedings of the 20th international conference on machine learning (ICML)* (pp. 416–423).
Lee, L. (1999). Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the association for computational linguistics on computational linguistics (ACL)* (pp. 25–32).
Lemay, A., Hoebel, K., Bridge, C. P., et al. (2022). Improving the repeatability of deep learning models with monte carlo dropout. *npj Digital Medicine, 5*(1), 174.
Levi, I. (1983). The enterprise of knowledge: An essay on knowledge, credal probability, and chance. MIT press
Mindermann, S., Brauner, J. M., & Razzak, M. T., et al. (2022). Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th international conference on machine learning (ICML)* (pp. 15630–15649). PMLR.
Montes, I., Miranda, E., & Destercke, S. (2020). Unifying neighbourhood and distortion models: Part i-new results on old models. *International Journal of General Systems, 49*(6), 602–635.
Mortier, T., Wydmuch, M., Dembczyński, K., et al. (2021). Efficient set-valued prediction in multi-class classification. *Data Mining and Knowledge Discovery, 35*(4), 1435–1469.
Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.
Murty, K. G. (1983). *Linear programming*. Springer.
Nakharutai, N., Troffaes, M. C., & Caiado, C. C. (2019). Improving and benchmarking of algorithms for decision making with lower previsions. *International Journal of Approximate Reasoning, 113*, 91–105.
Nguyen, V. L., Destercke, S., Masson, M. H., et al. (2018). Reliable multi-class classification based on pairwise epistemic and aleatoric uncertainty. In *Proceedings of the 27th international joint conference on artificial intelligence (IJCAI)* (pp. 5089–5095).
Nguyen, V. L., Hüllermeier, E., Rapp, M., et al. (2020). On aggregation in ensembles of multilabel classifiers. In *Proceedings of the 23rd international conference on discovery science (DS)* (pp. 533–547). Springer.
Nguyen, V. L., Yang, Y., de Campos, C. P. (2023a). Probabilistic multi-dimensional classification. In *Proceedings of the 39th conference on uncertainty in artificial intelligence (UAI)*.
Nguyen, V. L., Zhang, H., & Destercke, S. (2023b). Learning sets of probabilities through ensemble methods. In *Proceedings of the 17th European conference on symbolic and quantitative approaches to reasoning with uncertainty (ECSQARU)*
Nguyen, V. L., & Hüllermeier, E. (2021). Multilabel classification with partial abstention: Bayes-optimal prediction under label independence. *Journal of Artificial Intelligence Research, 72*, 613–665.
Nguyen, V. L., Shaker, M. H., & Hüllermeier, E. (2022). How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning, 111*(1), 89–122.
O'brien, D. B., Gupta, M. R., & Gray, R. M. (2008). Cost-sensitive multi-class classification from probability estimates. In *Proceedings of the 25th international conference on machine learning (ICML)* (pp. 712–719).
Pugh, C. C. (2015). Real mathematical analysis. Undergraduate Texts in Mathematics

Quost, B., & Destercke, S. (2018). Classification by pairwise coupling of imprecise probabilities. *Pattern Recognition, 77*, 412–425.
Rahimian, H., & Mehrotra, S. (2019). Distributionally robust optimization: A review. arXiv preprint arXiv:1908.05659
Rockafellar, R. T. (1993). Lagrange multipliers and optimality. *SIAM Review, 35*(2), 183.
Sáez, J. A., Krawczyk, B., & Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition, 57*, 164–178.
Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4), e1249.
Seidenfeld, T., Schervish, M. J., & Kadane, J. B. (1995). A representation of partially ordered preferences. *The Annals of Statistics, 23*(6), 2168–2217.
Shaker, M. H., & Hüllermeier, E. (2020). Aleatoric and epistemic uncertainty with random forests. In *Proceedings of the eighteenth international symposium on intelligent data analysis (IDA)* (pp. 444–456).
Sriperumbudur, B. K., Gretton, A., Fukumizu, K., et al. (2010). Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research, 11*, 1517–1561.
Steinley, D. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology, 59*(1), 1–34.
Troffaes, M. C. (2007). Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning, 45*(1), 17–29.
Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*(5), 988–999.
Yang, G., Destercke, S., & Masson, M. H. (2014). Nested dichotomies with probability sets for multi-class classification. In *Proceedings of the Twenty-first European conference on artificial intelligence (ECAI)* (pp. 363–368).
Zaffalon, M., Corani, G., & Mauá, D. (2012). Evaluating credal classifiers by utility-discounted predictive accuracy. *International Journal of Approximate Reasoning, 53*(8), 1282.
Zhang, H., Quost, B., & Masson, M. H. (2023). Cautious decision-making for tree ensembles. In *Proceedings of the 17th European conference on symbolic and quantitative approaches to reasoning with uncertainty (ECSQARU)*.
Zhu, J., Wang, H., Hovy, E., et al. (2010). Confidence-based stopping criteria for active learning for data annotation. *ACM Transactions on Speech and Language Processing, 6*(3), 1–24.