Iterated Belief Change as Learning

Nicolas Schwind[1] , **Katsumi Inoue**[2] , **Sébastien Konieczny**[3] and **Pierre Marquis**[3,4]
[1]National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
[2]National Institute of Informatics, Tokyo, Japan
[3]Univ. Artois, CNRS, CRIL, Lens, France
[4]Institut Universitaire de France
nicolas-schwind@aist.go.jp, inoue@nii.ac.jp, konieczny@cril.fr, marquis@cril.fr

Abstract

In this work, we show how the class of improvement operators – a general class of iterated belief change operators – can be used to define a learning model. Focusing on binary classification, we present learning and inference algorithms suited to this learning model and we evaluate them empirically. Our findings highlight two key insights: first, that iterated belief change can be viewed as an effective form of online learning, and second, that the well-established axiomatic foundations of belief change operators offer a promising avenue for the axiomatic study of classification tasks.

1 Introduction

Belief Change Theory [Alchourrón *et al.*, 1985; Gärdenfors, 1988; Katsuno and Mendelzon, 1991] provides a principled framework for modifying an agent's current beliefs in response to new information. Iterated belief revision [Darwiche and Pearl, 1997; Jin and Thielscher, 2007; Booth and Meyer, 2006] extends this framework to accommodate sequences of new information, addressing the challenge of revising an agent's beliefs over time. In both cases, the ultimate goal is to improve the agent's beliefs to better reflect the real world. While the methodologies of these two approaches differ, their objective aligns with that of Machine Learning (ML): deriving an accurate approximation of the real world from data.

Despite this conceptual similarity, connections between Belief Change Theory and ML remain largely unexplored, apart from a few notable contributions in philosophical logic [Kelly, 1998; Kelly, 2014; Baltag *et al.*, 2011; Baltag *et al.*, 2019], inductive logic programming [Wrobel, 1994; Pagnucco and Rajaratnam, 2005], and computational learning theory [Goldsmith *et al.*, 2004].

A major difference is that *primacy of update*, which requires fully adopting new information after each revision, is a key principle in belief revision. However, this principle is incompatible with typical ML scenarios involving noisy data, as it leads to substantial changes in the agent's epistemic state at each learning step.

Improvement operators [Konieczny and Pino Pérez, 2008; Konieczny *et al.*, 2010], which generalize iterated belief revision, relax the primacy of update. Soft improvement operators [Konieczny and Pino Pérez, 2008], in particular, allow incremental changes that better reflect the iterative nature of learning. When the same information is encountered again, its reliability is slightly adjusted. This mirrors the behavior of online ML methods for classification, where each labeled example causes gradual changes to the estimated probabilities of the classes.

Figure 1 illustrates this analogy. Most ML methods also adjust examples similar to the labeled example, those "near" the observed instance. A comparable mechanism can be introduced into the iterated belief change framework: both the observed example and its neighbors can have their reliability updated through an improvement operator.

This paper focuses on binary classification and shows that improvement-based models built this way deliver reasonable learning performance. We compare them to standard ML methods on benchmark datasets. Results show the improvement-based model slightly outperforms Naive Bayes and achieves better recall than most existing methods. Thus, soft improvement operators offer a promising new approach to learning from examples.

Although an initial exploration, this connection is important for both KR and ML communities. It links two fundamentally different tasks – belief revision and supervised learning – with distinct goals and methods. From the ML side, this work opens the way to learning models that emphasize interpretability and offer strong guarantees linking the model to the data. These properties are crucial for trustworthy AI but often lacking in current ML models. If rationality principles are established, they could serve as safeguards to ensure the model evolves correctly with new evidence.

The proofs and code used to retrieve datasets and conduct experiments are available in [Schwind *et al.*, 2025].

2 Preliminaries

We assume the reader familiar with basics of ML, including standard models (see, e.g., [Shalev-Shwartz and Ben-David, 2014] for an introduction). In this work, the focus is laid on tabular datasets, where instances are represented as vectors of Boolean features, aligning with many learning methods, especially in data mining. Such a Boolean encoding aids normalization, which often improves model performance. Although features in tabular data are usually not Boolean in

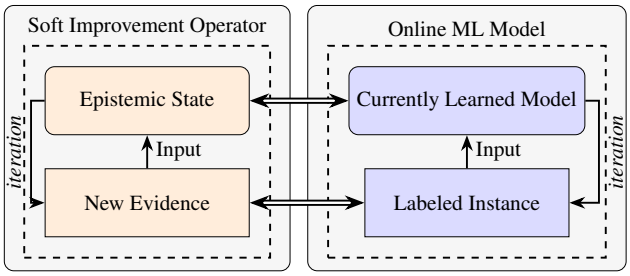

Figure 1: Analogies between classifier learning models and belief change operators.

essence, they can be converted accordingly. Categorical features are transformed via standard one-hot encoding, creating a Boolean feature for each domain value. Numerical features are converted by selecting thresholds within their domain to produce Boolean features. These thresholds are chosen by analyzing the data distribution (e.g., percentiles) or by optimizing split metrics like Gini impurity or entropy, as commonly done in tree-based models such as decision trees, random forests, and boosted trees.

Let $\mathcal{L}$ be a propositional language built from a finite set of propositional variables P, standard logical connectives, and the Boolean constants $\bot$ (false) and $\top$ (true). A world on P is a mapping from P to $\{0, 1\}$, and Ω denotes the set of all such worlds. A world ω satisfies a formula φ if ω makes φ true; the set of such worlds is denoted by $[\varphi]$. A formula is consistent if $[\varphi] \neq \emptyset$, and it is complete if $[\varphi] = \{\omega\}$ for some $\omega \in \Omega$. The symbol $\models$ denotes logical entailment, and $\equiv$ logical equivalence: $\varphi \models \psi$ iff $[\varphi] \subseteq [\psi]$, and $\varphi \equiv \psi$ iff $[\varphi] = [\psi]$.

Iterated Belief Change provides a principled framework for modeling how a rational agent's beliefs evolve with successive pieces of evidence. An agent's belief state, called an *epistemic state*, includes both the agent's current beliefs and conditional information that guides how beliefs should change in response to new inputs. Formally, an epistemic state can be any object Ψ, from which the agent's current beliefs are extracted via a mapping Bel, such that $Bel(\Psi) \in \mathcal{L}$ [Darwiche and Pearl, 1997]. An *epistemic space* is then a tuple $\mathcal{E} = \langle E, Bel \rangle$, where E is the set of all epistemic states in the space [Schwind *et al.*, 2022].

A standard example of epistemic space is built with Ordinal Conditional Functions (OCFs) [Spohn, 1988; Williams, 1995]. An OCF κ is a mapping associating each world with a non-negative integer [1] such that $\kappa(\omega) = 0$ for some world ω.

Definition 1 (OCF epistemic space)**.** *The* OCF epistemic space *is the epistemic space $\mathcal{E}_{ocf} = \langle E_{ocf}, Bel_{ocf} \rangle$ where:*

- *E_{ocf} is the set of all* OCFs *over Ω;*
- *Bel_{ocf} is the mapping associating each* OCF *κ from E_{ocf} with a formula $\psi \in \mathcal{L}$ such that $[\psi] = \{\omega \in \Omega \mid \kappa(\omega) = 0\}$.*

Given an epistemic space $\mathcal{E} = \langle E, Bel \rangle$, an iterated belief change operator $\circ$ on $\mathcal{E}$ is a mapping associating each epistemic state $\Psi \in E$ and each formula $\mu \in \mathcal{L}$ with a new epistemic state $\Psi \circ \mu \in E$, i.e., $\circ : E \times \mathcal{L} \to E$.

Improvement operators [Konieczny and Pino Pérez, 2008; Konieczny *et al.*, 2010; Medina Grespan and Pino Pérez, 2013] generalize iterated revision operators [Darwiche and Pearl, 1997] by dropping the success postulate (R*1), which requires the revised beliefs to entail the input formula. They are defined by nine rationality principles (I1)–(I9) (see [Konieczny and Pino Pérez, 2008; Konieczny *et al.*, 2010] for details). We recall only the weak primacy of update:

(I1) $\exists k \in \mathbb{N}_*$ s.t. $Bel(\Psi \circ^k \alpha) \models \alpha$,

where $\Psi \circ^1 \alpha = \Psi \circ \alpha$, and if $k > 1$, $\Psi \circ^k \alpha = (\Psi \circ^{k-1} \alpha) \circ \alpha$.

This property states that after receiving α repeatedly, it eventually becomes believed.

We now introduce a simple example of an improvement operator defined over the OCF epistemic space:

Definition 2 (Basic shifting operator $\circ_{+1}$)**.** *The* basic shifting operator *$\circ_{+1}$ on $\mathcal{E}_{ocf}$ is defined for any* OCF *κ and formula α by $\kappa' = \kappa \circ_{+1} \alpha$, where for each world $\omega \in \Omega$:*

$$\kappa'(\omega) = \begin{cases} \kappa(\omega) - x & \text{if } \omega \in [\alpha] \\ \kappa(\omega) + 1 - x & \text{otherwise,} \end{cases}$$

where $x = 0$ if $Bel(\kappa) \wedge \alpha \not\models \bot$, and $x = 1$ otherwise.

Upon receiving new input α, the improvement operator $\circ_{+1}$ adjusts κ by increasing the value of worlds $\omega \in [\neg\alpha]$ by 1 (i.e., $\kappa'(\omega) = \kappa(\omega) + 1$), while leaving $\kappa(\omega)$ unchanged for worlds $\omega \in [\alpha]$. A normalization step ($-x$) then ensures $\min\{\kappa'(\omega) \mid \omega \in \Omega\} = 0$, as required by the definition of an OCF. This operator resembles Spohn's n-conditionalization [Spohn, 1988], except n here depends on the prior plausibility of α (shifted by -1). It also relates to the one-improvement operator in [Konieczny and Pino Pérez, 2008].

2.1 Morphological Dilation and Erosion

An essential component for defining our learning operators is the notion of formula dilation and, dually, formula erosion [Bloch and Lang, 2002]. Both rely on a *neighborhood* B, which is a mapping associating each world $\omega \in \Omega$ with a set of worlds $B_\omega \subseteq \Omega$ such that (i) $\omega \in B_\omega$, and (ii) $\omega' \in B_\omega$ implies that $\omega \in B_{\omega'}$, for all worlds $\omega, \omega' \in \Omega$.

A neighborhood B induces a dilation δ_B and an erosion ϵ_B, both mappings from formulas to formulas, defined for each $\varphi \in \mathcal{L}$ as follows:

- $[\delta_B(\varphi)] = \{\omega \in \Omega \mid B_\omega \cap [\varphi] \neq \emptyset\}$
- $[\epsilon_B(\varphi)] = \{\omega \in \Omega \mid B_\omega \subseteq [\varphi]\}$

For $k \geq 0$, the k-dilation $\delta_B^k(\varphi)$ and k-erosion $\epsilon_B^k(\varphi)$ are defined inductively as follows: $\delta_B^0(\varphi) = \varphi$, and $\delta_B^k(\varphi) = \delta_B(\delta_B^{k-1}(\varphi))$ for $k > 0$. Similarly, $\epsilon_B^0(\varphi) = \varphi$, and $\epsilon_B^k(\varphi) = \epsilon_B(\epsilon_B^{k-1}(\varphi))$ for $k > 0$.

Neighborhoods can be derived from various similarity or distance measures on Boolean vectors (see [Choi *et al.*, 2009] for an overview of such measures). In the present setting, the neighborhood B_ω of each world ω depends solely on ω and need not be defined uniformly over Ω. Leveraging such instance-specific neighborhoods is known to be beneficial from a learning perspective [Ye *et al.*, 2016].

[1] In the original definition OCFs are defined on ordinals [Spohn, 1988], but here, as in most cases, integers suffice.

3 From Epistemic Spaces to Classifier Spaces

In this section, we formalize the change operation underlying the learning process of a binary classifier, drawing on the framework of iterated belief change introduced earlier.

The binary classification task consists in predicting whether a given instance is a member of some target class or concept. We assume that each instance is described by a vector of values associating each *feature* x_i from a given finite and fixed set $\mathbf{X}$ with a value. We focus on binarized instance descriptions, so that the feature set $\mathbf{X}$ consists of a set of binary features. An instance can be associated with an output (1 or 0), characterizing whether the instance is predicted as positive or negative.

We represent each binary feature as a propositional variable, so that the feature set $\mathbf{X}$ corresponds to the set $P_{\mathbf{X}} = \{x_1, \ldots, x_n\}$. A world over $P_{\mathbf{X}}$ is called an *instance description*. Let $\Omega_{\mathbf{X}}$ be the set of all such instance descriptions, i.e., all worlds over $P_{\mathbf{X}}$, and let $\mathcal{L}_{\mathbf{X}}$ denote the propositional language generated from $P_{\mathbf{X}}$ using the standard connectives. Each formula $\varphi \in \mathcal{L}_{\mathbf{X}}$ can represent a concept predicted by a binary classifier: an instance description $\omega_{\mathbf{X}} \in \Omega_{\mathbf{X}}$ satisfies φ if and only if it is classified as positive. This modeling approach follows that of [Schwind *et al.*, 2023].

To represent labeled data, as in training examples, we extend $P_{\mathbf{X}}$ with an additional *output* (class) variable $\mathbf{y}$ indicating whether an instance is positive (1) or negative (0). Formally, we define $P = P_{\mathbf{X}} \cup \{\mathbf{y}\}$, with $\mathbf{y} \notin P_{\mathbf{X}}$. A world ω over P is called a *labeled instance*, and Ω denotes the set of all such labeled instances. A labeled instance ω is positive if $\omega(\mathbf{y}) = 1$ and negative if $\omega(\mathbf{y}) = 0$.

For any labeled instance $\omega \in \Omega$, its feature description (i.e., the corresponding world over $P_{\mathbf{X}}$) is denoted $\omega_{\mathbf{X}}$. The complete formula $\varphi_\omega \in \mathcal{L}$ such that $[\varphi_\omega] = \{\omega\}$ is called a *training instance*, with ω its associated labeled instance. Let $\mathcal{L}^c$ denote the set of all such training instances. A *dataset* D is defined as a finite sequence of training instances: $\mathsf{D} = (\varphi^s_\omega)_{1 \leq s \leq m}$, and $\mathcal{D}$ denotes the set of all possible datasets.

To model binary classifiers, we move beyond treating a classifier as a static propositional formula (as in [Schwind *et al.*, 2023]) and instead adopt a dynamic perspective that captures the learning process itself. This shift mirrors the move from one-step belief change – where an agent's epistemic state is represented by a single propositional formula [Katsuno and Mendelzon, 1991] – to *iterated* belief change, where epistemic states have a richer structure and yield formulas via the mapping Bel [Darwiche and Pearl, 1997].

Similarly, modeling the evolution of a classifier during learning requires a more expressive framework. To address this, the notion of an epistemic state is lifted to that of a *binary classifier state*. Likewise, the mapping Bel, which assigns each epistemic state with a propositional formula representing the agent's beliefs, is lifted to a mapping Pos, which assigns each classifier state with a formula characterizing the concept it predicts. The pair consisting of a classifier state and the mapping Pos thus forms a *classifier space*, in direct analogy with an epistemic space. The distinct notation Pos highlights the shift in perspective: from beliefs held by an agent to predictions made by a classifier. Formally:

Definition 3 (Classifier space)**.** *A* classifier space *is a tuple* $\mathcal{E} = \langle E, Pos \rangle$*, where* E *is a set of* binary classifiers *and* Pos *is a mapping from* E *to* $\mathcal{L}_{\mathbf{X}}$*.*

Thus, the formula $Pos(\Psi)$ represents the concept predicted by Ψ: the set $[Pos(\Psi)]$ contains the instance descriptions that Ψ classifies as positive, while $[\neg Pos(\Psi)]$ contains those classified as negative. Note that $Pos(\Psi)$ is not required to be a consistent formula, i.e., $[Pos(\Psi)]$ may be empty.

Standard epistemic spaces used for defining improvement operators can be directly adapted to build classifier spaces. Given an epistemic space $\mathcal{E} = \langle E, Bel \rangle$, one can construct a classifier space $\mathcal{E}' = \langle E, Pos \rangle$ by simply setting $Pos = Bel$. However, more meaningful examples of classifier spaces will be introduced in the next section.

Based on classifier spaces, we define an (online) learning operator:

Definition 4 (Learning operator)**.** *Let* $\mathcal{E} = \langle E, Pos \rangle$ *be a classifier space. A* learning operator $\odot$ *on* $\mathcal{E}$ *is a mapping* $\odot : E \times \mathcal{L}^c \to E$*.*

Thus, $\odot$ specifies how a classifier $\Psi \in E$ changes upon receiving a training instance $\varphi_\omega \in \mathcal{L}^c$, yielding a new classifier $\Psi \odot \varphi_\omega$. The framework is general enough to capture any binary classifier and online learning process, assuming binarized training data.

We now have the tools to define what we call a *learning framework*, representing a full learning process:

Definition 5 (Learning framework)**.** *A* learning framework *is a pair* $(\Psi_*, \odot)$*, where* $\Psi_* \in E$ *is a binary classifier called the* anchor*, and* $\odot$ *is a learning operator on* $\mathcal{E} = \langle E, Pos \rangle$*.*

The anchor Ψ_* serves as the initial classifier state, providing the starting point for learning. Thus, a learning framework is fully determined by two components: the learning operator $\odot$, which governs how classifiers evolve, and the anchor Ψ_*, which sets the initial state. Given a learning framework $(\Psi_*, \odot)$ and a training dataset $\mathsf{D} \in \mathcal{D}$, the resulting learned classifier, denoted $\Psi_* \odot \mathsf{D}$, is defined inductively by:[2]

- $\Psi_* \odot \emptyset = \Psi_*$
- $\Psi_* \odot (\mathsf{D} \sqcup (\varphi_\omega)) = (\Psi_* \odot \mathsf{D}) \odot \varphi_\omega$

The choice of anchor depends on the context: it may be a classifier pre-trained on prior data, an untrained classifier (e.g., initialized as trivially positive, $Pos(\Psi_*) \equiv \top$, or negative, $Pos(\Psi_*) \equiv \bot$), or a random element from E.

This formalization captures the iterative nature of online binary classifier learning and leads into the next section, where we introduce a specific class of learning frameworks.

4 Improvement-Based Learning

In this section, we introduce a concrete class of improvement-based learning operators. These operators are defined on a classifier space that extends the OCF epistemic space. In this space, each binary classifier is represented as a tuple $(\mathsf{D}, \kappa, \tau)$, consisting of a training dataset D, an OCF κ, and a threshold $\tau \in \mathbb{R}$. Such classifiers are called TOCFS:

[2] $\sqcup$ denotes vector concatenation.

Definition 6 (TOCF classifier space)**.** *The* TOCF classifier space *is the tuple* $\mathcal{E}_{tocf} = \langle E_{tocf}, Pos_{tocf} \rangle$*, where:*

- $E_{tocf} = \mathcal{D} \times E_{ocf} \times \mathbb{R}$ *is the set of all* TOCFS
- Pos_{tocf} *is the mapping associating each* TOCF $(\mathsf{D}, \kappa, \tau) \in E_{tocf}$ *with a formula* $\psi \in \mathcal{L}_{\mathbf{X}}$ *such that* $[\psi] = \{\omega \in \Omega_{\mathbf{X}} \mid \kappa(\omega) \leq \tau\}$.

In this setting, a binary classifier is thus a TOCF $\Psi = (\mathsf{D}, \kappa, \tau)$.

The dataset D represents the training data used to construct Ψ. The OCF κ assigns a value to each instance description $\omega_{\mathbf{X}} \in \Omega_{\mathbf{X}}$, with lower values indicating greater plausibility that the instance is classified as positive by Ψ. The threshold τ specifies the classification boundary: an instance $\omega_{\mathbf{X}}$ is classified as positive (i.e., $\omega_{\mathbf{X}} \in [Pos(\Psi)]$) if $\kappa(\omega_{\mathbf{X}}) \leq \tau$, and as negative otherwise.

Our concrete class of learning operators is defined on the TOCF classifier space. Every operator in this class is specified by the following components:

- an improvement operator $\circ$ on the OCF epistemic space,
- a neighborhood B on $\Omega_{\mathbf{X}}$ (see Sec. 2.1), used to characterize formula dilation and erosion, and
- a *performance metric* $\mathbf{m} : \mathbb{N}^4 \to \mathbb{R}$.

In a nutshell, the improvement operator $\circ$ and the neighborhood B are used together to adjust the plausibility of worlds in the underlying OCF κ of a TOCF $\Psi = (\mathsf{D}, \kappa, \tau)$, while the performance metric $\mathbf{m}$ sets the threshold τ that best separates predicted positive and negative instances.

Given an improvement operator $\circ$, a neighborhood B, and a performance metric $\mathbf{m}$, we denote the corresponding learning operator by $\odot_{(\circ,B,\mathbf{m})}$. Then, for any TOCF $\Psi = (\mathsf{D}, \kappa, \tau)$ and any training instance φ_ω, the resulting TOCF $\Psi' = (\mathsf{D}', \kappa', \tau') = \Psi \odot_{(\circ,B,\mathbf{m})} \varphi_\omega$ is defined as follows.

First, we set $\mathsf{D}' = \mathsf{D} \sqcup (\varphi_\omega)$, that is, we augment the current training dataset by including the new instance φ_ω.

Second, we build the new OCF κ' using the improvement operator $\circ$ and the neighborhood B. This construction proceeds differently depending on whether the training instance φ_ω is labeled as positive or negative.

Learning from a positive instance. If the training instance φ_ω satisfies $\varphi_\omega \models \mathbf{y}$, the process unfolds as follows. Let $\varphi_{\omega_{\mathbf{X}}} \in \mathcal{L}^c_{\mathbf{X}}$ be the formula describing the instance (recall that $\omega_{\mathbf{X}}$ denotes its feature description). We begin by applying the improvement operator $\circ$ to $\varphi_{\omega_{\mathbf{X}}}$ in κ, yielding $\kappa \circ \varphi_{\omega_{\mathbf{X}}}$. We then continue this process iteratively on successive dilations of $\varphi_{\omega_{\mathbf{X}}}$: first on $\delta_B(\varphi_{\omega_{\mathbf{X}}})$, then on $\delta^2_B(\varphi_{\omega_{\mathbf{X}}})$, and so on, until reaching a fixed point.

Semantically, this process increases the plausibility of ω in κ, followed by an increase in the plausibility of its neighborhood, and then of neighborhoods of neighborhoods, so that the increase in plausibility assigned to a world becomes smaller as its distance from ω grows. Worlds that are not reachable from ω (that is, those not included in any dilation $\delta^k_B(\varphi_{\omega_{\mathbf{X}}})$ for any k) remain unaffected; their plausibility in κ stays the same.

Formally, when $\varphi_\omega \models \mathbf{y}$, we define the resulting OCF as $\kappa' = \kappa \bullet_+ \varphi_{\omega_{\mathbf{X}}} = \kappa \bullet^n_+ \varphi_{\omega_{\mathbf{X}}}$, where $\kappa \bullet^n_+ \varphi_{\omega_{\mathbf{X}}}$ is inductively defined as follows:

- $\kappa \bullet^0_+ \varphi_{\omega_{\mathbf{X}}} = \kappa$,
- $\kappa \bullet^{k+1}_+ \varphi_{\omega_{\mathbf{X}}} = (\kappa \bullet^k_+ \varphi_{\omega_{\mathbf{X}}}) \circ \delta^k_B(\varphi_{\omega_{\mathbf{X}}})$ for $k \geq 0$,
- $n = \min(\{k \in \mathbb{N} \mid [\delta^{n+1}_B(\varphi_{\omega_{\mathbf{X}}})] \in \{[\delta^n_B(\varphi_{\omega_{\mathbf{X}}})], \Omega_{\mathbf{X}}\}\})$

Learning from a negative instance. Conversely, when the training instance is negative ($\varphi_\omega \models \neg\mathbf{y}$), the process targets $\neg\varphi_{\omega_{\mathbf{X}}}$ and its successive erosions: $\epsilon_B(\neg\varphi_{\omega_{\mathbf{X}}})$, $\epsilon^2_B(\neg\varphi_{\omega_{\mathbf{X}}})$, and so on. At each step, plausibility is increased for the worlds satisfying the current erosion, continuing until reaching a fixed point (or stopping just before if the fixed point is an inconsistent formula). Semantically, this increases the plausibility of worlds that are farther from $\omega_{\mathbf{X}}$ under B, with the increase being greater the farther the world is. The world $\omega_{\mathbf{X}}$ itself remains unaffected. Formally, we define $\kappa' = \kappa \bullet_- \varphi_{\omega_{\mathbf{X}}} = \kappa \bullet^n_- \varphi_{\omega_{\mathbf{X}}}$, where $\kappa \bullet^n_- \varphi_{\omega_{\mathbf{X}}}$ is defined inductively as follows:

- $\kappa \bullet^0_- \varphi_{\omega_{\mathbf{X}}} = \kappa$,
- $\kappa \bullet^{k+1}_- \varphi_{\omega_{\mathbf{X}}} = (\kappa \bullet^k_- \varphi_{\omega_{\mathbf{X}}}) \circ \epsilon^k_B(\neg\varphi_{\omega_{\mathbf{X}}})$ for $k \geq 0$,
- $n = \min(\{k \in \mathbb{N} \mid [\epsilon^{n+1}_B(\neg\varphi_{\omega_{\mathbf{X}}})] \in \{[\epsilon^n_B(\neg\varphi_{\omega_{\mathbf{X}}})], \emptyset\}\})$

An alternative would be to handle negative instances using a *decrement* operator applied to $\varphi_{\omega_{\mathbf{X}}}$ and its successive dilations, mirroring the treatment of positive instances. While such decrement operators have been proposed [Sauerwald and Beierle, 2019], we do not adopt them here because, unlike the well-established duality between erosion and dilation, no formal duality exists between improvement and decrement operators. By relying on erosion and dilation – dual operations defined over a single neighborhood B – we maintain a unified learning mechanism using the same improvement operator for both positive and negative instances.

Given the distinction outlined above between learning from positive and negative instances, the new OCF is defined as:

$$\kappa' = \kappa \bullet \varphi_\omega = \begin{cases} \kappa \bullet_+ \varphi_{\omega_{\mathbf{X}}} & \text{if } \varphi_\omega \models \mathbf{y}, \\ \kappa \bullet_- \varphi_{\omega_{\mathbf{X}}} & \text{if } \varphi_\omega \models \neg\mathbf{y} \end{cases}$$

Third, we define the threshold τ', which acts as the "optimal separator." Recall that τ' partitions the space of instance descriptions such that instances with plausibility less than or equal to τ' are classified as positive, and those with plausibility greater than τ' are classified as negative. According to Def. 6, this corresponds to $[Pos(\Psi')] = \{\omega_{\mathbf{X}} \mid \kappa'(\omega_{\mathbf{X}}) \leq \tau'\}$. We aim to select τ' so as to optimize performance over the dataset D', using the performance metric $\mathbf{m}$.

For any candidate threshold $\tau \in \mathbb{N}$, we compute a confusion matrix $\mathsf{cm}(\tau) = (\mathsf{tp}, \mathsf{fp}, \mathsf{tn}, \mathsf{fn})$ as follows:

- $\mathsf{tp} = \{|\omega_{\mathbf{X}} \in \Omega_{\mathbf{X}} \mid \varphi_\omega \in \mathsf{D}', \varphi_\omega \models \mathbf{y}, \kappa'(\omega_{\mathbf{X}}) \leq \tau|\}$
- $\mathsf{fp} = \{|\omega_{\mathbf{X}} \in \Omega_{\mathbf{X}} \mid \varphi_\omega \in \mathsf{D}', \varphi_\omega \models \neg\mathbf{y}, \kappa'(\omega_{\mathbf{X}}) \leq \tau|\}$
- $\mathsf{tn} = \{|\omega_{\mathbf{X}} \in \Omega_{\mathbf{X}} \mid \varphi_\omega \in \mathsf{D}', \varphi_\omega \models \neg\mathbf{y}, \kappa'(\omega_{\mathbf{X}}) > \tau|\}$
- $\mathsf{fn} = \{|\omega_{\mathbf{X}} \in \Omega_{\mathbf{X}} \mid \varphi_\omega \in \mathsf{D}', \varphi_\omega \models \mathbf{y}, \kappa'(\omega_{\mathbf{X}}) > \tau|\}$

The final threshold τ' is selected as the median among all integer thresholds $\tau \in [0, up]$ having the maximum score $\mathbf{m}(\mathsf{cm}(\tau))$, where $up = \arg\max(\{\kappa'(\omega_{\mathbf{X}}) \mid \omega_{\mathbf{X}} \in \Omega_{\mathbf{X}}\})$:[3]

$$\tau' = median(\arg\max(\{\mathbf{m}(\mathsf{cm}(\tau)) \mid \tau \in \mathbb{N}\}))$$

[3]It suffices to search in $[0, up]$: thresholds $\tau < 0$ yield the same confusion matrix as $\tau = 0$, and $\tau > up$ as $\tau = up$.

Having characterized both the new OCF κ' and the threshold τ', we now summarize the complete definition of the improvement-based learning operator $\odot_{(\circ,B,\mathbf{m})}$:

Definition 7 (Improvement-based learning operator $\odot_{(\circ,B,\mathbf{m})}$)**.** *Let $\circ$ be an improvement operator, B be a neighborhood, and $\mathbf{m}$ be a performance metric. The* improvement-based learning operator *induced by $\circ$, B, and $\mathbf{m}$, and denoted by $\odot_{(\circ,B,\mathbf{m})}$, is defined on the* TOCF *classifier space $\mathcal{E}_{tocf}$ as follows. For each binary classifier $(\mathsf{D},\kappa,\tau) \in E_{tocf}$ and each training instance $\varphi_\omega \in \mathcal{L}^c$, we define $(\mathsf{D},\kappa,\tau) \odot_{(\circ,B,\mathbf{m})} \varphi_\omega = (\mathsf{D}',\kappa',\tau')$, where:*

$$\begin{cases} \mathsf{D}' = \mathsf{D} \sqcup (\varphi_\omega) \\ \kappa' = \kappa \bullet \varphi_\omega \\ \tau' = \arg\max(\{\mathbf{m}(\mathsf{cm}(\tau)) \mid \tau \in \mathbb{N}\}) \end{cases}$$

To define the "full" learning framework (cf. Def. 5), we must specify an anchor serving as an initial state. We choose $\Psi_*^\top = (\emptyset, \kappa_\top, 0)$, where $\kappa_\top$ is the constant OCF assigning 0 to every world $\omega_\mathbf{X} \in \Omega_\mathbf{X}$. Hence, the full improvement-based learning framework is given by the pair $(\Psi_*^\top, \odot_{(\circ,B,\mathbf{m})})$.

5 An Instantiated Learning Operator

We now present a concrete instantiation of an improvement-based learning operator, $\odot_{(\circ_+,B^H,\mathbf{m}_{ba})}$. This operator is fully defined by the following choices:

- $\circ_+$, the basic shifting improvement operator (cf. Sec. 2)
- B^H, that defines pairs of worlds as direct neighbors when they differ on at most one variable, i.e., for each $\omega_\mathbf{X} \in \Omega_\mathbf{X}$:
$$B^H_{\omega_\mathbf{X}} = \{\omega'_\mathbf{X} \mid |\{x_i \in P_\mathbf{X} \mid \omega'_\mathbf{X}(x_i) \neq \omega_\mathbf{X}(x_i)\}| \leq 1\}$$
- $\mathbf{m}_{ba}$, the *balanced accuracy* metric, defined for each confusion matrix $(\mathsf{tp},\mathsf{fp},\mathsf{tn},\mathsf{fn})$ as:
$$\mathbf{m}_{ba}(\mathsf{tp},\mathsf{fp},\mathsf{tn},\mathsf{fn}) = 1/2(\mathsf{tp}/(\mathsf{tp}+\mathsf{fn}) + \mathsf{tn}/(\mathsf{tn}+\mathsf{fp}))$$

The choice of balanced accuracy is illustrative rather than principled. Nonetheless, it is particularly appropriate for imbalanced datasets, where one class outweighs the other. By giving equal importance to the true positive rate and the true negative rate, balanced accuracy ensures a more equitable evaluation of classifier performance across both classes.

Representation of $(\Psi_*^\top, \odot_{(\circ_+,B^H,\mathbf{m}_{ba})})$. We now show that, in this specific learning framework, both a training algorithm and an inference algorithm (predicting the label of any instance $\omega_\mathbf{X} \in \Omega_\mathbf{X}$ from a trained classifier) can be designed. Both algorithms run in polynomial time with respect to the number of training instances and the number of features.

Given an instance description $\omega_\mathbf{X} \in \Omega_\mathbf{X}$, let $\overline{\omega_\mathbf{X}} \in \Omega_\mathbf{X}$ be defined as $\overline{\omega_\mathbf{X}}(x_i) = 1 - \omega_\mathbf{X}(x_i)$, for each $x_i \in P_\mathbf{X}$. Let $d_H : \Omega_\mathbf{X} \times \Omega_\mathbf{X} \to \mathbb{N}$ be the Hamming distance between instance descriptions, i.e., for all $\omega_\mathbf{X}, \omega'_\mathbf{X} \in \Omega_\mathbf{X}$,

$$d_H(\omega_\mathbf{X}, \omega'_\mathbf{X}) = |\{x_i \in P_\mathbf{X} \mid \omega_\mathbf{X}(x_i) \neq \omega'_\mathbf{X}(x_i)\}|.$$

We extend the Hamming distance to define a distance between any instance description $\omega'_\mathbf{X}$ and dataset D as $d_H(\omega'_\mathbf{X}, \emptyset) = 0$, and if $\mathsf{D} \neq \emptyset$:

$$d_H(\omega'_\mathbf{X}, \mathsf{D}) = \sum\{d_H(\omega'_\mathbf{X}, \varphi_\omega) \mid \varphi_\omega \in \mathsf{D}\}, \text{ with}$$

$$d_H(\omega'_\mathbf{X}, \varphi_\omega) = \begin{cases} d_H(\omega'_\mathbf{X}, \omega_\mathbf{X}) & \text{if } \varphi_\omega \models \mathbf{y} \\ d_H(\omega'_\mathbf{X}, \overline{\omega_\mathbf{X}}) & \text{if } \varphi_\omega \models \neg\mathbf{y} \end{cases}$$

On the other hand, given any dataset D, let $\omega^\mathsf{D}_\mathbf{X} \in \Omega_\mathbf{X}$ be any instance description such that for each $x_i \in P_\mathbf{X}$:

$$\omega^\mathsf{D}_\mathbf{X}(x_i) = \begin{cases} 1 & \text{if } \mathsf{D}^1(x_i) \geq \mathsf{D}^0(x_i) \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathsf{D}^1(x_i) = |\{\varphi_\omega \in \mathsf{D} \mid \varphi_\omega \models \mathbf{y} \text{ iff } \omega(x_i) = 1\}|$ and $\mathsf{D}^0(x_i) = m - \mathsf{D}^1(x_i)$ (recall that m is the number of training instances from D.)

We can show that such an instance description ω^D_X is one of the "closest" instance description to the dataset D in terms of Hamming distance:[4]

Lemma 1. *For each dataset $\mathsf{D} \in \mathcal{D}$ and each instance description $\omega_\mathbf{X} \in \Omega_\mathbf{X}$, we have $d_H(\omega^\mathsf{D}_\mathbf{X}, \mathsf{D}) \leq d_H(\omega_\mathbf{X}, \mathsf{D})$.*

Then, we can show that for each instance description $\omega_\mathbf{X} \in \Omega_\mathbf{X}$ the value $\kappa(\omega_\mathbf{X})$ can always be characterized via computations of Hamming distances to the currently trained dataset:

Proposition 1. *For each* TOCF *$\Psi = (\mathsf{D},\kappa,\tau)$ and each instance description $\omega_\mathbf{X} \in \Omega_\mathbf{X}$, we have that:*

$$\kappa(\omega_\mathbf{X}) = d_H(\omega_\mathbf{X}, \mathsf{D}) - d_H(\omega^\mathsf{D}_\mathbf{X}, \mathsf{D})$$

This has several important implications for our learning framework $(\Psi_*^\top, \odot_{(\circ_+,B^H,\mathbf{m}_{ba})})$ in terms of computational complexity. Recalling that n is the size of the feature set and m is the number of training instances from D, we have:

Proposition 2. *Let $\mathsf{D} \in \mathcal{D}$, and let $\Psi_*^\top \odot_{(\circ_+,B^H,\mathbf{m}_{ba})} \mathsf{D} = \Psi = (\mathsf{D},\kappa,\tau)$.*

1. *τ can be computed in time $\mathcal{O}(n \cdot m^2)$*
2. *given (D,τ), for each $\omega_\mathbf{X} \in \Omega_\mathbf{X}$, deciding whether $\omega_\mathbf{X} \in [Pos(\Psi)]$ can be done in time $\mathcal{O}(n \cdot m)$*

As a consequence of Proposition 2, in the learning framework $(\Psi_*^\top, \odot_{(\circ_+,B^H,\mathbf{m}_{ba})})$, we obtain both a learning algorithm (the computation of τ) and an inference algorithm (the prediction for any instance $\omega_\mathbf{X} \in \Omega_\mathbf{X}$ given (D,τ)) that run in polynomial time with respect to the number of training instances and the number of features (Proposition 2.1 and 2.2, respectively). Since inference can be performed in polynomial time from (D,τ) alone, the learned classifier (D,κ,τ) is fully characterized by (D,τ), i.e., the OCF κ is not needed.

Another noteworthy property of this learning framework is its robustness to dataset permutations. Given any dataset $\mathsf{D} = (\varphi^s_\omega)_{1\leq s\leq m}$, let $\mathsf{D}^\pi = (\varphi^{\pi(s)}_\omega)_{1\leq s\leq m}$ denote the dataset obtained by applying a permutation $\pi : \{1,\ldots,m\} \to \{1,\ldots,m\}$:

Proposition 3. *For any permutation $\pi : \{1,\ldots,m\} \to \{1,\ldots,m\}$ and any dataset $\mathsf{D} \in \mathcal{D}$,*

$$\Psi_*^\top \odot_{(\circ_+,B^H,\mathbf{m}_{ba})} \mathsf{D} = \Psi_*^\top \odot_{(\circ_+,B^H,\mathbf{m}_{ba})} \mathsf{D}^\pi$$

[4] All proofs are available online [Schwind *et al.*, 2025].

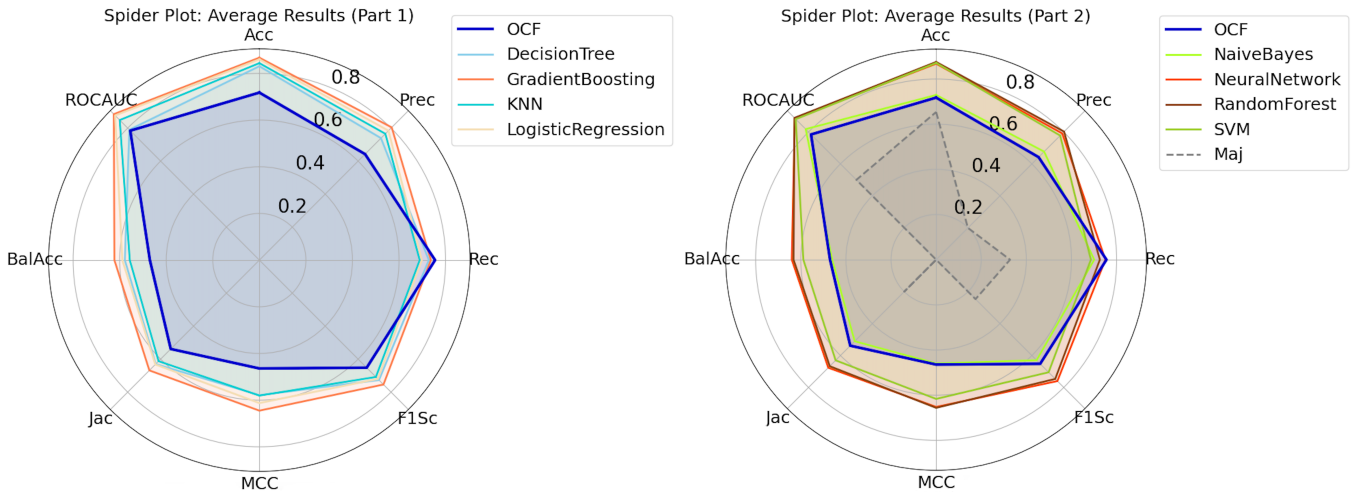

Figure 2: A comparison of performance between all learning models across all datasets.

In other words, the learned classifier is invariant under reordering of the training instances. This is significant, as many standard learning models, such as decision trees, random forests, and k-nearest neighbors (k-NN), may produce different outputs depending on the order in which the data is processed. This corresponds to the commutativity postulate for improvement operators [Schwind and Konieczny, 2020], which states that no piece of information should be treated as more important than another when the order of input carries no meaningful priority.

6 Experiments

Empirical protocol. We implemented in Python the improvement-based learning model $\odot_{(\circ_{+1}, B^H, \mathbf{m}_{ba})}$ (simply denoted by $\odot$ onward). Its predictive performance was compared against the following standard ML models (learned using the scikit-learn library [Pedregosa *et al.*, 2011] and considering default parameters): Logistic Regression, Naive Bayes, Decision Tree, Gradient Boosting, KNN, Neural Network, Random Forest, and SVM. In addition, a trivial learning model (named Maj) always predicting the majority class (positive or negative) in the training dataset was considered as a baseline.

The evaluation considered a large set of performance metrics that are standard in ML: balanced accuracy, accuracy, F1 score, Jaccard index, Matthews correlation coefficient (MCC), precision, recall, and ROC-AUC (receiver operating characteristic area under the curve).

The experimental protocol involved selecting 58 binary classification datasets from the UCI repository,[5] with each dataset containing up to 12,684 instances and up to 1,203 numerical or categorical features. A 10-fold cross-validation has been conducted: each dataset was split into ten random samplings, with a 90%/10% division for training and test sets. Missing values in the data were imputed by filling each numerical feature with the mean value from the dataset and each categorical feature with the most frequent value. The datasets were then standardized using a standard scaler.

For our learning framework $\odot$, numerical attributes were further re-scaled linearly to the interval $[0, 10]$ with integer values, while categorical attributes were left unchanged. For these experiments, binarizing the features was not mandatory: instead, we used a modified weighted Hamming distance between instance descriptions $d'_H(\omega_{\mathbf{X}}, \omega'_{\mathbf{X}})$. This was made only for convenience as this is equivalent in terms of performance and computational complexity to binarizing all datasets in such a way to ensure that $d_H = d'_H$.

Empirical results. We did an extensive comparative analysis which is visualized through various plots (the full set is available in [Schwind *et al.*, 2025]). Box plots were generated for each metric, showing model performance across the 58 datasets, scatter plots compared the proposed model with each baseline, and spider plots have been used to compare the performance of all models for individual datasets and in an aggregate view. We present some of them hereafter.

The spider plots in Figure 2 show in a synthetic way the average predictive performances (over the 58 datasets) of the ten ML models learned for each of the ten families of models considered in our experiments and assessed for each of the eight metrics (our learning framework is simply named OCF in the figures). One can see that $\odot$ outperforms the baseline Maj model (which is mandatory to be considered as a significant learning operator). It turns out that in practice $\odot$ achieves performances that are similar to the ones of Naive Bayes. Finally, $\odot$ performs slightly better than the other models when recall is used to measure the predictive performance. This is an important point since high recall is expected in applications where missing a positive case can be costly, such as in healthcare or when dealing with attack detection.

Focusing on the balanced accuracy metric, Figure 3 (left) presents a scatter plot for contrasting in a more precise way the predictive performances of $\odot$ and Naive Bayes; and Figure 3 (right) presents boxplots representing the distributions of predictive performance achieved over all datasets by each

[5] https://archive.ics.uci.edu/datasets/

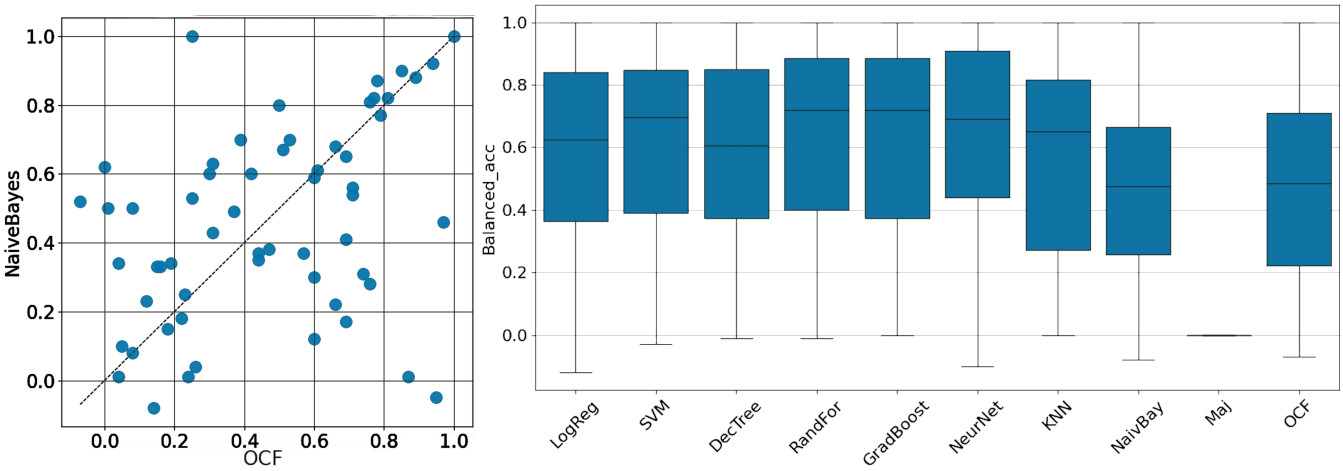

Figure 3: A comparison between our learning model and the Naive Bayes learning model across all datasets (left), and a comparison of performance between all learning models across all datasets (right), both focusing on the balanced accuracy performance metric.

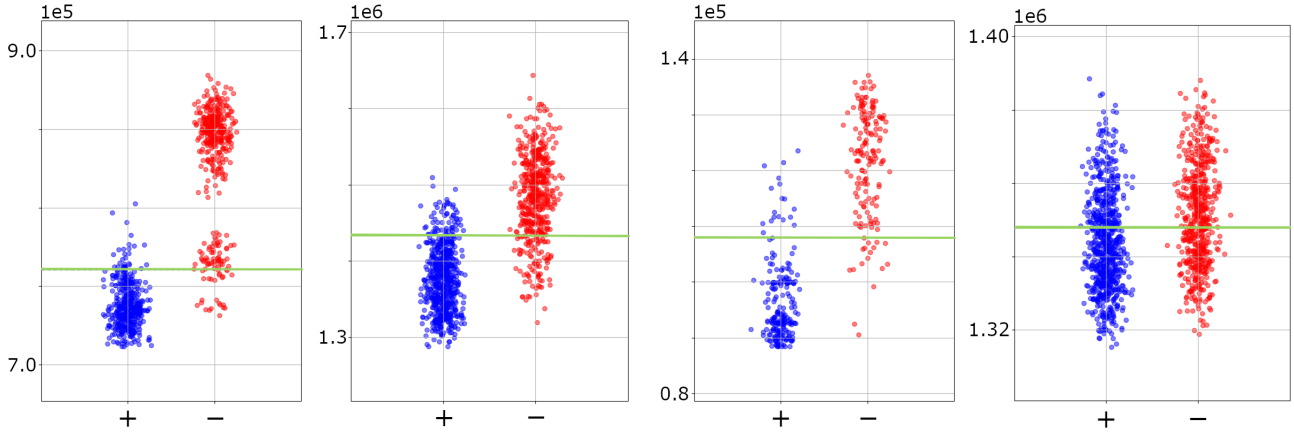

Figure 4: Examples of TOCFS (OCF and threshold) showing how positive and negative test instances are separated, on samplings randomly chosen from four datasets (from left to right: 73-Mushroom, 327-Phishing Websites, 545-Rice, 603-In-Vehicle Coupon Recommendation).

of the ten ML models at hand. The results presented are coherent with those shown in Figure 2 (especially, the averaged balanced accuracy of $\odot$ is slightly greater than the one of Naive Bayes and much better than the one of Maj).

Additionally, Figure 4 showcases some learned classifers $(\mathsf{D}, \kappa, \tau)$ for some test sets in some randomly chosen dataset samplings. Each figure depicts the value distribution of all $\kappa(\omega_{\mathbf{X}})$ for each test instance $\omega_{\mathbf{X}}$ from the sampling, represented as a blue (resp. red) dot when $\varphi_{\omega_{\mathbf{X}}} \models \mathbf{y}$ (resp. $\varphi_{\omega_{\mathbf{X}}} \models \neg\mathbf{y}$), and the green horizontal line corresponds to the threshold τ. It can be observed that the learned model separates quite accurately the positive instances from the negative ones for the first three samples, but clearly not for the last one (603-In-Vehicle Coupon Recommendation).

7 Conclusion

In this paper, we showed how improvement operators, grounded in Belief Change Theory, can give rise to a family of online learning models, and we have presented learning and inference algorithms designed for this family. We empirically evaluated the predictive performance of a specific model within this family using a range of datasets. While the results indicate that this simple model does not match the accuracy of more advanced models (e.g., neural nets), it performs comparably to all other methods in terms of recall. This makes it particularly suitable for applications where failing to detect positives is critical. Additionally, the model outperforms the trivial Majority class model and shows performance on par with Naive Bayes, suggesting a significant potential.

This work opens up several avenues for future research. Theoretically, the next step is to analyze the properties of learning operators derived from improvement operators. Empirically, we plan to explore and evaluate additional models within the proposed family. For instance, while the learning algorithm used in the experiments is based on Hamming distance, other distances, such as the Mahalanobis distance, could be explored [Mahalanobis, 1936]. Because it accounts for feature correlations which could be measured on the training set, the Mahalanobis distance appears as a valuable candidate for improving model performance at inference.

Acknowledgements

This work was supported in part by the following grants: JSPS KAKENHI Grant Numbers JP25K00375 and JP25K03190; JST CREST Grant Number JPMJCR22D3; the AI Chairs BE4musIA (ANR-20-CHIA-0028) and EXPEKCTATION (ANR-19-CHIA-0005-01) funded by the French National Research Agency.

References

[Alchourrón *et al.*, 1985] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[Baltag *et al.*, 2011] Alexandru Baltag, Nina Gierasimczuk, and Sonja Smets. Belief revision as a truth-tracking process. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'11)*, pages 187—190, 2011.

[Baltag *et al.*, 2019] Alexandru Baltag, Nina Gierasimczuk, and Sonja Smets. Truth-tracking by belief revision. *Studia Logica*, 107(5):917–947, 2019.

[Bloch and Lang, 2002] Isabelle Bloch and Jérôme Lang. *Towards Mathematical Morpho-Logics*, pages 367–380. Physica-Verlag HD, 2002.

[Booth and Meyer, 2006] Richard Booth and Thomas A. Meyer. Admissible and restrained revision. *Journal of Artificial Intelligence Research*, 26:127–151, 2006.

[Choi *et al.*, 2009] Seung-Seok Choi, Sung-Hyuk Cha, and Charles Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 11 2009.

[Darwiche and Pearl, 1997] Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29, 1997.

[Gärdenfors, 1988] Peter Gärdenfors. *Knowledge in flux*. MIT Press, 1988.

[Goldsmith *et al.*, 2004] Judy Goldsmith, Robert H. Sloan, Balázs Szörényi, and György Turán. Theory revision with queries: Horn, read-once, and parity formulas. *Artificial Intelligence*, 156(2):139–176, 2004.

[Jin and Thielscher, 2007] Yi Jin and Michael Thielscher. Iterated belief revision, revised. *Artificial Intelligence*, 171(1):1–18, 2007.

[Katsuno and Mendelzon, 1991] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

[Kelly, 1998] Kevin T. Kelly. The learning power of belief revision. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'98)*, pages 111–124, 1998.

[Kelly, 2014] Kevin T. Kelly. *A Computational Learning Semantics for Inductive Empirical Knowledge*, pages 289–337. Springer International Publishing, 2014.

[Konieczny and Pino Pérez, 2008] Sébastien Konieczny and Ramon Pino Pérez. Improvement operators. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 177–187, 2008.

[Konieczny *et al.*, 2010] Sébastien Konieczny, Mattia Medina Grespan, and Ramon Pino Pérez. Taxonomy of improvement operators and the problem of minimal change. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR'10)*, pages 161–170, 2010.

[Mahalanobis, 1936] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55, 1936.

[Medina Grespan and Pino Pérez, 2013] Mattia Medina Grespan and Ramón Pino Pérez. Representation of basic improvement operators. In *Trends in Belief Revision and Argumentation Dynamics*, pages 195–227. College Publications, 2013.

[Pagnucco and Rajaratnam, 2005] Maurice Pagnucco and David Rajaratnam. Inverse resolution as belief change. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 540–545, 2005.

[Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Mathieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Sauerwald and Beierle, 2019] Kai Sauerwald and Christoph Beierle. Decrement operators in belief change. In *Proceedings of the 15th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'19)*, pages 251–262, 2019.

[Schwind and Konieczny, 2020] Nicolas Schwind and Sébastien Konieczny. Non-prioritized iterated revision: Improvement via incremental belief merging. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR'20)*, pages 738–747, 2020.

[Schwind *et al.*, 2022] Nicolas Schwind, Sébastien Konieczny, and Ramón Pino Pérez. On the representation of Darwiche and Pearl's epistemic states for iterated belief revision. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR'22)*, 2022.

[Schwind *et al.*, 2023] Nicolas Schwind, Katsumi Inoue, and Pierre Marquis. Editing boolean classifiers: A belief change perspective. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI'23)*, pages 6516–6524, 2023.

[Schwind *et al.*, 2025] Nicolas Schwind, Katsumi Inoue, Sébastien Konieczny, and Pierre Marquis. Iterated belief

change as learning: Supplementary material. https://github.com/nicolas-schwind/IteratedBeliefChangeML, 2025.

[Shalev-Shwartz and Ben-David, 2014] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[Spohn, 1988] Wolfgang Spohn. *Ordinal Conditional Functions: A Dynamic Theory of Epistemic States*, pages 105–134. Springer Netherlands, 1988.

[Williams, 1995] Mary-Anne Williams. Iterated theory base change: A computational model. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1541–1549, 1995.

[Wrobel, 1994] Stefan Wrobel. *Concept formation and knowledge revision*. Kluwer Academic, Netherlands, 1994.

[Ye *et al.*, 2016] Han-Jia Ye, De-Chuan Zhan, and Yuan Jiang. Instance specific metric subspace learning: A bayesian approach. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 2272–2278, 2016.