Auto-apprentissage à l'aide de prédicteurs de Venn-Abers

Côme Rodriguez, Vitor Martin Bordini, Sébastien Destercke, Benjamin Quost

Auto-apprentissage à l'aide de prédicteurs de Venn-Abers

Côme Rodriguez[1] Vitor Martin Bordini[1,2] Sébastien Destercke[2,3] Benjamin Quost[1,2]

[1] Université de Technologie de Compiègne, France
[2] Laboratoire Heudiasyc, UMR UTC-CNRS 7253
[3] Centre National de la Recherche Scientifique

come.rodrig@gmail.com
vitor.martin-bordini@hds.utc.fr
sebastien.destercke@hds.utc.fr
benjamin.quost@hds.utc.fr

3 janvier 2024

Résumé :

Dans les problèmes d'apprentissage supervisé, il est courant de disposer d'un grand nombre de données non étiquetées, mais de peu de données étiquetées. Il est alors souhaitable d'exploiter les données non étiquetées pour améliorer la procédure d'apprentissage. L'un des moyens pour y parvenir consiste à demander à un modèle de prédire des « pseudo-étiquettes » pour les données non étiquetées, afin de les utiliser pour l'apprentissage. Dans le cadre de l'auto-apprentissage, les pseudo-étiquettes sont fournies par le même modèle que celui qui les exploite. Comme ces pseudo-étiquettes sont par nature incertaines et seulement partiellement fiables, il est naturel de tenir compte de l'incertitude d'étiquetage dans le processus d'apprentissage, ne serait-ce que pour renforcer la procédure d'auto-apprentissage. Cet article décrit une telle approche, dans laquelle nous utilisons des prédicteurs Venn-Abers pour produire des étiquettes crédales calibrées afin de quantifier l'incertitude d'étiquetage. Ces étiquettes sont ensuite intégrées au processus d'apprentissage au moyen d'une fonction de cout adaptée. Les expériences montrent que la prise en compte de l'incertitude d'étiquetage renforce la procédure d'auto-apprentissage et lui permet généralement de converger plus rapidement.

Mots-clés :

Auto-apprentissage, prédicteurs de Venn-Abers, étiquettes crédales

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

L'utilisation des données et de l'apprentissage automatique devient de plus en plus fréquente, en partie parce que les utilisateurs génèrent de plus en plus de données. Cependant, une grande partie de ces données n'est pas étiquetée. Dans l'apprentissage supervisé classique, ces dernières ne peuvent pas être utilisées pour former le modèle. Ceci est particulièrement problématique dans les situations où la quantité de données étiquetées est faible, ce qui se produit généralement lorsque l'obtention d'une expertise est coûteuse.

Les techniques d'apprentissage semi-supervisé répondent à cette situation particulière en proposant des méthodes qui apprennent à partir de données étiquetées et non étiquetées. Ces techniques sont nombreuses et nous renvoyons par exemple à [22] pour une étude et une taxonomie récentes. L'auto-apprentissage est une approche spécifique de l'apprentissage semi-supervisé qui consiste à remplacer les étiquettes manquantes par des prédictions du modèle, puis

à incorporer ces données dans l'ensemble d'apprentissage. Ces étiquettes prédites sont souvent appelées pseudo-étiquettes. Bien que l'idée de l'auto-apprentissage et de l'étiquetage automatique ne soit pas nouvelle [27] et qu'elle soit appliquée avec succès depuis un certain temps dans différents domaines tels que le traitement d'images [5], elle a récemment connu un regain d'intérêt [19, 17]. Cependant, le remplacement des étiquettes par des prédictions erronées peut conduire à des performances moindres, qui peuvent même être aggravées de manière significative dans certains cas.

Une solution à ce problème consiste à remplacer les étiquettes inconnues par des étiquettes incertaines, généralement probabilistes [11], et à entraîner le modèle en utilisant une fonction de perte adéquate (par exemple, l'entropie croisée). Cependant, de telles estimations probabilistes ne sont pas garanties comme étant précises et fiables, par exemple lorsque les méthodes d'apprentissage reposent sur des hypothèses trop fortes [4] ou lorsqu'elles affichent une variance trop élevée [18]. Un moyen classique d'obtenir des estimations probabilistes plus fiables consiste à *calibrer* ces probabilités [20]. Toutefois, ces probabilités calibrées peuvent ne pas être en mesure de refléter le degré de fiabilité des estimations, en ce sens qu'elles ne quantifieront pas correctement leur *incertitude épistémique*, par exemple si elles s'appuient ou non sur un grand nombre de données. Afin d'augmenter l'expressivité des pseudo-étiquettes fournies, [13] ont récemment proposé de considérer des ensembles convexes de probabilités comme pseudo-étiquettes et, plus récemment, d'utiliser la prédiction conformelle (voir [12]) afin d'obtenir de tels ensembles convexes.

Si les ensembles de probabilités considérés par [13] ont l'avantage d'être simples et de ne pas augmenter de manière significative la complexité calculatoire, ils présentent l'inconvénient de ne pas pouvoir modéliser des probabilités précises, car ils contiendront toujours au moins une distribution de probabilité dégénérée (qui place toute la masse de probabilité sur une des classes). [1]. Dans cet article, nous considérons la même idée d'utiliser des ensembles de probabilités comme pseudo-étiquettes, mais plutôt que la prédiction conformelle, nous utilisons des prédicteurs de Venn [10], et plus précisément des prédicteurs de Venn-Abers [23]car nous nous concentrons sur le cas binaire.

Le document est organisé comme suit. La section 2 rappelle les notions d'apprentissage, d'étiquettes crédales et de prédicteurs de Venn-Abers. La section 3 explique ensuite comment apprendre à partir d'étiquettes crédales issues de prédicteurs de Venn-Abers en adaptant la divergence de Kullback-Leibler à ce contexte, et présente notre schéma d'auto-apprentissage proposé. La section 4 fournit des résultats expérimentaux sur différents ensembles de données, montrant que l'utilisation d'étiquettes crédales calibrées améliore généralement la procédure d'auto-apprentissage et peut prévenir une dégradation potentielle du modèle.

2 Préliminaires

Nous introduisons les éléments nécessaires à la présentation de notre approche.

2.1 Apprentissage semi-supervisé, auto-apprentissage

L'approche semi-supervisée classique considère à la fois un ensemble de données étiquetées

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

et un ensemble de données non étiquetées

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

supposés provenir de la même distribution sous-jacente, où $\mathcal{X}, \mathcal{Y}$ sont les espaces d'entrée et de sortie (discrète catégorique). Dans le présent document, nous considérerons la classification binaire, c'est-à-dire $\mathcal{Y} = \{0, 1\}$.

1. La raison en est qu'il s'agit de distributions de possibilités, voir [7].

L'objectif des méthodes d'apprentissage semi-supervisé, et des méthodes d'apprentissage en général, est d'apprendre une fonction de score à valeurs réelles $h_\theta : \mathcal{X} \to \mathbb{R}$ à partir des données $\mathcal{D}_L \cup \mathcal{D}_U$ disponibles. Le modèle h_θ peut alors être utilisé pour faire une prédiction $\hat{y}$ pour n'importe quel observation x en utilisant un seuil c, c'est-à-dire $\hat{y} = 1$ si $h_\theta(x) > c$, et $\hat{y} = 0$ sinon. Des exemples typiques sont les SVM avec $h_\theta(x) \in (-\infty, \infty)$ et $c = 0$, ou la régression logistique avec $h_\theta(x) \in [0, 1]$ et $c = 0.5$. Notons qu'une transformation sigmoïde permet de retrouver le second cas à partir du premier. Dans cet article, et pour faciliter la lecture, nous supposerons que $h_\theta \in [0, 1]$, et peut être interprété comme une probabilité (non calibrée), avec $h_\theta(x) = \hat{p}(y = 1|x)$.

L'apprentissage auto-supervisé [21] consiste à (1) apprendre un modèle h_{θ^0} à partir des données $\mathcal{D}_L$, (2) sélectionner quelques données non étiquetées $x \in \mathcal{D}_U$ et les compléter par une prédiction $h_{\theta^0}(x)$, avant que (3) $(x, h_{\theta^0}(x))$ ne soit ajouté à $\mathcal{D}_L$: la procédure peut alors être répétée de manière itérative, en apprenant un nouveau modèle h_{θ^1}, et ainsi de suite, jusqu'à ce qu'une condition d'arrêt soit remplie. Bien qu'une telle approche puisse tirer parti d'un modèle précis h_{θ^j}, elle peut également souffrir lorsque des prédictions erronées (inexactes ou non fiables) sont incorporées à l'ensemble de données. Pour contourner ce problème, nous proposons une approche où tous les exemples de $\mathcal{D}_U$ sont étiquetés en une seule fois, mais par des ensembles convexes de probabilités — qui se réduisent à des intervalles dans la classification binaire — plutôt que par des étiquettes précises ou probabilistes. De tels ensembles convexes constituent un moyen générique, riche et flexible de représenter les prédictions du modèle avec leur incertitude associée, évitant ainsi la nécessité de sélectionner les données pseudo-étiquetées à ajouter à $\mathcal{D}_L$, et permettant de différencier les prédictions fiables des autres.

2.2 Ensemble crédal, apprentissage crédal

Un ensemble crédal [3] est un ensemble convexe fermé de distributions de probabilités. Nous désignerons par $\Delta_\mathcal{Y}$ l'ensemble des distributions de probabilité sur $\mathcal{Y}$, et par $K \subseteq \Delta_\mathcal{Y}$ un ensemble crédal défini sur $\mathcal{Y}$. L'avantage des ensembles crédaux est qu'ils peuvent modéliser toutes types de connaissance sur une étiquette : étiquettes manquantes ou partielles (sous-ensemble de classes), étiquettes probabilistes, étiquettes précises. Dans le cas binaire, tout ensemble crédal peut être résumé par des bornes $\underline{p}$ et $\overline{p}$ sur la probabilité $p := \mathbb{P}(Y = 1)$ d'intérêt : $\underline{p} \leq p \leq \overline{p}$. L'intervalle $[\underline{p}, \overline{p}]$ encode toute l'information disponible, une étiquette probabiliste étant modélisée par $\underline{p} = \overline{p} = p$, et une étiquette manquante (ou totalement non fiable) par $[\underline{p}, \overline{p}] = [0, 1]$. La figure 1 illustre différentes situations, avec leurs intervalles de probabilité associés.

La question est alors de savoir comment apprendre à partir de telles étiquettes crédales. En classification binaire supervisée « classique », nous apprenons un classificateur h_θ à partir de données de type (x, y) en optimisant une fonction de coût $\mathcal{L}$ basée sur les étiquettes précises y et les probabilités de sortie h_θ. Nous considérons ici une fonction de coût $\mathcal{L} : \Delta_\mathcal{Y} \times \Delta_\mathcal{Y} \to \mathbb{R}$ définie sur l'espace des probabilités ; un exemple classique dans le cas binaire est l'entropie croisée binaire :

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

Cependant, dans notre travail, une prédiction n'est pas une pseudo-étiquette $\hat{y}$ ou une probabilité sur y, mais un ensemble crédal K de distributions de probabilité, ce qui rend les fonctions de coût habituelles inutilisables. Dans ce cas, une approche populaire, « optimiste »[2, 13], consiste à remplacer l'étiquette manquante par la distribution de K minimisant la perte :

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

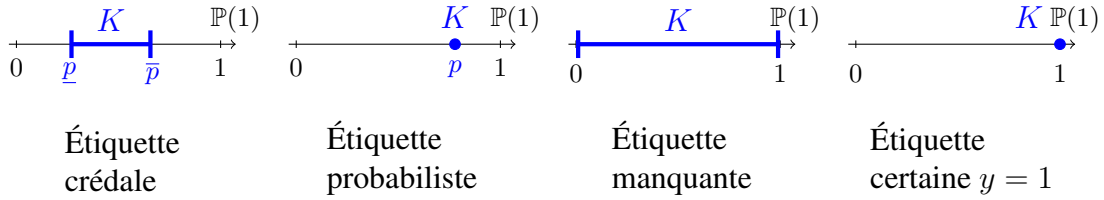


FIGURE 1 – Quelques exemples d'étiquettes crédales

Cette substitution correspond aux hypothèses standard de l'apprentissage semi-supervisé (les données de $\mathcal{D}_U$ sont supposées choisies au hasard et de même distribution que celles de $\mathcal{D}_L$). Le risque empirique associé à un modèle h_θ lors de l'observation de n données (x_i, y_i) est alors

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

Cette approche étend naturellement les fonctions de perte utilisées dans le cas d'étiquettes partielles (l'ensemble crédal comprenant alors toutes les probabilités avec un support donné), voir par exemple [1, 14]. Dans une perspective optimiste, nous voulons que la fonction de perte soit minimisée si la probabilité estimée par le modèle se trouve à l'intérieur de l'ensemble crédal : $p(y) \in K$. Par conséquent, dans cet article, nous considérons la divergence de Kullback-Leibler D_{KL} (comme dans [12]), plutôt que $\mathcal{L}^{BCE}$ qui ne remplit pas ce critère (voir l'annexe A pour plus de détails). Rappelons que

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

Dans une tâche de classification binaire, cela permet de définir la fonction de coût

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

et la fonction de coût optimiste $\mathcal{L}^{KL}_{min}(K, h_\theta(x))$ correspondante est (voir l'annexe A)

$$\mathcal{L}^{KL}_{min}(K, h_\theta(x)) = \begin{cases} 0 & \underline{p} \le h_\theta(x) \le \overline{p}, \\ \mathcal{L}^{KL}(\underline{p}, h_\theta(x)) & h_\theta(x) \le \underline{p}, \\ \mathcal{L}^{KL}(\overline{p}, h_\theta(x)) & h_\theta(x) \ge \overline{p}. \end{cases}$$

Une question naturelle est alors de savoir comment obtenir l'intervalle $[\underline{p}, \overline{p}]$ pour une observation donnée. Dans la suite, nous proposons d'utiliser les prédicteurs de Venn-Abers pour obtenir des intervalles calibrés.

2.3 Prédicteurs Venn-Abers

En général, le modèle $h_\theta(x)$ ne satisfait pas $h_\theta(x) = \mathbb{P}(1|x)$, ou même la propriété plus faible [2] $h_\theta(x) = \mathbb{P}(1|h_\theta(x))$, ce qui revient à exiger que le prédicteur $h_\theta(x)$ soit bien calibré.

Les prédicteurs de Venn [24] offrent un moyen facile d'obtenir des estimateurs avec des garanties de calibration. En résumé, si $\mathcal{Y}$ est un espace à K éléments, un prédicteur de Venn produit K estimations de probabilité $p_0, \ldots, p_K$, dont l'une est garantie comme étant calibrée. Nous nous intéressons ici au cas binaire, c'est-à-dire $\mathcal{Y} = \{0, 1\}$, pour lequel nous utiliserons des *prédicteurs de Venn-Abers inductifs* (IVAP) [25, 15, 16]. Le principe est le suivant :

1. Diviser l'ensemble d'apprentissage $\mathcal{D}_L$ en un ensemble d'apprentissage $\mathcal{D}_T$ de taille l et un ensemble de calibration $\mathcal{D}_C$ de taille $k = n - l$.
2. Entraîner un classifieur h_θ sur $\mathcal{D}_T$, par exemple en résolvant $h_\theta = \arg\min_{\theta\in\Theta} R_{emp}(\theta)$.

2. car différents x peuvent recevoir le même score

3. Pour tous les exemples $x \in \mathcal{D}_C$, calculer les scores $h_\theta(x)$ (par exemple, les probabilités données par le classifieur).
4. Pour tout nouvel exemple x (de test) :
   (a) calculer son score $h_\theta(x)$ avec le classifieur ;
   (b) apprendre deux modèles de *régression isotonique* des étiquettes à partir des scores : g_0 (respectivement, g_1) à partir des couples $\{(h_\theta(x_i), y_i)_{i=1}^k, (h_\theta(x), 0)\}$ (resp., $\{(h_\theta(x_i), y_i)_{i=1}^k, (h_\theta(x), 1)\}$), avec $(x_i, y_i) \in \mathcal{D}_C$;
   (c) considérons les deux valeurs obtenues $g_0(h_\theta(x))$ et $g_1(h_\theta(x))$: l'une d'entre elles est une probabilité calibrée.

Le pseudo-code de l'algorithme pour l'IVAP est rappelé en annexe B. L'idée clé de notre travail est d'utiliser les prédicteurs de Venn-Abers pour obtenir les ensembles crédaux, en considérant les intervalles de probabilité correspondant à l'enveloppe convexe des prédictions données par les prédicteurs. Cela signifie qu'une observation x serait associée à l'intervalle $K_x = [\underline{p}_x, \overline{p}_x]$ avec $\underline{p}_x = g_0(h_\theta(x))$ et $\overline{p}_x = g_1(h_\theta(x))$.

Dans la section suivante, nous verrons comment combiner l'apprentissage crédal et les Venn-Abers dans le contexte de l'apprentissage auto-supervisé, afin de créer des ensembles crédaux mieux calibrés et d'inclure l'incertitude sur les pseudo-étiquettes dans le processus d'apprentissage, améliorant ainsi la calibration du modèle de prédiction.

3 Auto-apprentissage à l'aide de prédicteurs de Venn-Abers

Nous abordons dans cette section l'exploitation des prédicteurs de Venn-Abers pour obtenir des ensembles mieux calibrés exploitables dans un schéma d'auto-apprentissage.

Nous pouvons déjà noter que dans le cadre crédal, et en utilisant la perte (1), les étiquettes précises et manquantes sont des cas spécifiques d'étiquettes crédales : il n'est donc pas nécessaire de distinguer les ensembles $\mathcal{D}_L$ et $\mathcal{D}_U$ l'un de l'autre, et $\mathcal{D}_L \cup \mathcal{D}_U$ peut être simplement vu comme un cas particulier d'ensemble de données crédales.

Notre approche consiste à diviser $\mathcal{D}_L$ en deux ensembles $\mathcal{D}_T$ et $\mathcal{D}_C$. Nous proposons ensuite d'apprendre un premier classifieur h_{θ^0} sur l'ensemble $\mathcal{D}_T$ étiqueté, par minimisation d'une fonction de coût classique. Nous appliquons ensuite la méthode IVAP pour produire des étiquettes crédales sur les observations de $\mathcal{D}_U$ nous obtenons ainsi un ensemble de données $\mathcal{D}_U^0$ avec des étiquettes crédales. Nous apprenons ensuite un nouveau modèle h_{θ^1} à partir de $\mathcal{D}_T \cup \mathcal{D}_U^0$ via l'équation (1), et recommençons la procédure. À une itération j de cette procédure itérative d'auto-apprentissage, nous apprenons

$$h_{\theta^j} = \arg\min_{\theta \in \Theta} \sum_{x \in \mathcal{D}_L \cup \mathcal{D}_U} \mathcal{L}_{min}(K_x^{j-1}, h_{\hat{\theta}^{j-1}}(x)), \tag{3}$$

où K_x^{j-1} est l'ensemble crédal résultant de l'application de l'IVAP à x avec $\mathcal{D}_C$ et $h_{\hat{\theta}^{j-1}}$ comme modèle, et où pour tout $x \in \mathcal{D}_L$,

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

afin d'éviter les problèmes numériques dus à $ln(0)$. Cette procédure itérative peut ensuite être répétée par exemple jusqu'à ce que les performances sur l'ensemble de test commencent à se dégrader. La première boucle de cette procédure itérative est illustrée par la figure 2. L'ensemble de la procédure itérative est résumé dans l'algorithme 2. Il convient de noter que nous pourrions commencer immédiatement par utiliser $\mathcal{D}_L \cup \mathcal{D}_U$, en utilisant simplement des intervalles vacants ou presque vacants pour les observations dans $\mathcal{D}_U$. Dans le présent article, nous choisissons d'initialiser en utilisant uniquement $\mathcal{D}_L$, pour la simple raison que cela facilite la comparaison avec les approches d'auto-apprentissage standard

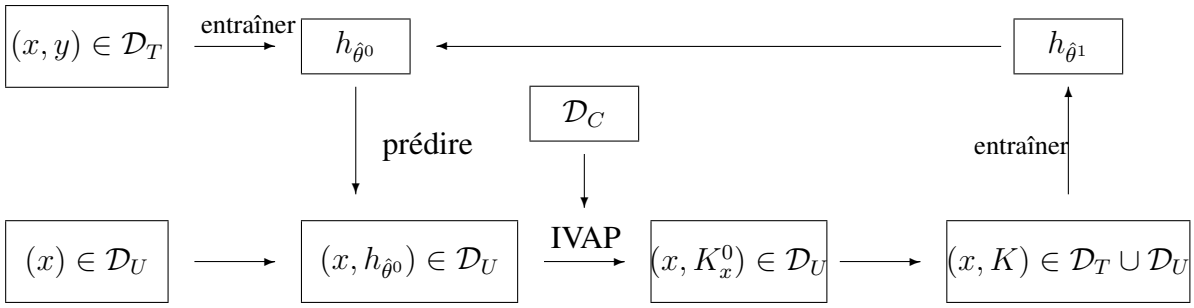


FIGURE 2 – Boucle d'initialisation de l'auto-apprentissage utilisant les prédicteurs de Venn-Abers

Algorithm 1 Auto-apprentissage avec prédicteurs de Venn-Abers

Require: Un ensemble étiqueté $\mathcal{D}_T$, un ensemble de calibrage $\mathcal{D}_C$, un ensemble non-étiqueté $\mathcal{D}_U$

Require: Un espace d'hypothèses Θ utilisé pour apprendre un modèle à valeurs réelles $h_\theta : \mathcal{X} \to \mathcal{Y}$

Require: Un nombre e de iterations (ou un critère d'arrêt)

$i \longleftarrow 0$

entraîner h_{θ^0} en utilisant $\mathcal{D}_T$

étiqueter l'observation $x \in \mathcal{D}_U$ par K_x généré par IVAP sur $h_{\theta^0}(x)$ en utilisant $\mathcal{D}_C$

while $i \leq e$ **do**

$i \longleftarrow i + 1$

entraîner h_{θ^i} en utilisant $\mathcal{D}_T \cup \mathcal{D}_U$

étiqueter l'observation $x \in \mathcal{D}_U$ par K_x généré par IVAP sur $h_{\theta^i}(x)$ en utilisant $\mathcal{D}_C$

end while

return h_{θ^e}

4 Expériences

Cette section présente des résultats expérimentaux, qui montrent notamment que l'utilisation de prédicteurs de Venn- Abers combinés à l'apprentissage crédal dans le cadre de l'auto-apprentissage permet en général d'obtenir une convergence plus rapide et de meilleurs résultats, ainsi qu'une plus grande robustesse dans les cas où l'auto-apprentissage classique ne donne pas de bons résultats.

4.1 Illustration sur des données synthétiques

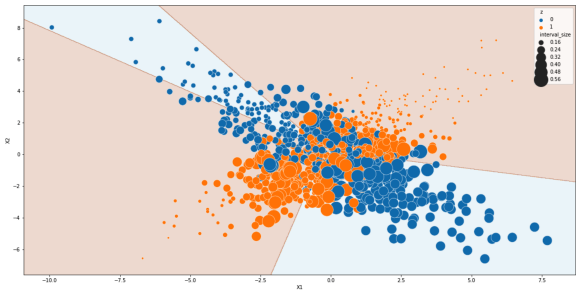
Avant de tester notre approche sur des ensembles de données réels, nous donnons une petite illustration du comportement de notre méthode sur un ensemble de données synthétiques. Cet ensemble de données avec $X \in \mathbb{R}^2$ est composé de deux distributions conditionnelles gaussiennes

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

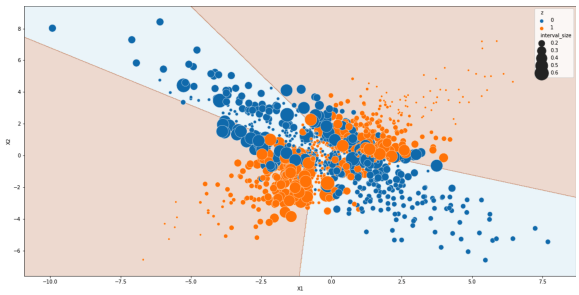
Nous avons fixé $n = 1000$ et divisé $\mathcal{D}$ en $\mathcal{D}_T$ de taille 80, $\mathcal{D}_C$ de taille 20 et $\mathcal{D}_U$ de taille 900. Nous avons entraîné un réseau de neurones avec une couche cachée de 3 neurones, avec un taux d'apprentissage $\lambda = 0,2$ sur $\mathcal{D}_T$; puis nous avons appliqué notre méthode avec 10 itérations. Nous avons choisi ce type de classifieur car ils sont souvent mal calibrés [8]. L'évolution de la frontière de décision, ainsi que la taille des ensembles crédaux produits par notre méthode, sont illustrées dans la figure 3.

Nous pouvons observer que les frontières de décision ainsi que notre certitude (ou confiance dans les prédictions effectuées) évoluent au fil du temps, même si l'ensemble $\mathcal{D}_C$ reste inchangé. Par exemple, on constate que la région de décision pour la classe négative a tendance à

FIGURE 3 – Évolution de la limite de décision et de la taille des intervalles sur 10 itérations d'auto-apprentissage à l'aide de prédicteurs de Venn-Abers (données synthétiques)



(a) Itération 1



(b) Itération 10

se rétrécir entre la 1ère et la 10ème itération, et que la taille moyenne de l'intervalle associé aux points de données de test appartenant à la classe négative a tendance à diminuer dans le quadrant inférieur droit, tout en augmentant dans le quadrant supérieur gauche. De même, la taille de l'intervalle associée aux exemples dans la région autour du point $(0, 0)$ où les classes se chevauchent a tendance à diminuer avec le nombre d'itérations, ce qui montre que nous sommes de plus en plus sûrs de nos estimations dans ces régions.

La figure 4 montre les probabilités (de la classe positive) produites par le réseau de neurones (sans calibration post-hoc) à la fin de l'apprentissage, ainsi que la taille des ensembles crédaux données par l'IVAP. Ces probabilités sont raisonnablement précises, et nous pouvons observer qu'il n'y a pas nécessairement de lien étroit entre les valeurs de probabilité et la fiabilité de l'estimation, que nous assimilons à la taille des intervalles de sortie : nous pouvons observer des estimations probabilistes plutôt extrêmes (éloignées de 0,5) associées à des intervalles grands ou petits, ainsi que des estimations probabilistes ambiguës (proches de 0,5) également associées à des intervalles grands ou petits.

4.2 Données réelles

Pour tester notre méthode sur des données réelles, nous avons utilisé six jeux de données : Breastcancer, Digits, Australian, Banknote, Heart disease et Adult (mis gracieusement à disposition par l'UCI [6]). Comme nous ne considérons que la classification binaire, pour l'ensemble de données Digits, nous avons regroupé les nombres pairs dans la classe 0 et les nombres impairs dans la classe 1. Pour l'ensemble de données Adult, nous avons sélectionné aléatoirement 5000 exemples sur l'ensemble du jeu de données, en respectant les proportions des classes de manière à obtenir un ensemble de test déséquilibré par rapport auquel les performances de notre modèle peuvent être évaluées. Nous avons utilisé comme classifieur un réseau neuronal à une couche cachée (l'architecture et le taux d'apprentissage λ sont présentés pour chaque jeu de données dans le tableau 1) appris par gradient stochastique. La taille du batch a été fixée à 10. Nous utilisons un réseau de neurones pour les mêmes raisons que nous avons fait pour les données synthétiques (mauvaise calibration).

Nous avons divisé chaque ensemble de données

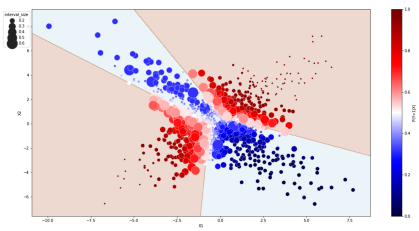


FIGURE 4 – Probabilités produites par le classifieur et tailles des intervalles produits par IVAP à l'itération 10

TABLEAU 1 – Architecture et hyperparamètre

Jeux de données	Nombre des neurones (couche cachée)	λ
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

en quatre nouveaux ensembles : $\mathcal{D}_T$, $\mathcal{D}_U$, $\mathcal{D}_C$ et $\mathcal{D}_t$ (l'ensemble de test). Nous avons comparé trois stratégies d'auto-apprentissage différentes :

1. une procédure standard et classique d'auto-apprentissage (SL) consistant à ajouter un sous-ensemble de nouvelles données étiquetées à chaque itération (le lot de données pour lequel la prédiction sont les plus éloignées de 0.5, c'est-à-dire celles pour lesquelles $|0.5 - h_{\theta^i}|$ est maximale). La taille du sous-ensemble est fixée à 2 % de l'ensemble de données initial ;
2. auto-apprentissage à l'aide d'étiquettes probabilistes (SLSL) : à chaque itération, nous étiquetons $\mathcal{D}_U$ avec $h_{\theta^i}(x)$, en ajoutant ces données pseudo-étiquetées à $\mathcal{D}_U$ avant d'entraîner le classifieur sur $\mathcal{D}_L \cup \mathcal{D}_U$;
3. l'auto-apprentissage à l'aide de prédicteurs Venn-Abers (SLVA), notre proposition décrite à la section 3 et l'algorithme 1.

La séparation des données est effectuée 10 fois sur des graines différentes : pour chaque séparation, nous appliquons chacune des trois stratégies avec 30 itérations. Pour chaque séparation, 20% des données ont été conservées avec $\mathcal{D}_t$ comme test, et sur les 80% restants, 80% ont été conservées comme données étiquetées pour $\mathcal{D}_T$, 5% ou 10% [3] ont été conservés pour $\mathcal{D}_C$ afin de garantir que $|\mathcal{D}_C| \geq 20$, et le reste a été placé dans $\mathcal{D}_U$.

Pour chaque stratégie, nous avons mesuré la précision moyenne $\overline{a}$ des 30 itérations sur les 10 différentes séparations, la précision moyenne a_{30} à l'itération 30 et son écart-type $\sigma(a_{30})$ à l'itération 30. Les résultats sont présentés dans le tableau 2 et la figure 5.

Ces résultats nous amènent à formuler plusieurs observations :

— SLVA obtient généralement de meilleurs résultats que les autres approches. C'est certainement vrai en moyenne, comme le montrent les trois premières colonnes du tableau 2, où notre approche surpasse les approches classiques 5 fois sur 6 ; mais aussi à la fin du processus d'apprentissage (a_{30} dans le tableau 2), où notre approche surpasse les approches classiques 4 fois sur 6, et reste proche de la meilleure méthode dans les deux autres cas.

— SLVA est également plus robuste pour tous les ensembles de données utilisés, puisqu'il s'agit de la méthode la plus performante ou qu'elle en reste proche, alors que SL et $SLSL$ présentent une plus grande variabilité (par exemple, $SLSL$ n'est pas performant sur Digits, SL est très mauvais sur Adult, et les deux sont moins performants sur Breastcancer).

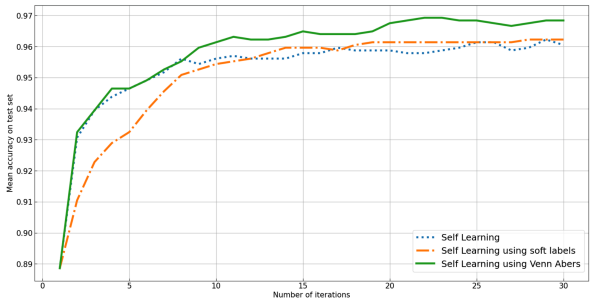
— SLVA converge également plus rapidement vers des performances « asymptotiques », comme l'indique le fait que la pente de la courbe verte est généralement plus raide dans la figure 5, et que $\overline{a}$ (la précision moyenne) est plus élevée pour SLVA.

— SLVA est la seule méthode qui montre de bonnes performances sur le jeu de données Adult : sur ce dernier, la convergence de $SLSL$ est très lente, et les performances de SL sont très mauvaises — et se dégradent même rapidement après la cinquième itération.

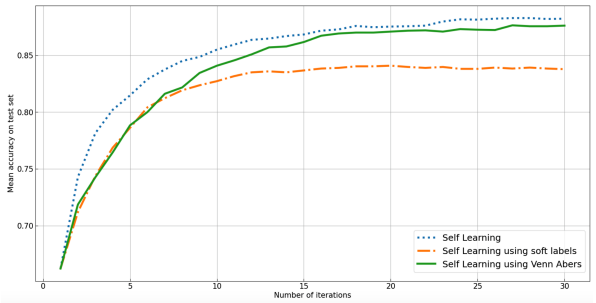
En termes de variance, toutes les méthodes semblent se situer au même niveau, aucune ne présentant d'avantage par rapport aux autres. Toutefois, les remarques ci-dessus indiquent que l'utilisation d'ensembles crédaux calibrés peut être considérée comme une alternative intéressante aux approches d'auto-apprentissage

3. soit 4% ou 8% de l'ensemble de données initial

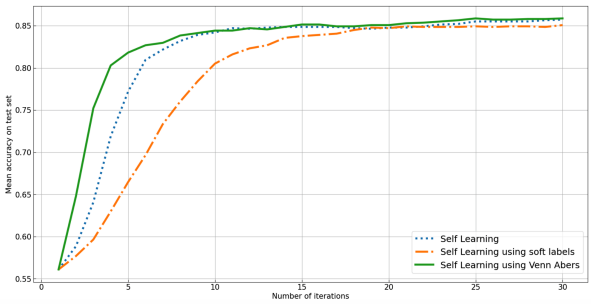
FIGURE 5 – Précisions obtenues sur $\mathcal{D}_t$ pour les six jeux de données.



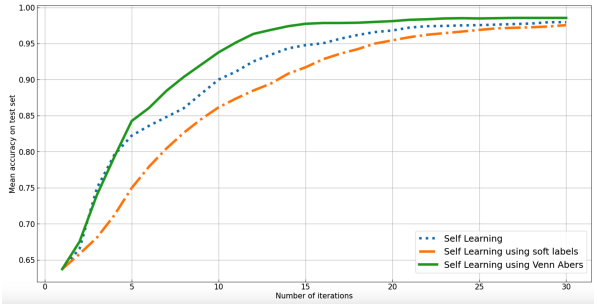
(a) Breastcancer



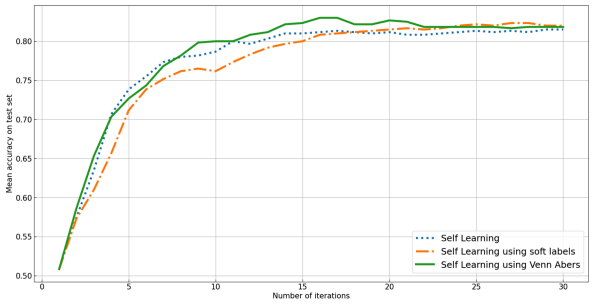
(b) Digits



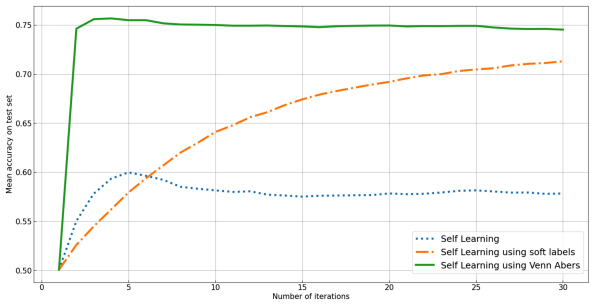
(c) Australian



(d) Banknote



(e) Heart disease



(f) Adult

standard, car elle présente généralement de meilleures performances sans surcoût calculatoire pour des problèmes de classification binaires lorsque la fonction de coût D_{KL} est utilisée.

5 Conclusions et perspectives

Dans cet article, nous avons introduit une nouvelle approche pour les problèmes d'auto-apprentissage, dans le cas de problèmes de classification binaires. Notre approche consiste à produire des pseudo-étiquettes pour un certain nombre de données supplémentaires non-étiquetées, qui seront utilisées pour réapprendre le modèle.

L'utilisation d'une fonction de coût appropriée permet de prendre en compte l'incertitude associée aux pseudo-étiquettes pour limiter les biais lors du réapprentissage du modèle. Cette approche permet d'exploiter au mieux les données disponibles, en particulier lorsqu'on dispose de peu de données étiquetées et de nombreuses données non étiquetées.

Les expériences montrent que notre approche se compare favorablement aux approches classiques pour l'auto-apprentissage : elle donne des performances meilleures ou comparables. En particulier, il semble qu'elle permet d'améliorer la robustesse et d'éviter une convergence lente pour les jeux de données déséquilibrés. En outre, la prise en compte des incertitudes (à la fois épistémiques et aléatoires) dans le processus d'auto-apprentissage semble à même d'améliorer l'acceptabilité des approches d'auto-apprentissage par les utilisateurs.

L'un de nos objectifs est effectuer d'autres expériences pour confirmer nos observations et tester les limites de l'approche présentée. Il serait particulièrement intéressant de confirmer le bon comportement de notre approche dans le cas de données déséquilibrées, mais aussi de voir comment elle se comporte lorsque la taille de l'ensemble d'étalonnage évolue. Nous pourrions également faire varier l'ensemble de calibration en composition et en taille : à chaque itération, on pourrait envisager de rééchantillonner l'ensemble de calibration à l'intérieur de $\mathcal{D}_L$, plutôt que de conserver les mêmes données. Nous pourrions également envisager de transférer une quantité croissante de données de $\mathcal{D}_L$ à $\mathcal{D}_C$ au fur et à mesure que les étiquettes crédibles de $\mathcal{D}_U$ deviennent plus précises.

Enfin, une extension évidente consisterait à étendre notre approche d'apprentissage auto-supervisé à des problèmes plus complexes, notamment aux problèmes de classification multiclasses. Dans ce cas, les prédicteurs de Venn [24] produisent $|\mathcal{Y}|$ probabilités. Une question ouverte est de savoir comment gérer l'augmentation de la complexité dans ce contexte — notons qu'il existe une version inductive des prédicteurs de Venn [10], qui présente une complexité $O(n^2 * |\mathcal{Y}|)$. Une autre direction prometteuse pour l'apprentissage crédal auto-supervisé utilisant des sorties calibrées est l'« adaptation graduelle du domaine », cadre dans lequel une approche d'auto-apprentissage est utilisée pour résoudre un problème d'apprentissage par transfert [28, 9, 26].

Références

[1] 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.

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

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

[4] Pedro Domingos and Michael Pazzani. Beyond independence : Conditions for the optimality of the simple bayesian classi er.

TABLEAU 2 – Performances sur $\mathcal{D}_t$ pour les 6 ensembles de données

	$\overline{a}$			a_{30}			$\sigma(a_{30})$		
Jeux des données	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**

In *Proc. 13th Intl. Conf. Machine Learning*, pages 105–112, 1996.

[5] 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.

[6] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

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

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

[9] 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.

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

[11] 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.

[12] Julian Lienen, Caglar Demir, and Eyke Hüllermeier. Conformal credal self-supervised learning, 2022.

[13] 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.

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

[15] 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.

[16] Jonathan Peck, Bart Goossens, and Yvan Saeys. Detecting adversarial manipulation using inductive venn-abers predictors. *Neurocomputing*, 416 :202–217, 2020.

[17] 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 Compu-*

ter Vision and Pattern Recognition, pages 1578–1588, 2022.

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

[19] 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.

[20] 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.

[21] 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.

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

[23] Vladimir Vovk and Ivan Petej. Venn-abers predictors. *arXiv preprint arXiv :1211.0025*, 2012.

[24] Vladimir Vovk and Ivan Petej. Venn-abers predictors, 2012.

[25] 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.

[26] 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.

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

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

A Divergence de Kullback-Leibler comme fonction de coût

Dans cette annexe, nous expliquons pourquoi nous préférons $D_{KL}(P \mid\mid h_\theta(x))$ à l'entropie croisée binaire lorsque nous considérons la perte crédale $\mathcal{L}_{min}$.

Comme expliqué, $\mathcal{L}_{min}$ est le minimum de la fonction de perte $\mathcal{L}$ obtenue sur l'ensemble des distributions correspondant à l'étiquette crédale K pour la probabilité estimée $h_\theta(x)$. Nous voulons que cette $\mathcal{L}_{min}$ ait les caractéristiques suivantes :

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

En effet, nous souhaitons ne pas pénaliser une prédiction $h_\theta(x)$ qui se situe à l'intérieur de l'étiquette crédale.

Toutefois, ces propriétés ne sont pas satisfaites par l'entropie croisée binaire habituelle BCE, la principale raison étant qu'étant donné un $h_\theta(x)$, BCE est linéaire en p et sa dérivée par rapport à p est constante :

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

Cela signifie que le minimum de l'entropie croisée binaire $\mathcal{L}_{min}(K_x, h_{\theta(x)})$ est atteint pour l'une des bornes $\underline{p}$ ou $\overline{p}$, que l'on ait $h_\theta(x) \in K$ ou non.

En revanche, une fois fixée $h_\theta(x)$, la divergence D_{KL} n'est pas linéaire en p et sa dérivée par rapport à p est donnée par :

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

Par conséquent, D_{KL} est convexe en p et son minimum est atteint pour

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

Comme D_{KL} est convexe, si $\overline{p} < h_\theta(x)$, le minimum est atteint pour $p = \overline{p}$ (et inversement pour $p = \underline{p}$ if $\underline{p} > h_\theta(x)$). Ainsi, D_{KL} satisfait aux trois propriétés mentionnées ci-dessus, ce qui explique pourquoi cette fonction de coût est préférable à BCE.

B Prédicteurs de Venn-Abers inductifs

L'algorithme 2 se réfère à l'approche IVAP décrite à la section 2.3.

Algorithm 2 Algorithme IVAP

Require: fonction h_θ entraînée sur $\mathcal{D}_T$, ensemble $\mathcal{D}_C$, observation x ;

calculer $(h_\theta(x_{l+1}), \ldots, h_\theta(x_n), h_\theta(x))$;
for $y \in \{0, 1\}$ **do**
  Apprendre la régression isotonique g sur $((h_\theta(x_{l+1}), y_{l+1}), \ldots, (h_\theta(x_n), y_n), (h_\theta(x), y))$
  définir p_y = $g(h_\theta(x, y))$
end for
return (p_0, p_1)