Proceedings of Machine Learning Research 204:1–18, 2023 Conformal and Probabilistic Prediction with Applications

# Self Learning using Venn-Abers predictors

**Côme Rodriguez** COME.RODRIG@GMAIL.COM
*Université de Technologie de Compiègne, France*

**Vitor Martin Bordini** VITOR.MARTIN-BORDINI@HDS.UTC.FR
**Sébastien Destercke** SEBASTIEN.DESTERCKE@HDS.UTC.FR
**Benjamin Quost** BENJAMIN.QUOST@HDS.UTC.FR
*Heudiasyc lab, Université de Technologie de Compiègne, France*

**Editors:** Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

## Abstract

In supervised learning problems, it is common to have a lot of unlabeled data, but little labeled data. It is then desirable to leverage the unlabeled data to improve the learning procedure. One way to do this is to have a model predict "pseudo-labels" for the unlabeled data, so as to use them for learning. In self-learning, the pseudo-labels are provided by the very same model to which they are fed. As these pseudo-labels are by nature uncertain and only partially reliable, it is then natural to model this uncertainty and take it into account in the learning process, if only to robustify the self-learning procedure. This paper describes such an approach, where we use *Venn-Abers Predictors* to produce calibrated *credal labels* so as to quantify the pseudo-labeling uncertainty. These labels are then included in the learning process by optimizing an adapted loss. Experiments show that taking into account pseudo-label uncertainty both robustifies the self-learning procedure and allows it to converge faster in general.

**Keywords:** Self learning, Venn-Abers predictors, credal labels

## 1. Introduction

The use of data and machine learning becomes more and more frequent, in part due to more and more data being generated by users. However, a large part of these data are unlabeled. In classical supervised learning, these latter cannot be used to train the model. This is particularly problematic in situations where the amount of labeled data is small, which usually happens when obtaining expertise is costly.

Semi-supervised learning techniques address this particular situations by proposing methods that learn from both labeled and unlabeled data. There are many such techniques, and we refer for example to Van Engelen and Hoos (2020) for a recent survey and taxonomy. Self-learning is a specific semi-supervised learning approach which consists in replacing missing labels with model predictions, and then incorporate these data in the training set. Such predicted labels are often called *pseudo-labels*. While the idea of self-learning and automatic labeling is not new (Yarowsky, 1995) and has been applied successfully for quite some time in different fields such as image processing (Dópido et al., 2013), it has recently known a revival of interest (Sohn et al., 2020; Petrovai and Nedevschi, 2022). However, replacing unknown labels by wrong predictions may lead to lower performances, which can even be made significantly worse in some cases.

A solution to this issue is to replace the unknown labels by uncertain labels, typically probabilistic ones Leistner et al. (2009), and to train the model by using an adequate loss function (e.g., cross-entropy). However, such probabilistic estimates are themselves not guaranteed to be accurate and reliable estimates, for instance when the learning methods either rely on too strong assumptions Domingos and Pazzani (1996) or when they display too high variance Provost and Domingos (2003). A classical way to obtain more reliable probabilistic estimates is to calibrate those probabilities (Song et al., 2021). However, such calibrated probabilities may not be able to reflect how reliable the estimates are, in the sense that they will not properly quantify their *epistemic uncertainty*, e.g. whether or not they rely on a lot of data. In order to even augment the expressiveness of provided pseudo-labels, Lienen and Hüllermeier (2021) recently proposed to consider specific convex sets of probabilities as pseudo-labels, and more recently to use conformal prediction (see Lienen et al. (2022)) in order to derive such convex sets.

While the probability sets considered by Lienen and Hüllermeier (2021) have the advantage to be simple and not to increase significantly computational costs, they have the caveat that precise probabilities cannot be modeled, since such sets will always contain at least one degenerate probability distribution which puts all the probability mass on a given class[1]. In this paper, we consider the same idea of using probability sets as pseudo-labels, but rather than considering conformal prediction outputs, we use Venn predictors (Lambrou et al., 2015), and more precisely Venn-Abers predictors (Vovk and Petej, 2012), as we focus on the binary case.

The paper is organized as follows. Reminders about learning, credal labels and Venn-Abers predictors are given in Section 2. Section 3 then discusses how learning from credal labels issued from Venn-Abers predictors can be performed by adapting the Kullback-Leibler divergence to this setting, and presents our proposed self-learning scheme. Section 4 provides experimental results on various data sets, showing that using calibrated credal labels generally improves the self-learning procedure, and can prevent from potential model degradation.

## 2. Preliminaries

These preliminaries will introduce the basic building blocks needed to present our proposed approach.

[1] The reason for this is that they are possibility distributions, see Dubois and Prade (1992).

### 2.1. Reminders on semi-supervised and self-learning

The classical semi-supervised setting considers both a labeled data set

$$\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^n \subseteq (\mathcal{X} \times \mathcal{Y})^n$$

and an unlabeled data set

$$\mathcal{D}_U = \{(x_i, \mathcal{Y})\}_{i=n+1}^m \subseteq (\mathcal{X} \times 2^{\mathcal{Y}})^{m-n},$$

assumed to come from the same underlying distribution, where $\mathcal{X}, \mathcal{Y}$ are the input and categorical discrete output spaces. In this paper, we will consider the binary classification setting, i.e. $\mathcal{Y} = \{0, 1\}$.

The goal of semi-supervised learning methods, and of learning methods in general, is usually to learn a real-valued scoring function $h_\theta : \mathcal{X} \to \mathbb{R}$ from the available data $\mathcal{D}_L \cup \mathcal{D}_U$. The model $h_\theta$ can then be used to make a prediction $\hat{y}$ for any observation $x$ by using a threshold $c$, that is $\hat{y} = 1$ if $h_\theta(x) > c, 0$ else. Typical examples are SVMs with $h_\theta(x) \in (-\infty, \infty)$ and $c = 0$, or logistic regression with $h_\theta(x) \in [0, 1]$ and $c = 0.5$. Note that a sigmoid transform makes it possible to retrieve the second case from the first one. In this paper and to facilitate reading, we will assume that $h_\theta \in [0, 1]$, and may be interpreted as a (non-calibrated) probability, with $h_\theta(x) = \hat{p}(y = 1|x)$.

In a nutshell, self-supervised learning (Triguero et al., 2015) in its basic form consists in (1) learning a model $h_{\theta^0}$ from $\mathcal{D}_L$, (2) to select some unlabeled data $x \in \mathcal{D}_U$ and complete them by a prediction $h_{\theta^0}(x)$, before (3) $(x, h_{\theta^0}(x))$ is added to $\mathcal{D}_L$: the procedure can then be repeated iteratively, learning a new model $h_{\theta^1}$ and so on, until a stopping condition is met. While such an approach can benefit from an accurate model $h_{\theta^j}$, it can also suffer from the incorporation of inaccurate or unreliable predictions to the data set. To circumvent such an issue, this paper proposes to consider an approach where all data points from $\mathcal{D}_U$ are labeled at once, yet not by precise labels or single probabilistic labels, but by convex sets of probabilities—which reduce to intervals in binary classification. The idea of using such convex sets is that they provide a generic, rich and flexible way to represent model predictions together with their associated uncertainty, thus bypassing the need to select the pseudo-labeled data to be added to $\mathcal{D}_L$, and allowing for a differentiation between reliable and unreliable predictions.

### 2.2. Credal sets and credal Learning

A credal set (Destercke and Dubois, 2014) is a closed convex set of probability distributions. We will denote by $\Delta_{\mathcal{Y}}$ the set of probabilities over $\mathcal{Y}$, and by $K \subseteq \Delta_{\mathcal{Y}}$ a credal set defined over $\mathcal{Y}$. The advantage of credal sets is that they can model all kinds of knowledge about a label, from missing and partial labels to probabilistic and classical ones. In the present case of binary spaces, one key advantage is that any credal set can be summarized by the interval $[\underline{p}(1), \overline{p}(1)]$, with the corresponding

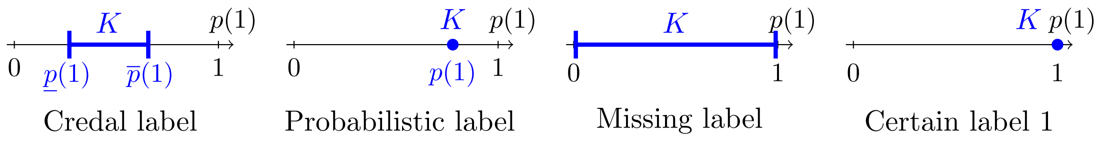


Figure 1: Some examples of credal labels

bounds for $p(0)$ being retrieved by using the relation $p(0)+p(1) = 1$. A probabilistic label is then modeled by $\underline{p}(1) = \overline{p}(1)$, and a missing (or totally unreliable) label by $[\underline{p}(1), \overline{p}(1)] = [0, 1]$. Figure 1 illustrates various situations, with their associated probability intervals.

A question is then to know how to learn from such credal labels. In classical supervised binary classification via loss minimization, we train a classifier $h_\theta$ on data having the form $(x, y)$ by optimizing a loss function $\mathcal{L}$ using the hard label $y$ and the probability output by $h_\theta$. Here, we consider a loss function defined on the space of probabilities, i.e., $\mathcal{L} : \Delta_{\mathcal{Y}} \times \Delta_{\mathcal{Y}} \to \mathbb{R}$; a classical example in the binary case is the binary cross-entropy

$$\mathcal{L}^{BCE}(p, h_\theta(x)) = p(1) \ln h_\theta(x) + (1 - p(1)) \ln(1 - h_\theta(x)).$$

However, in our setting, predictions are not pseudo-labels $\hat{y}$ nor probabilities over $y$, but a credal set $K$, making usual loss functions ill-defined. In such a case, a popular choice well-fitting the standard assumptions of semi-supervised learning (i.e., that data in $\mathcal{D}_U$ are randomly selected and have the same distribution as data in $\mathcal{D}_L$) is to consider an optimistic assumption (Destercke, 2022; Lienen and Hüllermeier, 2021) by considering the distribution within $K$ minimising the loss, i.e.,

$$\mathcal{L}_{min}(K, h_\theta(x)) = \min_{p \in K} \mathcal{L}(p, h_\theta(x)). \tag{1}$$

The final associated empirical risk of a model $h_\theta$ when observing $n$ data $(x_i, y_i)$ is then:

$$R_{emp}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{min}(h_\theta(x_i)). \tag{2}$$

Note that such a choice naturally extends loss functions used in the case of partial labels (corresponding to credal set including all probabilities with a given support), see e.g. (Cabannes et al., 2021; Liu and Dietterich, 2014). In the optimistic approach, we want the loss function to be minimized if the probability estimated by the model is inside the credal set, that is if $p(y) \in K$. Therefore, in this paper we consider the Kullback-Leibler divergence $D_{KL}$ (as in Lienen et al. (2022)) instead of the $BCE$ loss function, since this latter does not fulfill this desideratum (see A for more details).

As reminder, the Kullback-Leibler (KL) divergence is defined as the following:

$$D_{KL}(P \mid\mid Q) = \sum_{x \in \mathcal{X}} P(x) \ln(\frac{P(x)}{Q(x)}).$$

In a binary classification task, the KL divergence becomes

$$\mathcal{L}^{KL}(p, h_\theta(x)) := D_{KL}(p \mid\mid h_\theta(x)) = p(1) \ln(\frac{p(1)}{h_\theta(x)}) + (1 - p(1)) \ln(\frac{1 - p(1)}{1 - h_\theta(x)}),$$

and the corresponding $\mathcal{L}^{KL}_{min}(K, h_\theta(x))$ is (see Appendix A for more details):

$$\mathcal{L}^{KL}(K, h_\theta(x)) = \begin{cases} 0 & \text{if } h_\theta(x) \in [\underline{p}(1), \overline{p}(1)], \\ \mathcal{L}^{KL}(\underline{p}(1), h_\theta(x)) & \text{if } h_\theta(x) \leq \underline{p}(1), \\ \mathcal{L}^{KL}(\overline{p}(1), h_\theta(x)) & \text{if } h_\theta(x) \geq \overline{p}(1). \end{cases} \qquad (3)$$

A natural question is then how to obtain the interval $[\underline{p}(1), \overline{p}(1)]$ for a given observation. In the sequel, we propose to use Venn-Abers predictors as a convenient way to get calibrated intervals, starting with recalling the basic idea behind such predictors.

### 2.3. Venn-Abers predictors

In general, one cannot expect to have $h_\theta(x) = p(1|x)$, or even the less demanding property[2] $p(1|h_\theta(x)) = h_\theta(x)$. Satisfying such constraints amounts to require the predictor $h_\theta(x)$ to be well calibrated.

Venn predictors (Vovk and Petej, 2012) offer an easy post-hoc means to obtain estimators with calibration guarantees. In a nutshell, if $\mathcal{Y}$ is a $K$ element space, a Venn predictor outputs $K$ probability estimates $p_0, \ldots, p_K$, one of which is guaranteed to be calibrated. We are here interested in the binary case $\mathcal{Y} = \{0, 1\}$, for which we will use *Inductive Venn-Abers predictors* (IVAP) (Vovk et al., 2015; Nouretdinov et al., 2018; Peck et al., 2020). The idea is the following:

1. Divide the learning set $\mathcal{D}_L$ into a training set $\mathcal{D}_T$ of size $l$ and a calibration set $\mathcal{D}_C$ of size $k = n - l$

2. Train a classifier $h_\theta$ on $\mathcal{D}_T$, for example by solving $\theta = \arg\min_{\theta \in \Theta} R_{emp}(\theta)$.

3. Compute the scoring values $h_\theta(x)$ (e.g., the probabilities output by the classifier) for all instances in the *calibration set*, i.e., all $h_\theta(x)$ for $x \in \mathcal{D}_C$.

4. For any new test object or observation $x$:

   (a) compute its score $h_\theta(x)$ with the classifier,

[2] This is less demanding since different $x$'s will receive the same score.

(b) fit an *isotonic regression* model on $((h_\theta(x_1), y_1), ..., (h_\theta(x_k), y_k), (h_\theta(x), 0))$ as a function $g_0$ and another one on $((h_\theta(x_1), y_1), ..., (h_\theta(x_k), y_k), (h_\theta(x), 1))$ as a function $g_1$ (for the pairs $(x_i, y_i) \in \mathcal{D}_C$)

(c) consider the two obtained values $(g_0(h_\theta(x)), g_1(h_\theta(x)))$: one of them is a calibrated probability.

The algorithm pseudo-code for the IVAP is recalled in Appendix B. The key idea in this paper is to consider the Venn-Abers predictors as credal sets, by considering the probability intervals corresponding to the convex hull of the predictions output by such a predictor. This means that an observation $x$ would be associated with the interval $K_x = [\underline{p}_x(1), \overline{p}_x(1)]$ with $\underline{p}_x(1) = g_0(h_\theta(x))$ and $\overline{p}_x(1) = g_1(h_\theta(x))$.

In the next section, we will see how to combine credal learning and Venn-Abers predictors in the context of self-supervised learning, so as to create more calibrated credal sets and include the uncertainty on the pseudo-labels in the training process, thus improving the calibration of the prediction model.

## 3. Self Learning using Venn-Abers predictors

Now that we have presented Venn-Abers predictors in Section 2.3 and how learning from credal sets in Section 2.2, we address in this Section leveraging Venn-Abers predictors to obtain better-calibrated credal sets in a self learning paradigm.

We can already note that in the credal setting and using the loss (1), both precise and missing labels are specific cases of credal labels: as such, the earlier data sets $\mathcal{D}_L$ and $\mathcal{D}_U$ need not be distinguished from each other, and $\mathcal{D}_L \cup \mathcal{D}_U$ can just be seen as a specific kind of credal dataset.

Our proposal goes as follows: starting out with $\mathcal{D}_L$, we split it into the two sets $\mathcal{D}_T$ and $\mathcal{D}_C$ in order to apply the IVAP method later on. We then propose to learn a first classifier $h_{\theta^0}$ on the fully labeled set $\mathcal{D}_T$, through standard loss minimisation. We then apply IVAP to produce credal labels on the observations within $\mathcal{D}_U$, denoting by $K_x^0$ the credal set $[\underline{p}_x^0(1), \overline{p}_x^0(1)]$ obtained for observation $x$, thus obtaining a data set $\mathcal{D}_U^0$ with credal labels. We then use Equation (1) on $\mathcal{D}_T \cup \mathcal{D}_U^0$ to obtain a new model $h_{\theta^1}$, and so on. More generally, at a given iteration $j$ in the iterative self-learning process, we have

$$\hat{\theta}^j = \arg\min_{\theta \in \Theta} \sum_{x \in \mathcal{D}_L \cup \mathcal{D}_U} \mathcal{L}_{min}(K_x^{j-1}, h_{\hat{\theta}^{j-1}}(x)), \tag{4}$$

where $K_x^{j-1}$ is the credal set output when applying IVAP to $x$ using $\mathcal{D}_C$ and $h_{\hat{\theta}^{j-1}}$ as a model, and where for any $x \in \mathcal{D}_L$,

$$K_x = \begin{cases} [0.999, 0.999], & \text{if } y = 1, \\ [0.001, 0.001], & \text{if } y = 0, \end{cases}$$

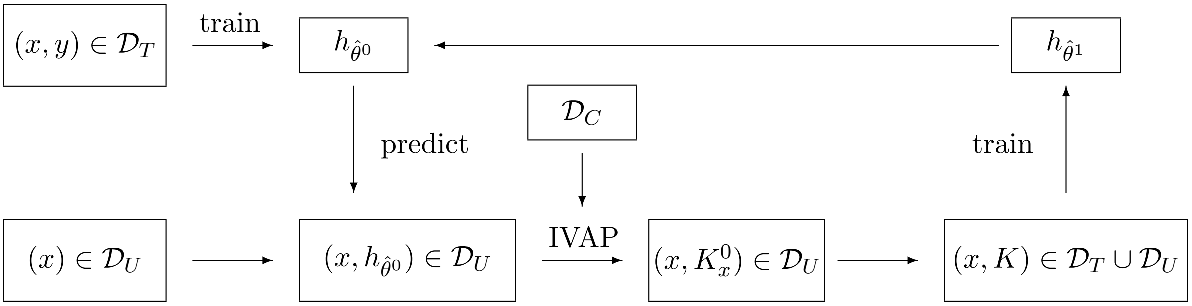

$(x, y) \in \mathcal{D}_T$
train
$h_{\hat{\theta}^0}$
$h_{\hat{\theta}^1}$
predict
$\mathcal{D}_C$
train
$(x) \in \mathcal{D}_U$
$(x, h_{\hat{\theta}^0}) \in \mathcal{D}_U$
IVAP
$(x, K_x^0) \in \mathcal{D}_U$
$(x, K) \in \mathcal{D}_T \cup \mathcal{D}_U$

Figure 2: Initializing loop of self-training using the Venn-Abers predictors algorithm

in order to avoid numerical issues due to $ln(0)$. This iterative procedure can then repeated at will, for example until performances on the test set begin to degrade. The first loop of this iterative procedure is illustrated in Figure 2. The whole iterative procedure is summarized in Algorithm 1.

It should be noted that we could start right away by using $\mathcal{D}_L \cup \mathcal{D}_U$, simply using vacuous or almost vacuous intervals for observations in $\mathcal{D}_U$. In this paper we choose to initialize by using only $\mathcal{D}_L$, for the simple reason that it makes the comparison with standard self-learning approaches easier.

**Algorithm 1** Self-learning using Venn-Abers predictors

```
Require: a labeled set $\mathcal{D}_T$, a calibration set $\mathcal{D}_C$, a unlabeled data set $\mathcal{D}_U$
Require: a hypothesis space $\Theta$ used to learn a real-valued model $h_\theta : \mathcal{X} \to \mathcal{Y}$
Require: a number $e$ of iteration (or a stopping criteria)
  $i \longleftarrow 0$
  train $h_{\theta^0}$ using $\mathcal{D}_T$
  label observation $x \in \mathcal{D}_U$ by $K_x$ generated by IVAP on $h_{\theta^0}(x)$ using $\mathcal{D}_C$
  while $i \leq e$ do
    $i \longleftarrow i + 1$
    train $h_{\theta^i}$ using $\mathcal{D}_T \cup \mathcal{D}_U$
    label observation $x \in \mathcal{D}_U$ by $K_x$ generated by IVAP on $h_{\theta^i}(x)$ using $\mathcal{D}_C$
  end while
  return $h_{\theta^e}$
```

## 4. Experiments

This section reports experimental results, showing in particular that using Venn-Abers predictors combined with credal learning in self-learning gives in general faster convergence and overall better results, as well as more robustness in those cases where classical self-learning performs badly.

### 4.1. Illustration on synthetic data

Before testing our approach on real-world data sets, we provide a small illustration of how our method behaves on a synthetic data set. This data set with $X \in \mathbb{R}^2$ is composed of two Gaussian conditional distributions

$$p(x|1) \sim \mathcal{N}(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \frac{1}{2}\begin{pmatrix} 11 & 9 \\ 9 & 11 \end{pmatrix}), \quad p(x|0) \sim \mathcal{N}(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \frac{1}{2}\begin{pmatrix} 11 & -9 \\ -9 & 11 \end{pmatrix}).$$

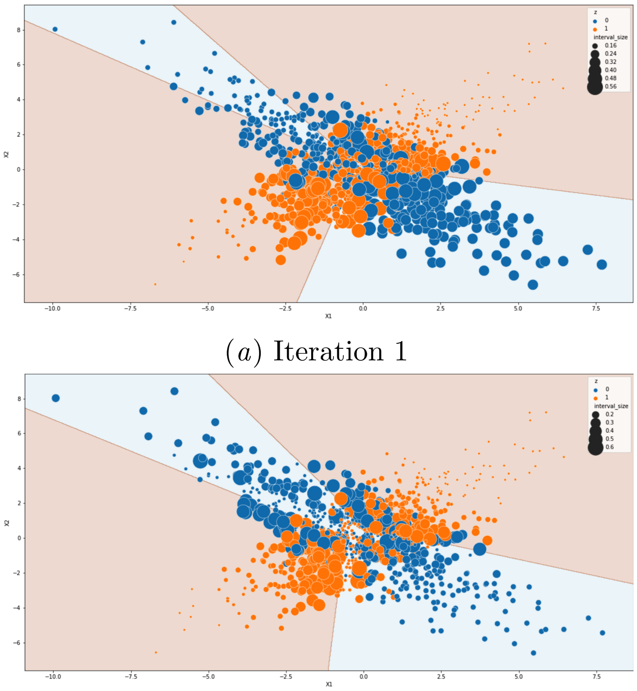
We set $n = 1000$, and split $\mathcal{D}$ into $\mathcal{D}_T$ of size 80, $\mathcal{D}_C$ of size 20 and $\mathcal{D}_U$ of size 900. We trained a neural network with a hidden layer of 3 neurons with learning rate $\lambda = 0.2$ on $\mathcal{D}_T$, and then applied our method over 10 iterations. We chose a neural network since it is often poorly calibrated (Johansson and Gabrielsson (2019)). The evolution of the decision boundary, as well as the size of the credal sets output by our method, are shown in Figure 3.

We can observe that both the decision boundaries as well as our certainty or confidence in the predictions made evolve over time, even if the set $\mathcal{D}_C$ remains the same. For example, we can see that the decision region for the negative class has a tendency to shrink between the 1st and 10th iteration, and that the average interval size associated to test data points belonging to the negative class tends to diminish in the lower right quadrant, while increasing in the upper left quadrant. Similarly, the interval size associated to the data points in the dense overlapping region around point $(0, 0)$ tends to decrease with the number of iterations, showing that we are more and more certain of our estimates in these regions.

Figure 4 displays the probabilities (of the positive class) output by the neural network (without post-hoc calibration) at the end of the learning, as well as the size of the credal sets. Those probabilities are reasonably accurate, and we can observe that there is not necessarily a strong link between the value of this probability and the reliability of the estimate, which we evaluate by the size of the output intervals: we can observe rather extreme probabilistic estimates (i.e., far from 0.5) associated to big or small intervals, as well as ambiguous probabilistic estimates (i.e., close to 0.5) also associated to big or small intervals.

### 4.2. Real data

To test our method on real data, we used six datasets: Breastcancer, Digits, Australian, Banknote, Heart disease and the Adult datasets (courtesy of the UCI repository (Dua and Graff, 2017)). As we consider only binary classification, for the



$(a)$ Iteration 1

$(b)$ Iteration 10

Figure 3: Evolution of the decision boundary and the interval sizes across 10 self-learning iterations using Venn-Abers predictors (synthetic data)

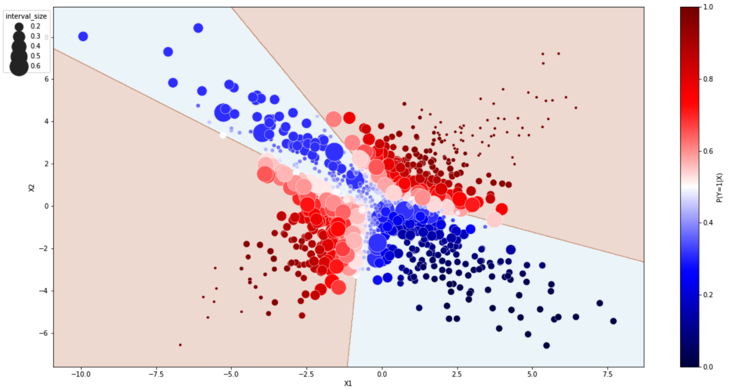


Figure 4: Probabilities output by the classifier with interval size output by *Venn-Abers predictors* at iteration 10

Table 1: Architecture and hyperparameter

| Dataset | number of neurons (hidden layer) | $\lambda$ |
|---|---|---|
| Breastcancer | 5 | 0.01 |
| Digits | 10 | 0.01 |
| Australian | 4 | 0.005 |
| Banknote | 2 | 0.01 |
| Heart disease | 5 | 0.005 |
| Adult | 10 | 0.001 |

Digits dataset, we grouped even numbers as class 0 and odd numbers as class 1. For the Adult dataset, we randomly selected 5000 instances over the entire dataset, respecting the proportions of the classes so as to obtain an imbalanced test set against which our model's performance can be assessed. We used as classifier a neural network with 1 hidden layer (the architecture and learning rate $\lambda$ for each dataset is presented on table 1). The optimizer was an SGD, with no momentum nor weight decay. Batch size was set to 10. We use a neural network for the same reason as we did for the synthetic data (poor native calibration). We split each dataset into four new sets: $\mathcal{D}_T$, $\mathcal{D}_U$, $\mathcal{D}_C$ and $\mathcal{D}_t$ (the test set). We compared three different self-learning strategies:

1. a standard, classical self-learning (SL) procedure consisting of adding a batch of new labeled data at each iteration (the batch of data for which the prediction probabilities are the furthest away from 0.5, that is for which $|0.5 - h_{\theta^i}|$ is maximal). The size of the batch is set to 2% of the initial data set;

2. self-learning using soft labels (SLSL): at each iteration, we label $\mathcal{D}_U$ with $h_{\theta^i}(x)$, adding this pseudo-labeled data to $\mathcal{D}_U$ before training the classifier on $\mathcal{D}_L \cup \mathcal{D}_U$;

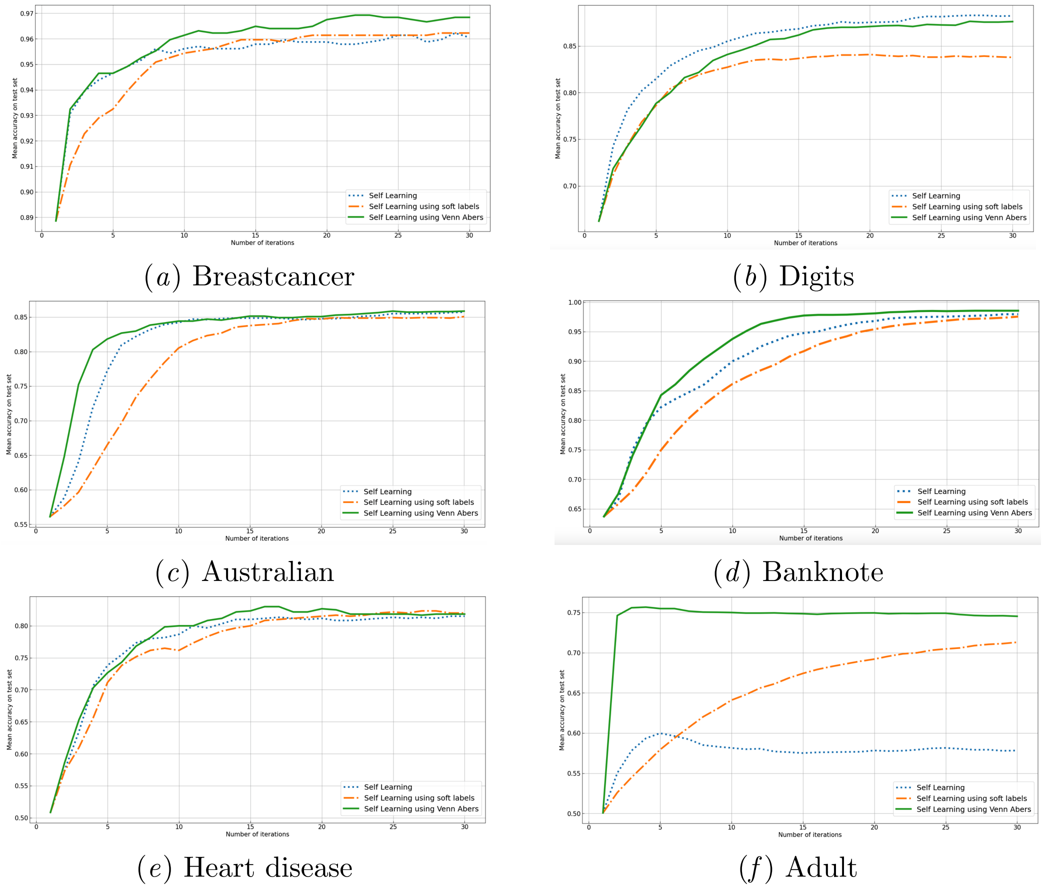
3. self-learning using Venn-Abers predictors (SLVA), our proposal described in Section 3 and Algorithm 1.

The splitting is done 10 times on different seeds: and for each split, we apply the three strategies on 30 iterations. For each split, 20% of the data was kept for $\mathcal{D}_t$ as test, and of the 80% remaining, 80 were kept as labeled data for $\mathcal{D}_T$, 5 or 10% were kept[3] for $\mathcal{D}_C$ to ensure that $|\mathcal{D}_C| \geq 20$, and the rest was put in $\mathcal{D}_U$. For each strategy, we measured the average accuracy $\overline{a}$ of the 30 iterations over the 10 different splits, the mean accuracy $a_{30}$ at iteration 30 and the standard deviation of the accuracy $\sigma(a_{30})$ at iteration 30. The results are given in Table 2 and Figure 5.

We can make quite several observations about these results:

- SLVA usually performs better than the other approaches. This is certainly true in average, as shown by the three first columns in Table 2, where our

[3] i.e., either 4% or 8% of the initial data set



$(a)$ Breastcancer

$(b)$ Digits

$(c)$ Australian

$(d)$ Banknote

$(e)$ Heart disease

$(f)$ Adult

Figure 5: Accuracies on $\mathcal{D}_t$ for the 6 datasets along the training of 30 iterations over 10 different seeds.

Table 2: Performances on $\mathcal{D}_t$ for the 6 datasets over 10 different seeds

| | $\overline{a}$ | | | $a_{30}$ | | | $\sigma(a_{30})$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | SL | SLSL | SLVA | SL | SLSL | SLVA | SL | SLSL | SLVA |
| Breastcancer | 0.953 | 0.951 | **0.959** | 0.961 | 0.962 | **0.968** | 0.019 | **0.009** | 0.015 |
| Digits | **0.851** | 0.817 | 0.838 | **0.882** | 0.838 | 0.876 | 0.032 | **0.018** | 0.025 |
| Australian | 0.815 | 0.789 | **0.827** | 0.857 | 0.851 | **0.859** | 0.026 | 0.023 | **0.014** |
| Banknote | 0.907 | 0.881 | **0.926** | 0.980 | 0.976 | **0.986** | **0.007** | 0.013 | 0.008 |
| Heart disease | 0.775 | 0.768 | **0.782** | 0.815 | **0.820** | 0.818 | 0.039 | 0.036 | **0.035** |
| Adult | 0.578 | 0.653 | **0.741** | 0.578 | 0.713 | **0.745** | 0.018 | 0.029 | **0.017** |

approach outperforms the baselines 5 times out of 6; but also at the end of the learning process ($a_{30}$ in Table 2), where our approach outperforms the baselines 4 times out of 6, and remains close to the best method in the two remaining ones.

- SLVA is also more robust accross all the used data sets, as it is either the best performing method or remains close to it, while $SL$ and $SLSL$ display greater variability (e.g., $SLSL$ does not perform well on Digits, $SL$ is very bad on Adult and both under-perform on Breastcancer).

- SLVA also converges faster to asymptotic performances, as indicated by the fact that the slope of the green curve is usually steeper in Figure 5, and that $\overline{a}$ (the average accuracy over iterations) is higher for SLVA.

- SLVA is the only method that has no problem with dealing with the Adult data set: for this latter one, the convergence of $SLSL$ is very slow, and the performances of $SL$ are extremely bad, and even quickly degrade after the fifth iteration.

In terms of variance, all methods seem to be on par, with no method really showing an advantage over the others. However, the remarks above indicate that using calibrated credal sets to replace unlabeled data can be considered as a serious alternative to standard self-learning approaches, as it usually exhibits better performances without adding any computational cost when dealing with binary classification problems using the $D_{KL}$ loss function.

## 5. Conclusion

In this paper, we have introduced a new approach of self learning, taking into account the uncertainty of new labeled data added to the training set. It concerns binary classification task by loss optimization. If we have little labeled data and many unlabeled data, using the approach presented here can help making the best use of the latter to train a classifier while being robust to pseudo-labels.

Experiments show that our approach compares favorably with classical self learning approaches, as it either gives better or comparable performances. In particular, it seems that when data are imbalanced, using this approach can improve robustness and avoid slow convergence or even failure of self-learning approaches using (possibly uncalibrated) soft labels or hard labels. In addition, it is likely that accounting for the uncertainties (both epistemic and aleatoric) in the self-learning process will increase the acceptability of the approaches ny users.

As a first follow-up to the present work, we would like to perform further experiments to confirm some of our observations as well as testing the limit of the presented framework. Of particular interest would be to confirm the good behavior of our approach in the case of imbalanced data, but also to see how the method performs when the size of the calibration set evolves. Related to this last item is also the idea to let the calibration set vary in composition and possibly in size over time. Indeed, at each iteration, one could think of re-sampling the calibration set within $\mathcal{D}_L$, rather than keeping the same one. We could also think about integrating an increasing amount of data from $\mathcal{D}_L$ into $\mathcal{D}_C$ as the credal labels within $\mathcal{D}_U$ become more precise.

Finally, an obvious extension is to take this idea of self-supervised learning to more complex learning problems. A first candidate is to go towards multiclass problems, in which case Venn predictors (Vovk and Petej, 2012) outputs $|\mathcal{Y}|$ probabilities, and to recast our learning optimization problem within such a setting. An open question is how to deal with the increasing computational time in this setting. There is an Inductive version of Venn Predictors Lambrou et al. (2015) , that has complexity $O(n^2 * |\mathcal{Y}|)$ which seems optimal for this case. Another promising direction for credal self-supervised learning using calibrated outputs is gradual domain adaptation, in which self-learning approaches are used to solved a transfer learning problem (Zhang et al., 2021; Kumar et al., 2020; Wang et al., 2022).

## References

Vivien A Cabannes, Francis Bach, and Alessandro Rudi. Disambiguation of weak supervision leading to exponential convergence rates. In *International Conference on Machine Learning*, pages 1147–1157. PMLR, 2021.

Sébastien Destercke. Uncertain data in learning: challenges and opportunities. *Conformal and Probabilistic Prediction with Applications*, pages 322–332, 2022.

Sébastien Destercke and Didier Dubois. Special cases. *Introduction to Imprecise Probabilities*, pages 79–92, 2014.

Pedro Domingos and Michael Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classi er. In *Proc. 13th Intl. Conf. Machine Learning*, pages 105–112, 1996.

Inmaculada Dópido, Jun Li, Prashanth Reddy Marpu, Antonio Plaza, José M Bioucas Dias, and Jon Atli Benediktsson. Semisupervised self-learning for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, 51 (7):4032–4044, 2013.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Didier Dubois and Henri Prade. When upper probabilities are possibility measures. *Fuzzy sets and systems*, 49(1):65–74, 1992.

Ulf Johansson and Patrick Gabrielsson. Are traditional neural networks well-calibrated? In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019. doi: 10.1109/IJCNN.2019.8851962.

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020.

Antonis Lambrou, Ilia Nouretdinov, and Harris Papadopoulos. Inductive venn prediction. *Annals of Mathematics and Artificial Intelligence*, 74:181–201, 2015.

Christian Leistner, Amir Saffari, Jakob Santner, and Horst Bischof. Semi-supervised random forests. In *2009 IEEE 12th international conference on computer vision*, pages 506–513. IEEE, 2009.

Julian Lienen and Eyke Hüllermeier. Credal self-supervised learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS, December 6-14, 2021, virtual*, 2021.

Julian Lienen, Caglar Demir, and Eyke Hüllermeier. Conformal credal self-supervised learning, 2022. URL https://arxiv.org/abs/2205.15239.

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

Ilia Nouretdinov, Denis Volkhonskiy, Pitt Lim, Paolo Toccaceli, and Alexander Gammerman. Inductive Venn-Abers predictive distribution. In Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Ralf Peeters, editors, *Proceedings of the Seventh Workshop on Conformal and Probabilistic Prediction and Applications*, volume 91 of *Proceedings of Machine Learning Research*, pages 15–36. PMLR, 11–13 Jun 2018. URL https://proceedings.mlr.press/v91/nouretdinov18a.html.

Jonathan Peck, Bart Goossens, and Yvan Saeys. Detecting adversarial manipulation using inductive venn-abers predictors. *Neurocomputing*, 416:202–217, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.11.113. URL https://www.sciencedirect.com/science/article/pii/S0925231220305087.

Andra Petrovai and Sergiu Nedevschi. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1578–1588, 2022.

Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine learning*, 52:199–215, 2003.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, Peter Flach, et al. Classifier calibration: How to assess and improve predicted class probabilities: a survey. *arXiv preprint arXiv:2112.10327*, 2021.

Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42:245–284, 2015.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

Vladimir Vovk and Ivan Petej. Venn-abers predictors, 2012. URL https://arxiv.org/abs/1211.0025.

Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/a9a1d5317a33ae8cef33961c34144f84-Paper.pdf.

Haoxiang Wang, Bo Li, and Han Zhao. Understanding gradual domain adaptation: Improved analysis, optimal path and beyond. In *International Conference on Machine Learning*, pages 22784–22801. PMLR, 2022.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Gradual domain adaptation via self-training of auxiliary models. *arXiv preprint arXiv:2106.09890*, 2021.

## Appendix A. Choice of Kullback-Leibler divergence as loss function

In this appendix, we explain why we choose $D_{KL}(P \mid\mid h_\theta(x))$ over binary cross entropy when considering the credal loss $\mathcal{L}_{min}$.

As explained, $\mathcal{L}_{min}$ is the minimum of the loss function $\mathcal{L}$ taken over the credal label $K$ and the estimated probability $h_\theta(x)$. We want this $\mathcal{L}_{min}$ having the following properties:

1. $\mathcal{L}_{min}(K_x, h_{\theta(x)}) = 0$ if $h_\theta(x) \in K$,
2. $\mathcal{L}_{min}(K_x, h_{\theta(x)}) = \mathcal{L}(\underline{p}(1), h_\theta(x))$ if $h_\theta(x) \leq \underline{p}(1)$,
3. $\mathcal{L}_{min}(K_x, h_{\theta(x)}) = \mathcal{L}(\overline{p}(1), h_\theta(x))$ if $h_\theta(x) \geq \overline{p}(1)$.

Indeed, if our prediction $h_\theta(x)$ is inside the interval-valued credal label, we would like to not penalize it.

However, such properties are not satisfied by usual binary cross entropy $BCE$, the main reason being that given a fixed $h_\theta(x)$, $BCE$ is linear in $p$ and its partial derivative according to $p$ is a constant:

$$\frac{\partial BCE}{\partial p} = \ln(h_\theta(x)) - \ln(1 - h_\theta(x)).$$

This means that the minimum of the binary cross entropy $\mathcal{L}_{min}(K_x, h_{\theta(x)})$ is one of its bounds $\underline{p}(1)$ or $\overline{p}(1)$, no matter if $h_\theta(x) \in K$ or not. Hence, this function cannot serve our purpose.

In contrast, once we fix $h_\theta(x)$, the KL divergence $D_{KL}$ is not linear in $p$ and its partial derivative according to $p$ is given by:

$$\begin{aligned}\frac{\partial D_{KL}}{\partial p} &= \ln(\frac{p(1)}{h_\theta(x)}) - \ln(\frac{1 - p(1)}{1 - h_\theta(x)}),\\ \frac{\partial^2 D_{KL}}{\partial p^2} &= \frac{1}{p(1)} + \frac{1}{1 - p(1)} \geq 0.\end{aligned}$$

Hence, $D_{KL}$ is convex in $p(1)$ and its minimum is reached for:

$$\frac{\partial D_{KL}}{\partial p} = 0 \Leftrightarrow p(1) = h_\theta(x).$$

Since $D_{KL}$ is convex, if $\overline{p}(1) < h_\theta(x)$, the minimum is reached for $p(1) = \overline{p}(1)$ (and conversely for $p(1) = \underline{p}(1)$ if $\underline{p}(1) > h_\theta(x)$). Thus, $D_{KL}$ satisfies the three properties mentioned above, explaining why it is preferable to $BCE$.

## Appendix B. Inductive Venn-Abers predictors

Algorithm 2 refers to IVAP described in Section 2.3.

**Algorithm 2** Inductive Venn-Abers predictors

```
Require: an underlaying scoring function $h_\theta$ trained on the training_set
    $\{(x_1, y_1), ..., (x_l, y_l)\}$,
Require: calibration_set $\{(x_{l+1}, y_{l+1}), ..., (x_m, y_m)\}$,
Require: $x \in$ test_set
  for $y \in \{0, 1\}$ do
    compute scoring function $(h_\theta(x_{l+1}), ..., h_\theta(x_m), h_\theta(x))$
    compute isotonic calibrator $(g(h_\theta(x_{l+1}), y_{l+1}), ..., g(h_\theta(x_m), y_m), g(h_\theta(x), y))$
    set $p_y = g(h_\theta(x, y))$
  end for
  return $(p_0, p_1)$
```