Controlling False Positives in Multiple Instance Learning: the "c-rule" Approach

Rosario Delgado[*]

Department of Mathematics, Universitat Autònoma de Barcelona, Spain

Abstract

This paper introduces a novel strategy for labeling bags in binary Multiple Instance Learning (MIL) under the *standard MI assumption*. The proposed approach addresses errors in instance labeling by classifying a bag as positive if it contains at least c positively labeled instances. This strategy seeks to balance the trade-off between controlling the *false positive rate* (mislabeling a negative bag as positive) and the *false negative rate* (mislabeling a positive bag as negative) while reducing labeling efforts.

The study provides theoretical justifications for this approach and introduces algorithms for its implementation, including determining the minimum value of c required to keep error rates below predefined thresholds. Additionally, it proposes a methodology to estimate the number of genuinely positive and negative instances within bags. Simulations demonstrate the superior performance of the "c-rule" compared to the *standard* rule (corresponding to $c = 1$) in scenarios with sparse positive bags and moderately low to high probability of misclassifying a negative instance. This trend is further validated through comparisons using two real-world datasets. Overall, this research advances the understanding of error management in MIL and provides practical tools for real-world applications.

Keywords: Multiple Instance Learning (MIL), Standard MI assumption, False positive rate, False negative rate

1. Introduction

Multiple Instance Learning (MIL) is a machine learning paradigm where data is organized into bags, each containing some instances. Introduced by [1] for drug activity prediction (see [2, 3] for contextualization, and references therein), MIL differs from traditional supervised learning by assigning labels

[*] Author at: Department of Mathematics. Universitat Autònoma de Barcelona. Edifici C- Campus de la UAB. Av. de l'Eix Central s/n. 08193 Bellaterra (Cerdanyola del Vallès), Spain. ORCID: 0000-0003-1208-9236

Email address: Rosario.Delgado@uab.cat (Rosario Delgado)

at both the bag and instance levels. Under the *standard MI assumption*, a negative bag contains only negative instances, while a positive bag contains at least one positive instance. This assumption is particularly useful when objects are described by a set of parts, where each instance contains partial information necessary for classification.

MIL finds practical applications in areas such as medical imaging, where obtaining precise annotations can be time-consuming and subjective. For example, in tumor detection, a medical image (bag) may contain patches (instances), and a positive image is labeled as such if at least one patch is positive, reducing the need for exhaustive manual annotations. Beyond medical imaging, MIL is also employed in computer vision, remote sensing, text mining and drug discovery, among other fields.

While the *standard MI assumption* has been effective across various domains, alternative assumptions have been explored. For example, [2] allows other different interactions between instance labels and the label assigned to the bag they belong to, and [4, 5], have proposed alternative premises in a heterodox perspective of MIL. The *unanimity MI assumption* ([3]) classifies bags by considering both the labels and the confidence levels of its constituent instances. This approach employs a rule typically used in constructing *ensembles* of probabilistic classifiers, and is applied to scenarios such as bags representing victims from the same homicide case. The *collective MI assumption* ([6]) requires multiple positive instances for a bag to be classified as positive, as in traffic jam detection from images.

In this paper, we focus on a different challenge within the MIL framework under the *standard MI assumption*: understanding how classification errors at the instance level impact the bag-level labeling process. Specifically, we propose a strategy where a bag is labeled positive if it contains at least c positively labeled instances, mitigating the effects of instance misclassification and providing more robust bag-level predictions. This method, referred to as the "c-rule", forms the core of this investigation.

Labeling at the instance-level

Classifiers trained to label instances within bags are prone to errors, which can negatively impact bag-level predictions. This issue becomes especially problematic in large-scale task such as autonomous driving, robotic control or medical imaging diagnostics (see [7]). We introduce the classifier's specifications p and q to denote the probabilities of misclassification of positive and negative instances, respectively.

Under the *standard MI assumption*, a bag is labeled positive if at least one instance is labeled positive. However, errors in instance labeling complicate this rule, increasing the likelihood of bag misclassification. To tackle this issue, several approaches have been proposed in the literature, including: (1) robust learning algorithms resilient to mislabeling. This can involve using algorithms with robust loss functions or regularization techniques ([8, 9, 10]), assigning trust scores to training samples for weighting training ([11, 7]), or apply boosting-based classifiers leveraging MIL principle [12]. (2) active learning strategies to query highly uncertain instances ([13]). (3) noise filtering techniques to detect and remove mislabeled data ([14]). Some methods focus on cleaning misclassified instances, such as the *confident learning* approach by [15], or on constructing bag-level features from instances likely to be positive, as seen in [16]. Label errors in test datasets have received less attention, despite having a distinct set of potential consequences ([17]). In [18], the authors apply Label Error Detection (LED) to token classification tasks, proposing a method to score sentences based on the likelihood of containing mislabeled tokens, for label review.

In contrast, our work introduces a simpler method within the *standard MI assumption* framework. We enhance the traditional rule by proposing a strategy that accounts for possible mislabelled instances without modifying the instance-level classifier or dataset. Instead, we rely on estimated probabilities of instance incorrect labeling, allowing for more robust bag-level labeling.

Labeling at the bag-level

The challenge of mislabeled instances within bags, which can lead to incorrect bag-level labels, has recently gained research attention. Notably, [14] has pioneered efforts in addressing this issue. In this context, two types of errors can occur during bag labeling:

False positive: labeling a truly negative bag as positive.

False negative: labeling a truly positive bag as negative.

Controlling for false positive rates is crucial in many real-world applications. For instance, in spam filtering misclassifying important emails as spam can lead to missed critical communications. In medical imaging, false positives can have significant consequences, leading to unnecessary patient anxiety and procedures ([19]). For example, breast cancer screening, including MRI (magnetic resonance imaging) and mammography, often results in false positives, with up to 50% of women experiencing at least one incorrect positive recall over ten years of screening ([20, 21, 22]). Similarly, PET/CT (Positron Emission and Computerized Tomography) scans, crucial for cancer diagnosis, report false positive rates of around 13% ([23]). Even in AI applications in radiology, such as chest radiographs, false positives for certain conditions exceed human interpretations (see [24]). Effectively managing false positive rates is essential for ensuring accurate diagnoses and optimizing patient care, despite technological advancements.

Paper Contributions and Organization

This study presents a novel approach for labeling bags under the *standard MI assumption* in the presence of mislabeled instances. We introduce the **"c-rule"**, which labels a bag as positive if it contains at least $c \geq 1$ positive instances. Adjusting the value of c allows for controlling both the *false positive* and *false negative rates*, by setting thresholds α and β, respectively, while simultaneously reducing labeling efforts. Our method builds on the *standard rule* (where $c = 1$), balancing error control and efficiency. Key findings include:

- False positive rates decrease as c increases, as shown in Proposition 1.
- False negative rates exhibit non-monotonic behavior, initially increasing and then decreasing with higher c (Lemma 2).
- This behavior enables the design of the "c-rule" strategy to control both rates when labeling bags subject to instance-level errors in classification (Proposition 2, Algorithm 1).
- Estimating the actual number of positive and negative instances within a bag is needed due to misclassification at the instance level (Algorithm 2).

Figure 1 outlines the process for implementing the methodology.

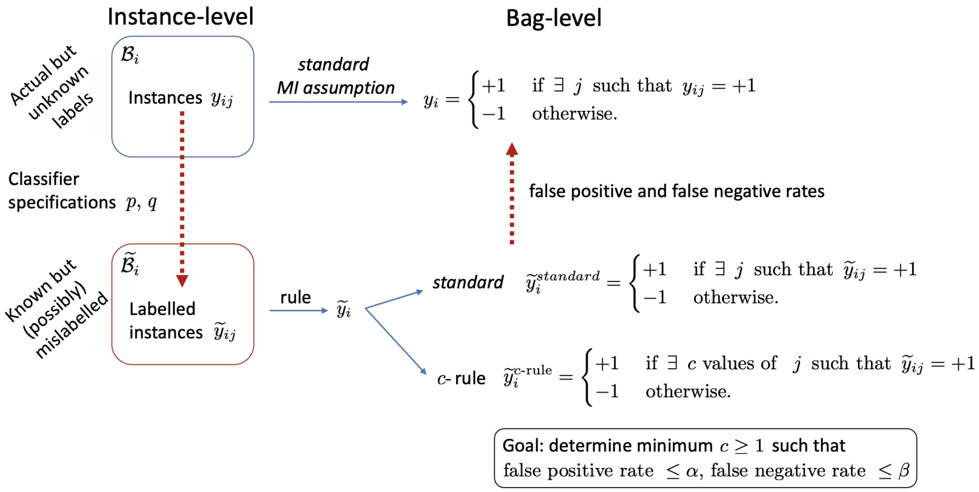

Instance-level
Bag-level
Actual but unknown labels
$\mathcal{B}_i$
Instances y_{ij}
standard MI assumption
$y_i = \begin{cases} +1 & \text{if } \exists\ j \text{ such that } y_{ij} = +1 \\ -1 & \text{otherwise.} \end{cases}$
Classifier specifications p, q
false positive and false negative rates
Known but (possibly) mislabelled
$\widetilde{\mathcal{B}}_i$
Labelled instances $\widetilde{y}_{ij}$
rule
$\widetilde{y}_i$
standard
$\widetilde{y}_i^{standard} = \begin{cases} +1 & \text{if } \exists\ j \text{ such that } \widetilde{y}_{ij} = +1 \\ -1 & \text{otherwise.} \end{cases}$
c- rule
$\widetilde{y}_i^{c\text{-rule}} = \begin{cases} +1 & \text{if } \exists\ c \text{ values of } j \text{ such that } \widetilde{y}_{ij} = +1 \\ -1 & \text{otherwise.} \end{cases}$
Goal: determine minimum $c \geq 1$ such that false positive rate $\leq \alpha$, false negative rate $\leq \beta$

Figure 1: Illustration of the procedural framework.

The false positive and false negative rates in bag labeling depend on the bag's composition, the classifier's specifications p and q, which are the probabilities of misclassification at the instance level, and the choice of c. This approach is particularly useful in scenarios with sparse positive instances, such as image-region classification ([8, 25, 26]).

As an illustration, consider a bag with 100 instances, a classifier with specifications $p = q = 0.01$, and a target error threshold of 0.01. Under the standard rule, the false positive rate is 0.63. However, using $c = 10$, this rate drops to 7.6×10^{-8}, with a false negative rate of 0.0024. We used 5 positive and 95 negative instances in the bag to calculate the false negative rate.

The paper is structured as follows: Section 2 introduces the "c-rule" and establish its effectiveness in controlling error rates at some fixed thresholds. We also present Algorithm 1 for determining the optimal c value. In Section

3, we describe a methodology for estimating the number of positive and negative instances in bags, along with Algorithm 2 for practical implementation. Subsection 4.1 provides simulations comparing the *standard rule* with the "c-rule" in two scenarios, showing the latter's superior performance under specific conditions. Subsections 4.2 and 4.3 extend this comparison to the real-world **Musk-2** and **Colon cancer** datasets using the `milr` and `CausalMIL` algorithms, respectively. Finally, Section 5 offers some closing remarks.

2. The "c-rule" strategy

Formally, let $\mathcal{X}$ represent the instance space, $\mathcal{Y} = \{-1, +1\}$ denote the label space, and $\mathcal{B}$ denote the set of bags. Each bag $B_i \in \mathcal{B}$ is a collection of multiple instances from $\mathcal{X}$, with the number of instances, denoted as M_i, varying between bags. The dataset is represented as $\{(B_i, y_i)\}_{i=1}^{N}$, where each $y_i \in \mathcal{Y}$ and N is the total number of bags. In MIL, the objective is to learn a function $h : \mathcal{B} \longrightarrow \mathcal{Y}$ from the training data. This is typically achieved by using a base classifier at the instance-level, denoted as $\mathcal{C} : \mathcal{X} \longrightarrow \mathcal{Y}$, which assigns a label to each instance within the bag, and then establishing a rule for assigning a label to the bag based on the instance labels, denoted as $\{y_{ij}\}_{j=1}^{M_i}$ for bag B_i. The bag-level label hinges on the labels assigned to the instances within the bag.

The *standard* rule, as previously explained, involves assigning the positive label to the bag if at least one instance carries a positive label; otherwise, a negative label is assigned. That is,

$$y_i = \begin{cases} +1 & \text{if } \exists \ j \text{ such that } \ y_{ij} = +1 \\ -1 & \text{otherwise.} \end{cases} \tag{1}$$

Let $\widetilde{y}_{ij}$ denote the assigned label for the j-th instance in bag B_i, and $\widetilde{y}_i$ represent the label assigned to the bag based on $\{\widetilde{y}_{ij}\}_{j=1}^{M_i}$ using a specified rule (*standard* or otherwise). With these notations, we can express the specifications of the classifier as:

$$\begin{aligned} p &= P(\widetilde{y}_{ij} = -1 \,/\, y_{ij} = +1) \quad (= 1 - \text{sensitivity})\,, \\ q &= P(\widetilde{y}_{ij} = +1 \,/\, y_{ij} = -1) \quad (= 1 - \text{specificity})\,. \end{aligned}$$

and the probabilities of misclassification at the bag level as:

$$\begin{aligned} \text{false positive rate } &= P(\widetilde{y}_i = +1 \,/\, y_i = -1)\,, \\ \text{false negative rate } &= P(\widetilde{y}_i = -1 \,/\, y_i = +1)\,. \end{aligned}$$

The operational procedure for using the "c-rule" entails the following steps for labeling a bag of size $M \geq 1$, given two thresholds, $\alpha,\, \beta \in (0,\, 1)$, to govern the acceptable false positive and false negative rates, respectively:

i) Acquire the specifications of classifier $\mathcal{C}$, p, $q \in [0,\, 0.5)$.

ii) Find the minimum value of c for the "c-rule", c^*, ensuring that the false positive and false negative rates, which are functions of c, denoted respectively by φ_+ and φ_-, satisfy the following conditions:

$$\varphi_+(c) \leq \alpha \quad \text{and} \quad \varphi_-(c) \leq \beta\,.$$

The process of identifying the single value c^* that satisfies these constraints will be carried out using Algorithm 1, which is based on the results established in this section.

iii) Assign the label $\widetilde{y}_i^{c\text{-rule}}$ to the bag, where

$$\widetilde{y}_i^{c\text{-rule}} = \begin{cases} +1 & \text{if } \exists \ c^* \text{ values of } \ j \ \text{ such that } \ \widetilde{y}_{ij} = +1 \\ -1 & \text{otherwise.} \end{cases}$$

The objective is to ensure that both error rates adhere to their respective upper thresholds. Our approach involves determining the minimum value of c, c_1, such that for any $c \geq c_1$, $\varphi_+(c) \leq \alpha$. Subsequently, we seek the minimum value of $c \geq c_1$, c^*, such that $\varphi_-(c) \leq \beta$. The theoretical underpinnings of this approach, along with certain necessary and sufficient conditions ensuring it is meaningful, will be further explored.

Proposition 1 addresses the false positive rate φ_+, i.e., the probability that a bag containing no genuinely positive instances is labeled positive by the "c-rule". The false negative rate φ_-, representing the probability of a bag containing at least one genuinely positive instance being labeled negative will be tackled in Proposition 2. Here, we utilize the notations m_+ and m_- to denote the instances genuinely positive and negative in the bag, respectively, with $M = m_+ + m_-$ the bag size, which are typically unknown in practical scenarios.

Proposition 1.

a) *$\varphi_+(c)$ is a decreasing function, for $1 \leq c \leq M$.*

b) *Fix $\alpha \in (0, 1)$. Consider the hypothesis*

$$(h_1) \qquad q^M \leq \alpha\,.$$

Then, (h_1) is a necessary and sufficient condition for the existence of a positive integer c_1, $1 \leq c_1 \leq M$, such that

$$\varphi_+(c) \leq \alpha \textit{ for all } c_1 \leq c \leq M\,.$$

Note that condition (h_1) is expressed in terms of the classifier's specification q, the bag size M, and the threshold α for the false positive rate. Notably, (h_1) is not overly restrictive since for a fixed q, the function q^M monotonically decreases as M increases, converging to 0.

Proof (Proposition 1). By definition, $\varphi_+(c)$ is the probability that a bag, with M instances actually negative, receives a positive label by the "c-rule", implying that at least c instances have been misclassified from negative to positive. Leveraging the Binomial distribution, as instances flip independently of each other and with the same misclassification rate q, we can express this probability as follows:

$$\varphi_+(c) = \sum_{\ell=c}^{M} \binom{M}{\ell} q^{\ell} \, (1-q)^{M-\ell} \,, \tag{2}$$

which is a decreasing function of c. Thus, statement a) has been proved.

As for statement b), $\varphi_+(c) \leq \alpha$ is equivalent by (2) to writing:

$$\sum_{\ell=c}^{M} \binom{M}{\ell} q^{\ell} \, (1-q)^{M-\ell} \leq \alpha \,.$$

A necessary and sufficient condition for the existence of c_1 such that for any $c \geq c_1$ it holds $\varphi_+(c) \leq \alpha$, is that the minimum value of this function, corresponding to $c = M$, not be greater than α:

$$\varphi_+(M) = \sum_{\ell=M}^{M} \binom{M}{\ell} q^{\ell} \, (1-q)^{M-\ell} = q^M \leq \alpha \,,$$

which is condition (h_1), finishing the proof. □

Remark 1. *If we opt to employ the standard rule, represented by $c = 1$, instead of using the "c-rule" with $c > 1$, expression* (2) *can be written as $1 - (1-q)^M$. Upon comparison of condition $1 - (1-q)^M \leq \alpha$ with (h_1), we observe that the condition derived for $c = 1$ is much more restrictive since $q^M \leq 1 - (1-q)^M$. Indeed, this is true due to the fact that*

$$q^M + (1-q)^M \leq \big(\, q + (1-q)\,\big)^M = 1 \,.$$

Proposition 2. *Fix $\beta \in (0,\, 1)$. Then,*

a) *Fix a positive integer c_1, $1 \leq c_1 \leq M$, and consider the hypothesis*

$$(h_{2.0}) \quad 1 - \sum_{j=0}^{M-c_1} \binom{M}{j} p^{j} \, (1-p)^{M-j} \leq \beta \,.$$

Then, $(h_{2.0})$ is a necessary and sufficient condition in the case $m_- = 0$ (then, $m_+ = M \geq 1$) for the existence of a positive integer c_2, $c_1 \leq c_2 \leq M$, such that

$$\varphi_-(c) \leq \beta \textit{ for all } c_1 \leq c \leq c_2 \,.$$

b) *Consider the hypothesis*

$$(h_{2.1}) \quad q^{m_- - 1}\Big(m_-(1-q) + q\big(1 - (1-p)^{m_+}\big)\Big) \leq \beta \,.$$

Then, $(h_{2.1})$ *is a necessary and sufficient condition in the case* $m_- \geq 1$ *and* $m_+ \geq 1$ *for the existence of a positive integer* c_2, $1 \leq c_2 \leq M$, *such that*

$$\varphi_-(c) \leq \beta \text{ for all } c_2 \leq c \leq M\,.$$

Remark 2. *Assumption* $(h_{2.0})$ *is expressed in terms of the classifier's specification* p, *the bag size* M, *and the fixed threshold* β *for the false negative rate. Besides, note that hypothesis* $(h_{2.1})$ *is not very restrictive. Indeed, since* $1-q \leq 1$ *and* $1-(1-p)^{m_+} \leq 1$, $q^{m_- - 1}\Big(m_-(1-q) + q\big(1-(1-p)^{m_+}\big)\Big) \leq q^{m_- - 1}\,(m_- + q)$ *and it is easy to see that* $f(x) = q^{x-1}\,(x+q)$ *is a decreasing function of* x *that drops very rapidly towards* 0 *as* x *increases.*

The proof of Proposition 2, provided in Appendix A, is derived from the following lemmas, which are also proved in Appendix A.

Lemma 1. *If* $m_+ \geq 1$, *we can express* $\varphi_-(c)$ *as follows:*

$$\varphi_-(c) = \sum_{i=(c-1-m_+)\vee 0}^{(c-1)\wedge m_-} \binom{m_-}{i} q^i (1-q)^{m_- - i} \Big(\sum_{j=i-(c-1-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big)$$

Lemma 2. *The following statements hold for function* $\varphi_-(c)$:

- *If* $m_- = 0$, $\varphi_-(c)$ *is an increasing function of* c, *for* $1 \leq c \leq M$.
- *If* $m_- \geq 1$ *and* $m_+ \geq 1$, *function* $\varphi_-(c)$ *has the following behavior:*

$$\text{a) if } 1 \leq c \leq m_+ + 1, \text{ then } \varphi_-(c) \text{ is an increasing function,}$$

$$\text{b) if } \frac{M + m_+ + 1}{2} \leq c \leq M, \text{ then } \varphi_-(c) \text{ is a decreasing function.}$$

Since m_+ and m_- are typically unknown in practical scenarios, we address the process of estimating these values in the subsequent section. A hat will be used to denote the estimates of the various quantities that depend on estimated m_+ and m_-. Fixed thresholds α and β, for the acceptable false positive and false negative rates, respectively, and the classifier's specifications p, $q \in [0,\, 0.5)$,we have that under the hypotheses

$$(h_1) \quad q^M \leq \alpha \quad \text{and}$$

$$(h_{2.0}) \quad 1 - \sum_{j=0}^{M-c_1} \binom{M}{j} p^j\,(1-p)^{M-j} \leq \beta\,, \quad \text{if } \widehat{m_-} = 0 \quad \text{or}$$

$$(h_{2.1}) \quad q^{\widehat{m_-}-1}\left(\widehat{m_-}\,(1-q) + q\,(1-(1-p)^{\widehat{m_+}})\right) \leq \beta\,, \quad \text{if } \widehat{m_-} \geq 1\,,$$

the problem of determining the estimated optimal value of c with the range of 1 to M for the "c-rule", denoted as c^*, which is the minimum value of c satisfying the condition

$$\varphi_+(c) \leq \alpha \quad \text{and} \quad \widehat{\varphi_-}(c) \leq \beta\,,$$

has a feasible solution and the algorithm to obtain it consists in the steps detailed in Algorithm 1. It is important to note that the false positive rate does not depend on m_+ and m_-, hence its value is not an estimation derived from $\widehat{m_+}$ and $\widehat{m_-}$.

Algorithm 1 `Determining Optimal` c `for the "`c`-rule"`

Input:

- Bag size $M \geq 1$
- Classifier's specifications in [0, 0.5): $p\,(1-\text{sensitivity})$ and $q\,(1-\text{specificity})$
- Thresholds for the false positive and false negative rates: α, $\beta \in (0,\,1)$, respect.
- Estimations $\widehat{m_-}$ and $\widehat{m_+} \geq 0$ summing up to M

Output: The estimated optimal value c^*

```
if $q^M > \alpha$ then
    print ($h_1$) fails. No feasible solution for $\varphi_+(c) \leq \alpha$
    stop
else
    Compute $c_1 = \min\{c \geq 1 \,:\, \varphi_+(c) \leq \alpha\}$, where

    $\varphi_+(c) = \sum_{\ell=c}^{M} \binom{M}{\ell} q^{\ell}\,(1-q)^{M-\ell}$

    if $\widehat{m_-} = 0$ and $1 - \sum_{j=0}^{M-c_1} \binom{M}{j} p^j\,(1-p)^{M-j} > \beta$ then
        print ($h_{2.0}$) fails. No feasible solution for $\widehat{\varphi_-}(c) \leq \beta$
        stop
    else
        if $\widehat{m_-}$, $\widehat{m_+} > 0$ and $q^{\widehat{m_-}-1}\,(\widehat{m_-}\,(1-q) + q\,(1-(1-p)^{\widehat{m_+}})) > \beta$ then
            print ($h_{2.1}$) fails. No feasible solution for $\widehat{\varphi_-}(c) \leq \beta$
            stop
        else
            Compute $c^* = \min\{c \geq c_1 \,:\, \widehat{\varphi_-}(c) \leq \beta\}$, where

            $\widehat{\varphi_-}(c)$ is the estimated false negative rate:

$\sum_{i=(c-1-\widehat{m_+})\vee 0}^{(c-1)\wedge\widehat{m_-}} \binom{\widehat{m_-}}{i} q^i(1-q)^{\widehat{m_-}-i}\Big(\sum_{j=i-(c-1-\widehat{m_+})}^{\widehat{m_+}} \binom{\widehat{m_+}}{j} p^j(1-p)^{\widehat{m_+}-j}\Big)$
Return $c^*$
```

3. Estimation of m_+ and m_-

As outlined in the preciding section, the operational procedure for employing the "c-rule" involves using Algorithm 1 to determine the optimal value c^*. This value ensures that the false positive and false negative rates remain below their respective thresholds. This process necessitates estimating the quantities m_+ and m_-, representing the number of genuinely positive and negative instances within the bag of known size M. In practical scenarios, these quantities are typically unknown but they can be estimated based on the specifications of the base classifier at the instance-level: p and q, which are assumed to be known.

To obtain the estimate $\widehat{m_+}$ for m_+ (and consequently, $\widehat{m_-} = M - \widehat{m_+}$, for m_-), we initially adopt a standard frequentist approach. We draw a random sample of size $n \geq 1$ from the bag, and obtain $\widehat{m_+}$ based on the information contained in this sample. Assume a proportion π of elements in the sample are labeled as positive, with $0 \leq \pi \leq 1$. The challenge lies in deriving an estimation of m_+ from π, considering that the labeling within the sample is subject to the instance-level classification errors. Therefore, instead of defining $\widehat{m_+}$ simply as $M\,\pi$, disregarding classification errors, we proceed as follows:

First, we define $\widehat{m_+} = M\,\widehat{p_+}$, where $\widehat{p_+}$ represents the estimation of the proportion of genuinely positive instances in the bag. This estimation differs from π due to classification errors at the instance level. Specifically, $\widehat{p_+}$ is defined as $\pi\,\widehat{P}(y_{ij} = +1\,/\,\widetilde{y_{ij}} = +1)$, where $\widehat{P}(y_{ij} = +1\,/\,\widetilde{y_{ij}} = +1)$ denotes the estimation of the **positive predictive value** corresponding to classifier $\mathcal{C}$ at the instance level. This value can be derived from the specifications p and q, along with the number of genuinely positive and negative instances in the bag, utilizing Bayes' Theorem in the following way:

$$\begin{aligned} &P(y_{ij} = +1\,/\,\widetilde{y_{ij}} = +1) \\ &\qquad = \frac{P(\widetilde{y_{ij}} = +1\,/\,y_{ij} = +1)\,\frac{m_+}{M}}{P(\widetilde{y_{ij}} = +1\,/\,y_{ij} = +1)\,\frac{m_+}{M} + P(\widetilde{y_{ij}} = +1\,/\,y_{ij} = -1)\,\frac{m_-}{M}} \\ &\qquad = \frac{(1-p)\,m_+}{(1-p)\,m_+ + q\,m_-}\,. \end{aligned}$$

Then,

$$\widehat{P}(y_{ij} = +1\,/\,\widetilde{y_{ij}} = +1) = \frac{(1-p)\,\widehat{m_+}}{(1-p)\,\widehat{m_+} + q\,\widehat{m_-}}\,,$$

and

$$\widehat{m_+} = M\,\widehat{p_+} = M\,\pi\,\widehat{P}(y_{ij} = +1\,/\,\widetilde{y_{ij}} = +1) = M\,\pi\,\frac{(1-p)\,\widehat{m_+}}{(1-p)\,\widehat{m_+} + q\,\widehat{m_-}}\,.$$

By isolating $\widehat{m_+}$ from this expression, considering that $\widehat{m_-} = M - \widehat{m_+}$, we derive that

$$\widehat{m_+} = M\,\frac{\pi\,(1-p)-q}{1-p-q}\,, \tag{3}$$

$$\widehat{m_-} = M - \widehat{m_+} = M\,\frac{(1-p)\,(1-\pi)}{1-p-q}\,. \tag{4}$$

Remark 3. *We must ensure that expressions* (3) *and* (4) *remain non-negative (and hence, between* 0 *and* M*, as they sum up to* M*). First,* $1-p-q>0$ *since we assume that both* p *and* q *are* < 0.5*. Second, from* (3)*, we deduce that* $\widehat{m_+} \geq 0 \Longleftrightarrow \pi \geq \dfrac{q}{1-p}$*. On the other hand,* $\widehat{m_-} \geq 0 \Longleftrightarrow \pi \leq 1$*, which is always satisfied. Therefore, if* $\pi < \frac{q}{1-p}$*, we set* $\widehat{m_+} = 0$ *and* $\widehat{m_-} = M$*. Otherwise, we utilize estimations* (3) *and* (4)*.*

This procedure is summarized in Algorithm 2 below.

Algorithm 2 `Estimating` m_+ `and` m_-

Input:

- Bag size $M \geq 1$
- Classifier's specifications in [0, 0.5): p (1−sensitivity) and q (1−specificity)
- Proportion $\pi \in [0,\,1]$ of positively classified instances in a random sample

Output: The estimations $\widehat{m_-}$ and $\widehat{m_+} \geq 0$ summing up to M

```
if $\pi < \frac{q}{1-p}$ then
    $\widehat{m_+} = 0$ and $\widehat{m_-} = M$
else
    $\widehat{m_+} = M\,\frac{\pi\,(1-p)-q}{1-p-q}$
    $\widehat{m_-} = M - \widehat{m_+} = M\,\frac{(1-p)\,(1-\pi)}{1-p-q}$
Return $\widehat{m_+}$, $\widehat{m_-}$
```

4. Experiments

We illustrate the application of the "c-rule" strategy for multiple instance learning through simulations in two different scenarios, as well as two real-world examples, by unveiling some situations in which the "c-rule" improves bag classification performance when compared to the standard MI assumption, which corresponds to $c = 1$. All the simulations and analysis have been performed using R Statistical Software (v2023.12.1+402) [27], except for obtaining the instance-level predictions using the `CausalMIL` model in Subsection 4.3, which was carried out using Python [28].

4.1. Simulations

We have conducted simulations in the experimental phase of our study, to compare the *standard* rule with the proposed alternative "c-rule", under different scenarios. For that, we construct synthetic examples, varying the *witness rate* (WR), which is defined as the proportion of (genuinely) positive instances in positive bags ([6]), that is, WR $= m_+ \,/\, M$, and under different degrees of misclassification rates at the instance level, p and q.

- **Scenario 1**

Firstly, we want to study the situation, usual in applications, in which WR is arbitrarily small, which causes behavioral problems, for example, in those algorithms that consider that the instances actually have the same label as the bag they form, which does not seem reasonable if WR is low (Section 4.2.1. [6]). For that, we consider a vector of size $N = 10$, say $\gamma = (\gamma_1, \ldots, \gamma_N)$ with values ranging from 0.001 to 0.01, with a step size of 0.001. For each $i = 1, \ldots, N$, we obtain a simulated bag $\mathcal{B}_i$ of size $M_i = 100$ by simulating values of y_{ij}, $j = 1, \ldots, 100$, in $\mathcal{Y}$, such that $P(y_{ij} = +1) = \gamma_i$. That is, bag $\mathcal{B}_i$ has a estimated WR value of γ_i. For bag $\mathcal{B}_i$, we obtain the genuine label y_i by (1). Note that since the estimated WR values of vector γ are small, we obtain few positive bags in the simulation, and even the positive bags have a small number of genuinely positive instances.

Then, we explore all potential combinations of the classifier's specifications p and q, each of them ranging from 0.001 to 0.499, incrementing by 0.005, which are $100 \times 100 = 10\,000$ different combinations. For each of them, we simulate the labels assigned by classifier $\mathcal{C}$ to the instances of any of the bags, $\widetilde{y}_{ij}$. Denote by $\widetilde{\mathcal{B}}_i$ the bag $\mathcal{B}_i$ when replacing y_{ij} by $\widetilde{y}_{ij}$. Following the *standard* rule we determine the label $\widetilde{y}_i^{standard}$ for $\widetilde{\mathcal{B}}_i$ by

$$\widetilde{y}_i^{standard} = \begin{cases} +1 & \text{if } \exists \ j \ \text{ such that } \ \widetilde{y}_{ij} = +1 \\ -1 & \text{otherwise.} \end{cases}$$

and compare with y_i, counting the number of successes in decision-making using the standard rule, which can vary from 0 to 10. Denote by $S^{standard}$ this count, that is,

$$S^{standard} = \#\left\{ i = 1, \ldots, N \, : \, \widetilde{y}_i^{standard} = y_i \right\}.$$

To obtain the labels for $\widetilde{\mathcal{B}}_i$ using the "c-rule", $\widetilde{y}_i^{c\text{-rule}}$, we first need to obtain estimates $\widehat{m_+}$ and $\widehat{m_-}$. To achieve this, we extract a random sample of size $n = 10$ from any of the bags $\widetilde{\mathcal{B}}_i$, calculate the proportion of positively classified instances in this sample, π, and utilize the formulae in Algorithm 2. Finally, Algorithm 1 with reference thresholds for the false positive and false negative rates $\alpha = 0.05$ and $\beta = 0.10$, respectively, provides the estimated value c^* required to implement the "c-rule". Applying the "c-rule" to $\widetilde{\mathcal{B}}_i$ we obtain the label $\widetilde{y}_i^{c\text{-rule}}$, which compare with y_i to get the count of the number of successes, which can also vary from 0 to 10, by defining $S^{c\text{-rule}}$ in this way:

$$S^{c\text{-rule}} = \#\left\{i = 1, \ldots, N \,:\, \widetilde{y}_i^{c\text{-rule}} = y_i\right\}.$$

Finally, we consider the increment in the number of successes using the "c-rule" with respect to use the *standard* rule, defined by

$$\Delta = S^{c\text{-rule}} - S^{standard}.$$

The summary of the three samples of size 10 000 is given in Table 1.

Table 1: Scenario 1: summary of the successes obtained by simulation with the *standard* and the "c-rule", and its difference.

	Min.	Q1	Q2	Mean	Q3	Max.	St. Dev.
$S^{c\text{-rule}}$	0.000	8.000	8.000	7.936	8.000	10.000	1.09278
$S^{standard}$	2.000	2.000	2.000	2.174	2.000	10.000	1.16023
Δ	-2.000	6.000	6.000	5.762	6.000	8.000	1.48638

We visualize the increment Δ for any combination of values of p and q, which are represented on the coordinate axes, in the heatmap in Figure 2.

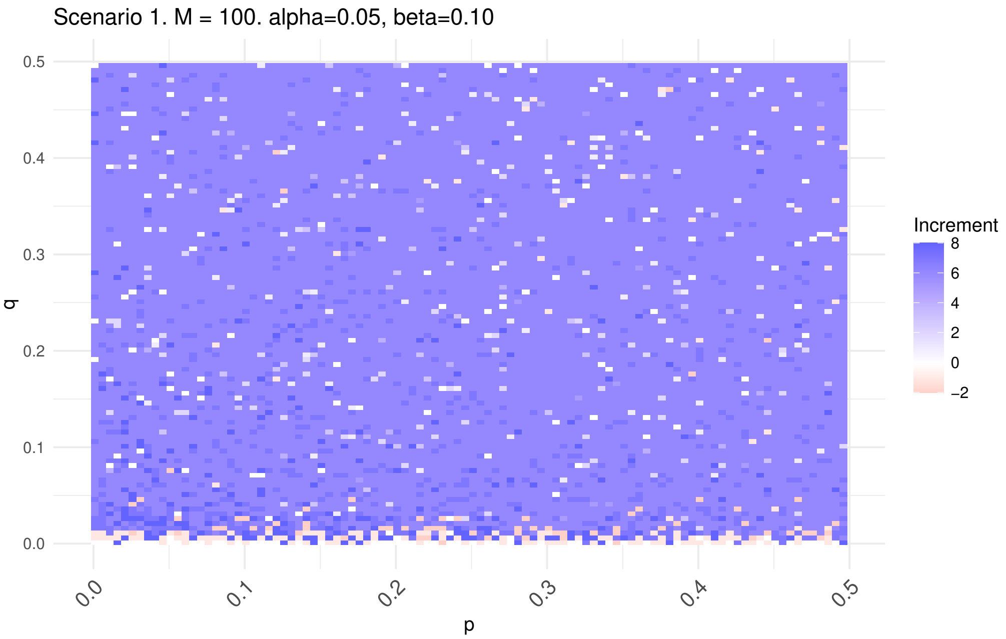
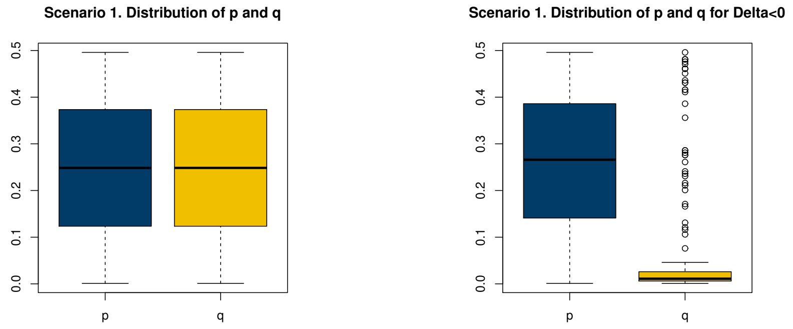


Figure 2: Heatmap for the increment in successes Δ using the "c-rule" with respect to the *standard* rule, for scenario 1.

Both from Table 1 and Figure 2, we observe that the vast majority of cases indicate that the "c-rule" yields more successes than the standard rule. Specifically, out of the 10 000 simulated cases, 9 630 show a positive increment, 202 show a negative increment, and 168 remain unchanged. This accounts for a percentage of 97.9 % favoring the "c-rule" over cases showing differences between the two rules, and 96.3 % over the total number of 10 000 cases. Additionally, the majority of cases with negative increment correspond to low values of q. This fact can be visualized in the distribution of values p and q for the 202 cases with a negative Δ value given in Table 2 and boxplots in Figure 3.

Table 2: Scenario 1: summary of the p and q values for cases with $\Delta < 0$.

	Min.	Q1	Q2	Mean	Q3	Max.	St. Dev.
p	0.001	0.141	0.266	0.259	0.386	0.496	0.14489
q	0.001	0.006	0.011	0.068	0.026	0.496	0.13335



(a) For all the cases

(b) For cases in which $\Delta < 0$

Figure 3: Boxplots for the distribution of the values of p and q in scenario 1.

As expected, the "c-rule" gives better results than the *standard* rule for sparse positive bags with q varying from not very low to high, independently of p.

- **Scenario 2**

Now we simulate a dataset analogous to the artificially created AIMed [8], with 670 truly positive bags (each one with only one genuinely positive instance), and 1 040 truly negative bags. Overall, 1 710 bags. As details about the size of the bags are no provided in [8], we assume that the size is $M_i = 100$ for all of them. Then, in this scenario, WR is constant and equal to 0.01 for all the bags. For each bag we explore all potential combinations of p and q, each of them ranging from 0.001 to 0.300, incrementing by 0.01, which are $30 \times 30 = 900$ different combinations. For each of them, we simulate the labels assigned by classifier $\mathcal{C}$ to the instances of any of the bags, analogously to the previous scenario. We assign the label to the bags following both the *standard* and the "c-rule", obtaining the number of successes in labeling the bags for any rule, which can vary between 0 and 1 710, as summarized in Table 3.

Table 3: Scenario 2, M = 100: summary of the successes obtained by simulation with the *standard* and the "c-rule, and its difference.

	Min.	Q1	Q2	Mean	Q3	Max.	St. Dev.
$S^{c\text{-rule}}$	1 001	1 034	1 040	1 042.0	1 048	1 126	14.3113
$S^{standard}$	669	670	670	714.4	671	1 634	161.9412
Δ	-538	357	366	327.6	373	403	152.3675

Figure 4 shows the increment Δ for any combination of values of p and q, represented on the coordinate axes. As in scenario 1, from Table 3 and Figure 4, we observe that for the majority of cases the "c-rule" overcomes the *standard* rule. Out of the 900 simulated cases, representing different combinations of p and q, 870 exhibit a positive increment, while only 30 display a negative increment, favoring the "c-rule" with a percentage of 96.7 %. The 30 cases with negative increments are associated with the lowest value of the probability of misclassifying negative instances, $q = 0.001$, while p varies across its defined range.

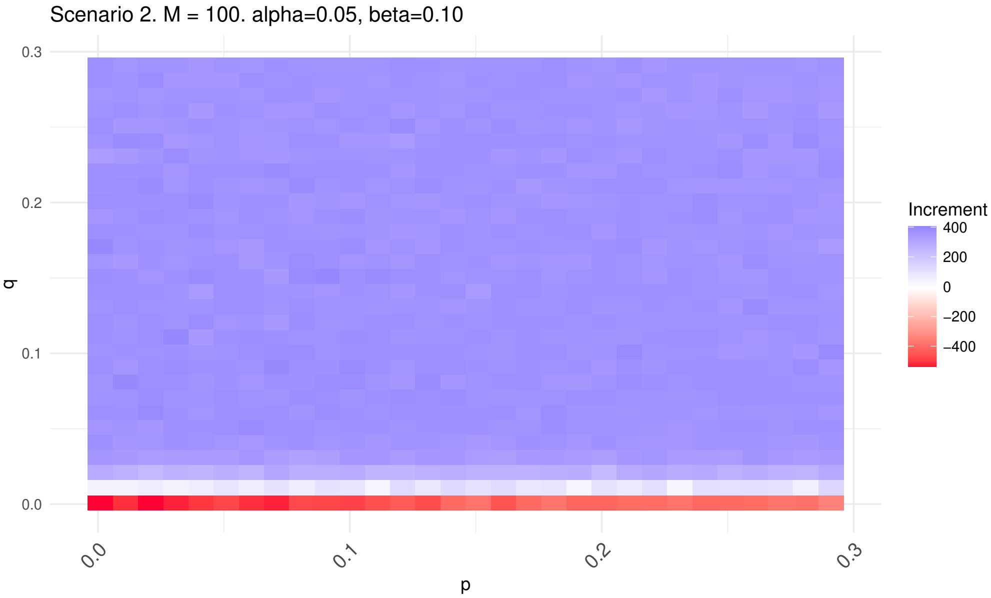


Figure 4: Heatmap for the increment in successes Δ using the "c-rule" with respect to the *standard* rule, for scenario 2 with $M = 100$.

Similar findings were observed when assuming bag sizes of $M_i = 50$. Specifically, 840 out of the 900 cases favored the "c-rule" (93.3%). The 60 cases with negative increments correspond also to very low values of q, ranging from 0.001 to 0.011, with an average of 0.006 and a standard deviation of 0.005042, irrespective of p.

4.2. First real-world case study: the **Musk-2** *dataset*

- **The Musk Dataset.** As mentioned in the introduction, MIL was initially introduced by [1] in the context of drug activity prediction. Specifically, in their study the authors addressed the challenge of determining whether a drug molecule exhibits a desired pharmacological activity, as is the case of the musk property, which is valuable in the production of cosmetics. Each molecule can adopt various shapes or conformations due to bond rotations. The entire molecule is classified as having the property If and only if any of these conformations exhibit the musk property.

This problem aligns naturally with the MIL framework and the standard MI assumption: each molecule is treated as a bag, and each possible conformation represents an instance within that bag. Both bags and instances are binary classified, but the instances labels are unobserved, while the bag labels are known. We focus on **Musk-2**, one of the two datasets from [1], which consists of 102 annotated bags (39 positive, 63 negative), 6,598 unlabeled instances, and 166 features describing the shape or conformation of the molecules[1]. The goal is to predict whether a new molecule must be classified as musk or non-musk. The original data set is partitioned using 10-fold cross-validation *procedure*, repeated five times, resulting in five distinct 10-fold cross validation partitions.

- **The `milr` R package.** The recently released `milr` package[2] for R is designed for building predictive models for multiple instance learning (MIL) datasets with binary outcomes, using a logistic regression framework.

Our implementation follows the *milr approach* ([29]), available via the `milr` function, and is complemented by variable selection using the *softmax approach* ([30]), implemented in the `softmax` function. Both techniques are key components to the `milr` package's MIL modeling process.

The predictive model generated by the `milr` function is primarily designed to predict bag-level labels, but it can also provide instance-level predictions, which are necessary for comparing the "c-rule" with the standard MIL assumption.

- **The Experiment.** For each training dataset in the 10-fold cross validation partitions, across the five different *procedures*, we first perform variable selection using both the $s(0)$ ([30]) and $s(3)$ ([31]) `softmax` methods. Features with absolute coefficients greater than 0.05 are selected, with a minimum of two features to avoid errors, and a maximum of ten to prevent overfitting. As recommended in the `milr` vignette[3], we set the parameter $\lambda = 10^{-7}$ to bypass Hessian matrix evaluation while keeping the default EM algorithm maximum of 500 iterations, and train the `milr` model to obtain the corresponding fitted instance-level labels. We then apply both the standard procedure ($c = 1$) and

[1] This dataset is available for download from the KDIS (Knowledge Discovery and Intelligent Systems) research group at the Department of Computer Science and Numerical Analysis of the University of Córdoba, Spain: https://www.uco.es/kdis/momil/

[2] https://CRAN.R-project.org/package=milr

[3] https://cran.r-project.org/web/packages/milr/vignettes/milr-intro.html#the-milr-and-softmax-apporaches

the "c-rule", implemented in Algorithm 1, after estimating m_+ and m_- by Algorithm 2.

We fix the parameter values at $\alpha = \beta = 0.10$, and q is estimated as the proportion of instances with a positive fitted label that belong to bags annotated as negative, denoted by q_{est}. To introduce some variability, we assume different values of q within a small interval centered around q_{est} with a radius of $q_{est}/3$, avoiding zero, as it represents a degenerate case. The parameter π is estimated as the average proportion of positively classified instances for each bag, for any given training dataset.

For the "c-rule", overall performance metrics are calculated for each training dataset as the average of their values across the different combinations of p and q. We consider three metrics: *Accuracy*, the proportion of correctly labelled bags; *Balance Accuracy*, the arithmetic mean of the proportion of correctly labeled positive and negative bags; and *F1-score*, with a focus on the negative class, since our primary interest lies in the false positive, considered the more critical of the two possible errors in bag-level classification.

• **The Results.** We present results based solely on the fitted values obtained from the training instances and not from the test instances, as the test sets contain too few bags to allow for a meaningful comparison between the two methods. Although we performed a comparative analysis on the test sets, the results did not yield any significant findings in either direction.

For each of the five *procedures* we performed a paired comparison between the 10 values of each metric obtained with the standard rule, and the 10 averages of the metric values across the different combinations of p and q obtained using the "c-rule". We first applied the Shapiro-Wilk normality test to the differences between the two sets of values to determine whether to use a paired Student's t-test (if normality was not rejected at a significance level of 0.05) or a paired Wilcoxon signed-rank test otherwise, since paired tests are specifically designed for comparing paired samples, which aligns with the cross-validation setup. p-values are reported in Tables 4-6. As usual, superscripts $^{\bullet}$ indicate statistical significance at 10%, * at 5%, ** at 1% and *** at 1‰. Statistical significance is observed in all cases in favour of the "c-rule".

Additionally, we present boxplots in Figures Appendix B.1-Appendix B.3, illustrating the distribution of the increase in metrics when applying the "c-rule" compared to the standard rule, for both the $s(0)$ and $s(3)$ feature selection methods. We observe that, in general, the boxplots lie above the red line, indicating the superiority of the "c-rule." The few datasets for which this is not the case have an estimated value of π which is 0 or very low. Among the three metrics, BA appears to be the most sensitive to this phenomenon.

We also compare the "c-rule" for bag labeling (derived from fitted instance labels using the `milr` model) with the fitted bag labels provided directly by the `milr` model. Although the differences are not significant in all cases or for every metric, when a significant difference does occur, it consistently favours our method. Table 7 shows the corresponding p-values.

Table 4: P-values for the one-sided paired tests with the alternative hypothesis that the mean (or median) Accuracy with the "c-rule" is greater than that with the standard procedure ($c = 1$). We also provide the p-values for the Shapiro-Wilk normality test, which determine the paired test to be used.

Accuracy	Var. selection	Shapiro-Wilk	t-test
Procedure 1	`softmax` $s(0)$	0.5830	0.01404^{*}
	`softmax` $s(3)$	0.1830	0.01746^{*}
Procedure 2	`softmax` $s(0)$	0.2734	0.01546^{*}
	`softmax` $s(3)$	0.9862	0.00180^{**}
Procedure 3	`softmax` $s(0)$	0.1329	0.00193^{**}
	`softmax` $s(3)$	0.3313	0.00317^{**}
Procedure 4	`softmax` $s(0)$	0.8454	0.00071^{***}
	`softmax` $s(3)$	0.9069	$2.207 \times 10^{-6\,***}$
Procedure 5	`softmax` $s(0)$	0.3779	0.00528^{**}
	`softmax` $s(3)$	0.2938	0.00122^{**}

Table 5: Analogous to Table 4 for the F1-score metric.

F1-score	Var. selection	Shapiro-Wilk	t-test	Wilcoxon
Procedure 1	`softmax` $s(0)$	0.0136^{*}		0.02632^{*}
	`softmax` $s(3)$	0.1487	0.01217^{*}	
Procedure 2	`softmax` $s(0)$	0.2182	0.01089^{*}	
	`softmax` $s(3)$	0.7976	0.00046^{***}	
Procedure 3	`softmax` $s(0)$	0.1573	0.00197^{**}	
	`softmax` $s(3)$	$5.037 \times 10^{-5\,***}$		0.00098^{***}
Procedure 4	`softmax` $s(0)$	0.7453	0.00216^{**}	
	`softmax` $s(3)$	0.04436^{*}		0.00098^{***}
Procedure 5	`softmax` $s(0)$	0.3622	0.00500^{**}	
	`softmax` $s(3)$	0.6320	0.00251^{**}	

Table 6: Analogous to Tables 4-5 for the Balance Accuracy (BA) metric.

BA	Var. selection	Shapiro-Wilk	t-test	Wilcoxon
Procedure 1	`softmax` $s(0)$	0.4499	$0.07151^{\bullet}$	
	`softmax` $s(3)$	0.1951	0.004578^{**}	
Procedure 2	`softmax` $s(0)$	0.3012	0.00795^{**}	
	`softmax` $s(3)$	0.3926	$0.05578^{\bullet}$	
Procedure 3	`softmax` $s(0)$	0.1248	0.00050^{***}	
	`softmax` $s(3)$	$5.419 \times 10^{-5\,***}$		0.04199^{*}
Procedure 4	`softmax` $s(0)$	0.5230	0.01963^{*}	
	`softmax` $s(3)$	0.8163	$3.016 \times 10^{-5\,***}$	
Procedure 5	`softmax` $s(0)$	0.7181	0.001073^{**}	
	`softmax` $s(3)$	0.6569	0.01028^{*}	

4.3. Second real-world case study: the **Colon Cancer** *dataset*

- **The Colon Cancer Dataset.** In this part of the experimental section, we analyze the processed patches from the **Colon Cancer** dataset, available

Table 7: P-values for Shapiro-Wilk and one-sided paired tests with the alternative hypothesis that the mean (or median) metric for with the"c-rule" is greater than that for bag labeling directly from the `milr` model. Only statistically significant results are reported.

	Metric	Var. selection	Shapiro-Wilk	t-test	Wilcoxon
Procedure 2	Accuracy	`softmax` $s(3)$	0.0048^{**}		$0.05273^{\bullet}$
	F1-score	`softmax` $s(3)$	0.8181	0.01396^{*}	
Procedure 3	Accuracy	`softmax` $s(3)$	0.8259	$0.06057^{\bullet}$	
	F1-score	`softmax` $s(0)$	0.01937^{*}		0.04147^{*}
		`softmax` $s(3)$	$4.922 \times 10^{-5\,***}$		0.00977^{**}
Procedure 4	F1-score	`softmax` $s(0)$	0.7162	$0.05156^{\bullet}$	
		`softmax` $s(3)$	0.9511	0.04843^{*}	
Procedure 5	F1-score	`softmax` $s(0)$	0.7066	$0.06038^{\bullet}$	
		`softmax` $s(3)$	0.0017^{**}		0.03223^{*}

at https://github.com/utayao/Atten_Deep_MIL (credit to Jiawen Yao). This dataset comprises 99 images derived from colon tissue, 51 of which are labeled as positive, indicating the presence of epithelial cells (considered malignant), while the remaining images are labeled as negative. For each image (bag), instances are generated as 27 × 27 pixel patches, resulting in a total of 20,361 instances.

Although the dataset includes ground truth labels for individual instances, we do not use them in this experiment (except to estimate the classifier's specifications p and q), as our aim is to reflect the practical scenario where such information is typically unavailable. However, ground truth instance labels are used indirectly to label bags as positive if and only if they contain at least one positive instance. Ground truth bag labels are then employed to make predictions at the instance level using the `CausalMIL` algorithm. These predictions are subsequently aggregated to obtain bag-level predictions using both the standard rule ($c = 1$) and the proposed "c-rule". The performance of these predictions is evaluated using various metrics to compare the effectiveness of the two methods, which is the main objective of this analysis.

• **The `CausalMIL` algorithm.** `CausalMIL` [32] is a Python-based algorithm implemented using PyTorch 1.12. It employs a neural network architecture with encoders and decoders to learn meaningful representations of the image patches, which serve as instances in the MIL framework.

The algorithm and its associated `colon_cancer.py` dataloader are publicly available at https://github.com/WeijiaZhang24/CausalMIL. The model is trained using 5-fold cross-validation to ensure robust performance evaluation, with metrics AUC (Area Under the Receiver Operating Characteristic Curve) and AUC-PR (Area Under the Precision-Recall Curve) used to assess its effectiveness.

• **The Experiment.** For each of the five folds of the Colon Cancer dataset generated using the `colon_cancer.py` dataloader, we first apply the CasualMIL algorithm to obtain predictions at the instance level. Folds 1 to 4 consist of 20 bags (images) each, comprising both positive and negative examples, while the fifth fold contains 19 bags. To assign a binary label (0 or 1) to the instances, we consider a range of thresholds from 0.01 to 0.50, incremented by 0.01. As the dataset includes ground truth annotated instance-level labels, we calculate the classifier's specifications p and q for each threshold, where p represents the proportion of positive instances misclassified as negative, and q represents the proportion of negative instances misclassified as positive. Figure 5 shows the estimated values of p and q across varying thresholds for each of the five folds.

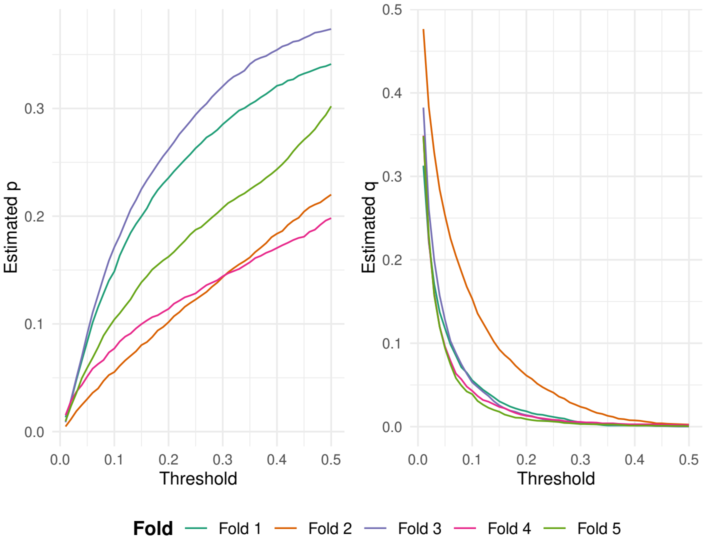


Figure 5: Estimated values of p (left panel) and q (right panel) as a function of the threshold, for each of the five folds of the dataset, with colors distinguishing the folds.

Subsequently, we use the standard procedure (with $c = 1$) to derive predicted bag-level labels. To implement the proposed "c-rule" (Algorithm 1), we estimate m_+ and m_- using Algorithm 2, and set $\alpha = \beta = 0.10$. This allows to obtain bag-level predictions based on the "c-rule" as well.

For each fold (1 through 5) and threshold (0.01 to 0.50), we obtain confusion matrices for bag-level classification. This yields values for various performance metrics (*Accuracy*, *F1-score*, and *Balance Accuracy*) for every fold-threshold combination. These metrics enable a statistical comparison between the standard rule and the proposed "c-rule", following the statistical methodology detailed in Section 4.2.

• **The Results.** For each metric and threshold, we perform paired comparisons using samples of size 5 (corresponding to the folds) between the values obtained with the standard rule and the "c-rule". Table 8 presents the total count of p-values favoring one rule over the other. The results show that the "c-rule" is favored more often than the standard rule, except for *Balance Accuracy*, which exhibits more erratic behavior in this experiment. Notably, the four cases where the standard rule is favored correspond to the highest thresholds (0.47 to 0.50), while the lowest thresholds favor the "c-rule".

Table 8: Count of p-values (our of 50 possible) that are < 0.05 for one-sided paired tests, evaluating the alternative hypothesis that the mean (or median) metric given by the column using the "c-rule" is greater than that of the standard procedure (first row), or vice versa (second row).

	Accuracy	F1-score	Balance Accuracy
In favor of "c-rule"	12	14	0
In favor of standard ($c = 1$)	4	4	4

We compare the distribution of performance metric improvements across varying thresholds when using the "c-rule" versus the standard rule, for each of the five folds, as illustrated in Figure 6.

It is noteworthy that fold 3 exhibits the most unfavorable behavior towards the "c-rule" in Figure 6, while the other folds show either favorable or neutral behavior, depending on the metric. This is because, among the five folds, fold 3 has the lowest proportion of truly negative bags –only 4 out of 16 (20%). In contrast, the other folds range from 47.4% in fold 5 to 65% in fold 2. Among the metrics, BA (*Balance Accuracy*) again demonstrates its erratic behavior, making it less suitable for this real-world example.

We also can analyze in Figure 7 the evolution of the increase in performance metrics when using the "c-rule" compared to the standard rule by plotting the mean value across the five folds as the threshold varies from 0.01 to 0.50. Since fold 3 exhibits significantly worse behavior compared to the others, we repeat the analysis excluding fold 3. In this latter case, it is observed that the results for the "c-rule" are better for thresholds that are not too high, up to approximately 0.35. This range of thresholds yields estimated p values that vary from very low to medium, while the estimated q values span from very low and high (Figure 5).

Additionally, Figure Appendix C.1 illustrates the evolution of the increase in performance metrics when applying the "c-rule" compared to the standard rule, as the threshold varies, for each of the five folds (the mean values of which are plotted in Figure 7). It reveals a similar pattern to that observed in Figure 7, along with the noticeably less favorable behavior of fold 3.

5. Conclusions

The research into addressing mislabeled instances in Multiple Instance Learning (MIL) within the framework of the *standard MI assumption* remains a dynamic and evolving research domain. In this paper the focus lies in attenuating

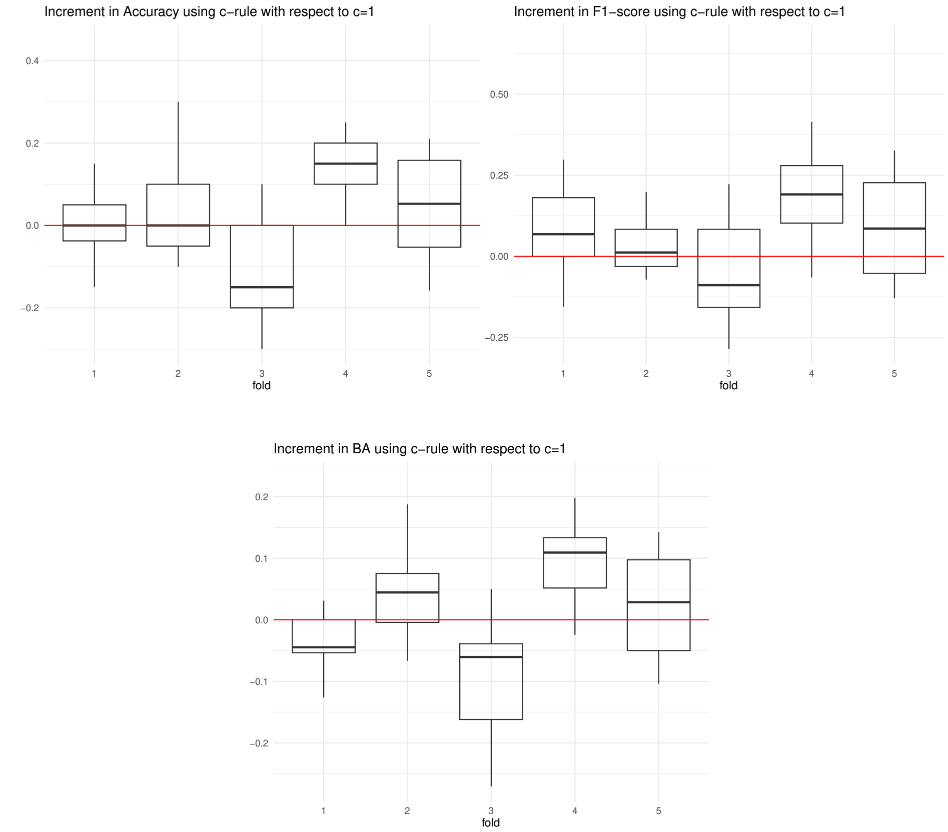


Figure 6: Boxplots showing the distribution of performance metric improvements at varying thresholds when using the "*c*-rule" compared to the standard rule across the five folds of the Colon Cancer dataset. The red line marks the zero level.

the propagation of errors inherent in bag labeling arising from errors at the instance-label level, particularly emphasizing the classification of negative bags.

In this sense, we are interested in control both, the false positive and false negative rates. In medical imaging classification, false positive rates can undermine trust in diagnostic processes and imaging technologies. This can result in skepticism among healthcare providers and patients regarding the reliability of MIL applications in medical contexts, potentially leading to delayed or missed diagnoses. It is imperative to minimize these rates through algorithmic refinement, protocol enhancements, and rigorous quality control measures. By prioritizing the mitigation of labeling errors, healthcare providers can bolster the overall efficacy and safety of medical imaging diagnostics, ultimately improving patient outcomes and reducing unnecessary healthcare costs. The *standard* rule often exhibits a very high false positive rate, prompting us to introduce the

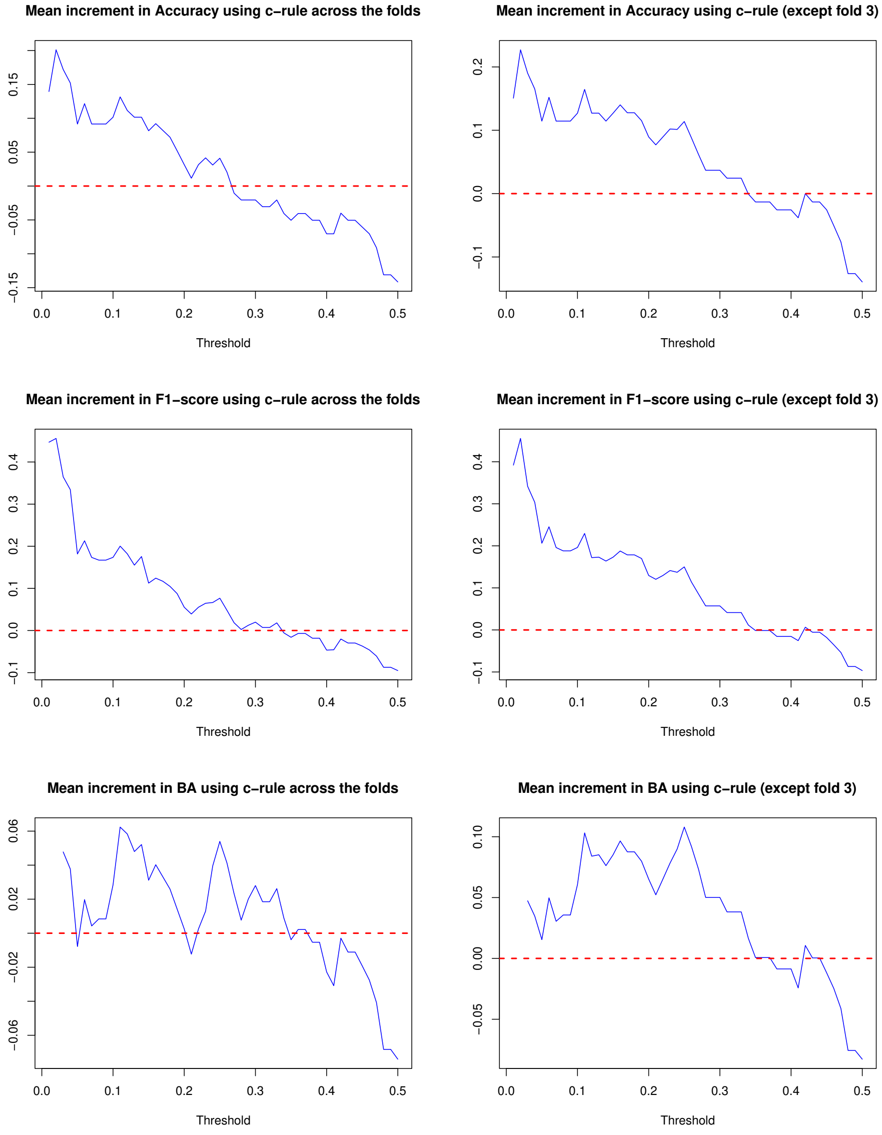


Figure 7: Evolution of the mean increase when using the "*c*-rule" compared to the standard rule, as the threshold varies. Left column refers to the mean for all five folds. Right column excludes fold 3 in the mean.

"*c*-rule" as an alternative approach to error control. Rooted in theoretical foundations, the "*c*-rule" excels in accurately classifying genuinely negative bags, since the false positive rate associated to this rule is:

$$\varphi_+(c^*) = \sum_{\ell=c^*}^{M} \binom{M}{\ell} q^\ell \, (1-q)^{M-\ell} \leq \sum_{\ell=1}^{M} \binom{M}{\ell} q^\ell \, (1-q)^{M-\ell} = \varphi_+(1) \, ,$$

the later corresponding to the *standard* rule. Through simulations conducted in real-world scenarios, and experiments with two actual datasets, we have illustrated the superiority of the "c-rule" over the *standard* rule, particularly in classifying sparse positive bags characterized by a low *witness rate* WR and a value of q varying from moderately low to high.

This investigation underscores the importance of tailored approaches to MIL, ensuring robust and accurate bag labeling across diverse application contexts. By improving how we handle incorrect labeling in Multiple Instance Learning (MIL), we make strides toward better, more dependable learning methods for challenging environments.

Declaration of competing interest

The author declare that has no competing interests nor conflict of interests in relation to the research reported in this paper.

Funding

The author is supported by Ministerio de Ciencia e Innovación, Gobierno de España, project ref. PID2021-123733NB-I00.

Code availability

The R scripts used in the experimental phase of this study (Section 4), along with detailed instructions for accessing the datasets used in Sections 4.2 and 4.3, are available at https://github.com/RosDelgado/MIL_Noise.

Acknowledgements

I sincerely thanks the anonymous referees and the editors for their valuable feedback, which has greatly enhanced the quality of this work.

Special thanks are also extended to the members of the KDIS (Knowledge Discovery and Intelligent Systems) research group at the Department of Computer Science and Numerical Analysis of the University of Córdoba, Spain, for granting access to the Musk-2 dataset via their website. Additional gratitude goes to Jiawen Yao for processing the patches of the Colon Cancer dataset and making them available on Github.

Finally, I wish to express my heartfelt gratitude to Xavier-Andoni Tibau for generously sharing his expertise. His efficient and selfless assistance in running

the `CasualMIL` algorithm on the Colon Cancer dataset was invaluable in generating the instance-level predictions used in the experimental phase described in Subsection 4.3.

Appendix A. Proofs of Lemmas and Proposition 2

In this appendix, we present the proofs of the lemmas used in the proof of Proposition 2 in Section 2, along with the proof of the proposition itself.

Lemma 1. *If $m_+ \geq 1$, we can express $\varphi_-(c)$ as follows:*

$$\varphi_-(c) = \sum_{i=(c-1-m_+)\vee 0}^{(c-1)\wedge m_-} \binom{m_-}{i} q^i (1-q)^{m_- - i} \Big(\sum_{j=i-(c-1-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big) \tag{A.1}$$

Proof. By definition, $\varphi_-(c)$ represents the probability that a bag containing at least one of the M instances being actually positive, receives a negative label. Under the *standard* MIL rule, this implies that the m_+ positive instances are misclassified as negative, while the m_- negative instances are correctly classified, happening with a probability of $p^{m_+}(1-q)^{m_-}$. This coincides with (A.1) if $c = 1$. However, with the "c-rule" for $c \geq 1$ in general, the situation becomes more complex, and this is what we will consider next.

Denote by F_+ (respectively, F_-) the number of the m_+ (respectively, m_-) actually positive (respectively, negative) instances that are wrongly classified. Then, under the "c-rule" we have that

$$\varphi_-(c) = P((m_+ - F_+) + F_- < c) \,.$$

In simpler terms, to assign a (wrong) negative label to the bag with the "c-rule", the sum of positive instances correctly classified, $m_+ - F_+$, plus the number of negative instances wrongly classified as positive, F_-, must be strictly less than c. Given that both F_- and F_+ are random variables assumed to follow independent Binomial distributions, specifically,

$$F_- \sim B(m_-,\, q) \quad \text{and} \quad F_+ \sim B(m_+,\, p)$$

(with the convention $F_- = 0$ if $m_- = 0$), we can express $\varphi_-(c)$ as follows:

$$\varphi_-(c) = \sum_{i=0}^{m_-} \sum_{j=0}^{m_+} P(F_- = i,\, F_+ = j,\, m_+ + i - j < c) \,. \tag{A.2}$$

Now, consider that $m_+ + i - j < c$ is equivalent to $j \geq m_+ + i + 1 - c$, and for this to being meaningful:

- ✓ $m_+ + i + 1 - c \geq 0$, which is equivalent to $i \geq c - 1 - m_+$, and
- ✓ $m_+ + i + 1 - c \leq m_+$, which is equivalent to $i \leq c - 1$.

Hence, the index i must range between $c-1-m_+$ and $c-1$. To ensure that i stays between 0 and m_-, we take the maximum of the first expression with 0 and the minimum of the second expression with m_-. Therefore, (A.2) can be rewritten as follows, which coincides with (A.1), finishing the proof of the lemma:

$$\varphi_-(c) = \sum_{i=(c-1-m_+)\vee 0}^{(c-1)\wedge m_-} \; \sum_{j=m_++i+1-c}^{m_+} P(F_- = i)\, P(F_+ = j)$$

$$= \sum_{i=(c-1-m_+)\vee 0}^{(c-1)\wedge m_-} \binom{m_-}{i} q^i (1-q)^{m_- - i} \Big(\sum_{j=i-(c-1-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big). \;\square$$

Lemma 2. *The following statements hold for function $\varphi_-(c)$ from Lemma 1:*

- *If $m_- = 0$, $\varphi_-(c)$ is an increasing function of c, for $1 \le c \le M$.*
- *If $m_- \ge 1$ and $m_+ \ge 1$, function $\varphi_-(c)$ has the following behavior:*

  a) *if $1 \le c \le m_+ + 1$, then $\varphi_-(c)$ is an increasing function,*

  b) *if $\dfrac{M + m_+ + 1}{2} \le c \le M$, then $\varphi_-(c)$ is a decreasing function.*

Proof. We will establish the validity of statements in the same order.

- If $m_- = 0$ (then, $M = m_+$), the index i in (A.1) must necessarily be 0. Therefore, by (A.1) we have that

  $$\varphi_-(c) = \sum_{j=M+1-c}^{M} \binom{M}{j} p^j \,(1-p)^{M-j}\,,$$

  and this expression trivially increases with c. Note that since $c \ge 1$, $M + 1 - c \le M$, while the fact that $c \le M$ implies that $M + 1 - c \ge 1$.

- In the case where $m_- \ge 1$, we distinguish between two cases:

  a) $1 \le c \le m_+ + 1$. In this case, from (A.1) we obtain that $\varphi_-(c)$ can be written as

  $$\sum_{i=0}^{(c-1)\wedge m_-} \binom{m_-}{i} q^i (1-q)^{m_- - i} \Big(\sum_{j=i-(c-1-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big).$$

  This expression is trivially increasing as function of c, as both the intervals of possible values for i and for j expand, or at least, do not contract, when c increases.

b) $m_+ + 1 \leq c \leq M$. The expression (A.1) for $\varphi_-(c)$ in this other case is given by:

$$\varphi_-(c) = \sum_{i=c-1-m_+}^{(c-1)\wedge m_-} a_i \quad \text{with}$$

$$a_i = \binom{m_-}{i} q^i (1-q)^{m_- - i} \Big(\sum_{j=i-(c-1-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big).$$

To study the decrease of $\varphi_-(c)$, that is, if it happens that $\varphi_-(c+1) \leq \varphi_-(c)$, we write $\varphi_-(c+1)$ as:

$$\varphi_-(c+1) = \sum_{\ell=c-m_+}^{c\wedge m_-} b_\ell \quad \text{with}$$

$$b_\ell = \binom{m_-}{\ell} q^\ell (1-q)^{m_- - \ell} \Big(\sum_{j=\ell-(c-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big)$$

and compare term by term with $\varphi_-(c)$. We will consider two scenarios:

b.1) If $c \leq m_-$, there are $m_+ + 1$ terms a_i, with $i = c-1-m_+, \ldots, c-1$, and also $m_+ + 1$ terms b_ℓ, with $\ell = c - m_+, \ldots, c$. We will make a direct comparison term-by-term of a_i and b_{i+1} for $i = c-1-m_+, \ldots, c-1$.

b.2) If $c > m_-$, there are $M - c + 2$ terms a_i with $i = c - 1 - m_+, \ldots, m_-$, and $M - c + 1$ terms b_ℓ with $\ell = c - m_+, \ldots, m_-$. In this case, we will compare one-to-one the terms a_i and b_{i+1} for $i = c - 1 - m_+, \ldots, m_-$, while the term a_{m_-} remains unpaired. Importantly, since this unpaired term belongs to $\varphi_-(c)$, it does not affect our proof establishing that $\varphi_-(c) \geq \varphi_-(c+1)$ for c big enough.

Then, for both scenarios, fixing $i = c-1-m_+, \ldots, (c-1) \wedge m_-$, we aim to prove that $a_i \geq b_{i+1}$, where

$$a_i = \binom{m_-}{i} q^i (1-q)^{m_- - i} \Big(\sum_{j=i-(c-1-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big),$$

$$b_{i+1} =$$

$$\binom{m_-}{i+1} q^{i+1} (1-q)^{m_- - (i+1)} \Big(\sum_{j=i+1-(c-m_+)}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+ - j} \Big).$$

Since the sums on j are identical, $a_i \geq b_{i+1}$ is equivalent to

$$\binom{m_-}{i} q^i (1-q)^{m_- - i} \geq \binom{m_-}{i+1} q^{i+1} (1-q)^{m_- -(i+1)}$$
$$\iff q\,(m_- - i) \leq (1-q)\,(i+1).$$

Taking into account that we are assuming that $q \leq 0.5$, then $q \leq 1-q$ and the previous inequality holds if

$$m_- - i \leq i+1 \iff i \geq \frac{m_- - 1}{2}\,.$$

Since $i \geq c-1-m_+$, it is sufficient if the following inequality holds:

$$c-1-m_+ \geq \frac{m_- - 1}{2} \iff c \geq \frac{M+m_+ +1}{2}\,.$$

Note that we can ensure that $m_+ + 1 \leq \frac{M+m_+ +1}{2} \leq M$ since we are in the case $m_- \geq 1$. □

Proposition 2. *Fix $\beta \in (0,\,1)$. Then,*

a) *Fix a positive integer c_1, $1 \leq c_1 \leq M$, and consider the hypothesis*

$$(h_{2.0}) \quad 1 - \sum_{j=0}^{M-c_1} \binom{M}{j} p^j (1-p)^{M-j} \leq \beta\,.$$

Then, $(h_{2.0})$ is a necessary and sufficient condition in the case $m_- = 0$ (then, $m_+ = M \geq 1$) for the existence of a positive integer c_2, $c_1 \leq c_2 \leq M$, such that

$$\varphi_-(c) \leq \beta \textit{ for all } c_1 \leq c \leq c_2\,.$$

b) *Consider the hypothesis*

$$(h_{2.1}) \quad q^{m_- -1}\Big(m_-(1-q) + q\big(1-(1-p)^{m_+}\big)\Big) \leq \beta\,.$$

Then, $(h_{2.1})$ is a necessary and sufficient condition in the case $m_- \geq 1$ and $m_+ \geq 1$ for the existence of a positive integer c_2, $1 \leq c_2 \leq M$, such that

$$\varphi_-(c) \leq \beta \textit{ for all } c_2 \leq c \leq M\,.$$

Proof. We distinguish between two situations:

a) $m_- = 0$. In this case, Lemma 2 establishes that $\varphi_-(c)$ is an increasing function. When $m_- = 0$ (then, $m_+ = M \geq 1$), we can express $\varphi_-(c)$ as:

$$\varphi_-(c) = \sum_{j=M-c+1}^{M} \binom{M}{j} p^j (1-p)^{M-j} = 1 - \sum_{j=0}^{M-c} \binom{M}{j} p^j (1-p)^{M-j}$$

and then, $\varphi_-(c_1) = 1 - \sum_{j=0}^{M-c_1} \binom{M}{j} p^j\,(1-p)^{M-j}$ is the minimum of $\varphi_-(c)$ with $c_1 \le c \le M$. Then, $(h_{2.0})$ is necessary and sufficient for the existence of $c_2 \ge c_1$ such that $\varphi_-(c) \le \beta$ for $c_1 \le c \le c_2$.

b) $m_- \ge 1$ and $m_+ \ge 1$. Lemma 2 shows that $\varphi_-(c)$ increases for $1 \le c \le m_+ + 1$ and decreases for $\frac{M+m_++1}{2} \le c \le M$. As a consequence, we can ensure the existence of c_2, $1 \le c_2 \le M$, such that $\varphi_-(c) \le \beta$ for all c within the range $c_2 \le c \le M$, if and only if $\varphi_-(M) \le \beta$. By replacing c with M in expression for $\varphi_-(c)$ in Lemma 1, we can write

$$\varphi_-(M) = \sum_{i=m_--1}^{m_-} \binom{m_-}{i} q^i (1-q)^{m_--i} \Big(\sum_{j=i+1-m_-}^{m_+} \binom{m_+}{j} p^j (1-p)^{m_+-j} \Big)$$

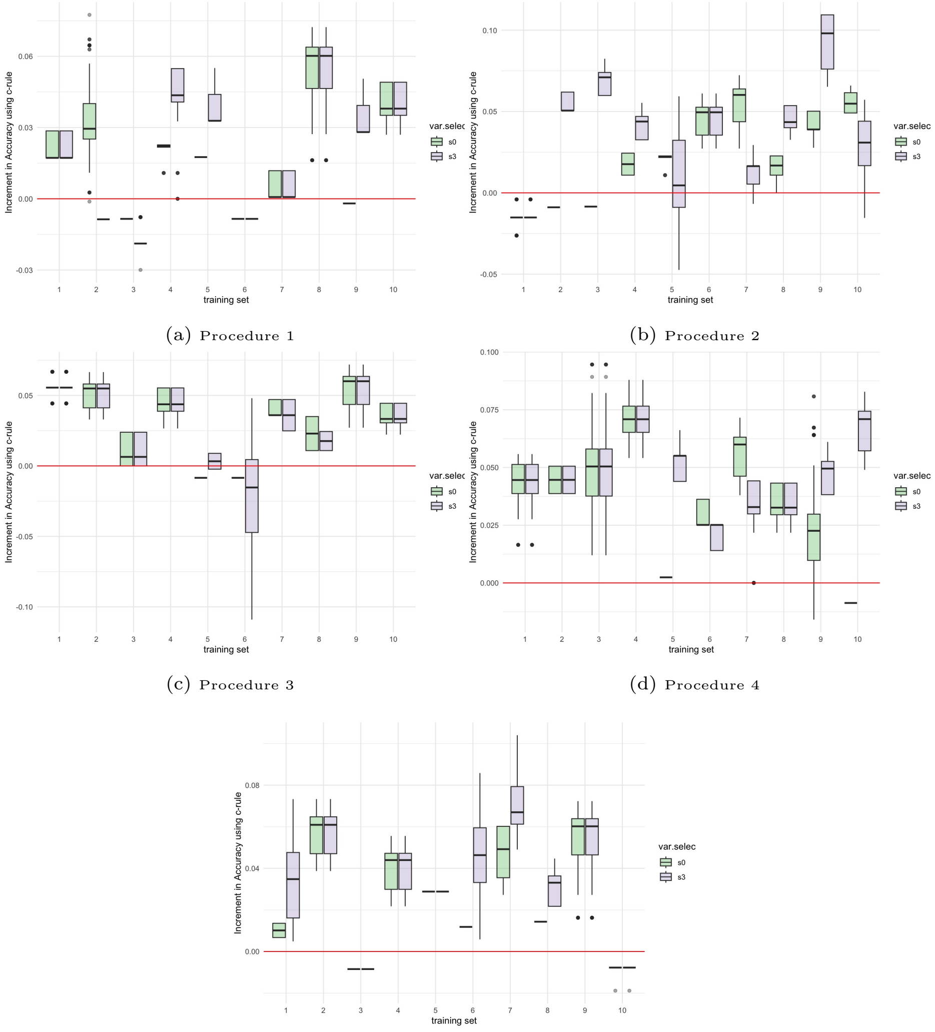
$$= m_-\, q^{m_--1}(1-q) \Big(\sum_{j=0}^{m_+} \binom{m_+}{j} p^j\,(1-p)^{m_+-j} \Big)$$

$$+ q^{m_-} \Big(\sum_{j=1}^{m_+} \binom{m_+}{j} p^j\,(1-p)^{m_+-j} \Big) = m_-\, q^{m_--1}(1-q)$$

$$+ q^{m_-} \Big(1 - (1-p)^{m_+} \Big) = q^{m_--1} \Big(m_-(1-q) + q\big(1-(1-p)^{m_+}\big) \Big).$$

Then, $(h_{2.1})$ coincides with $\varphi_-(M) \le \beta$, concluding the proof. □

Appendix B. Boxplots for the results in Subsection 4.2



(a) Procedure 1

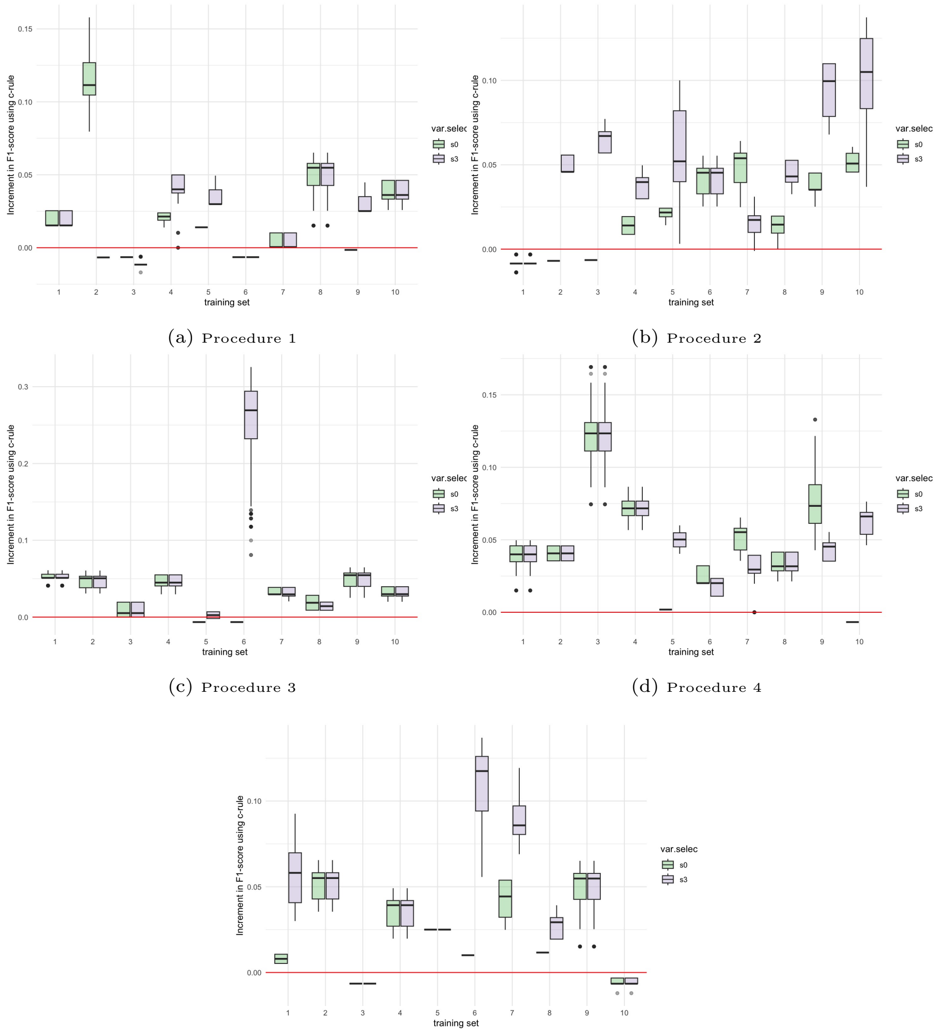
(b) Procedure 2

(c) Procedure 3

(d) Procedure 4

(e) Procedure 5

Figure Appendix B.1: Boxplots illustrating the increase in Accuracy when using the "c-rule" compared to the standard rule for the five different procedures in the 10-fold cross-validations partitions. The red line marks the zero level. Different colors are used to distinguish between the two `softmax` methods employed for variable selection prior to constructing the predictive `milr` model.



(a) Procedure 1

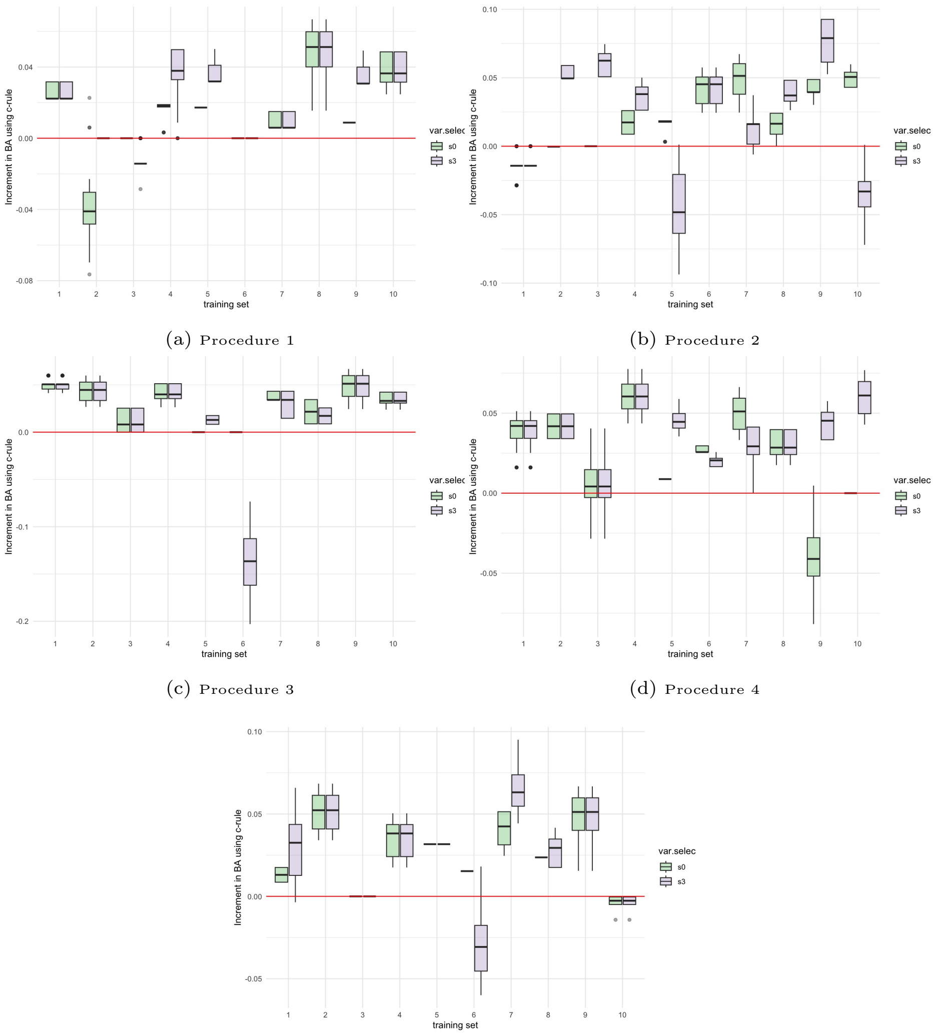
(b) Procedure 2

(c) Procedure 3

(d) Procedure 4

(e) Procedure 5

Figure Appendix B.2: Boxplots illustrating the increase in F1-score when using the "*c*-rule" compared to the standard rule for the five different procedures in the 10-fold cross-validations partitions. The red line marks the zero level. Different colors are used to distinguish between the two `softmax` methods employed for variable selection prior to constructing the predictive `milr` model.



(a) Procedure 1

(b) Procedure 2

(c) Procedure 3

(d) Procedure 4

(e) Procedure 5

Figure Appendix B.3: Boxplots illustrating the increase in BA when using the "*c*-rule" compared to the standard rule for the five different procedures in the 10-fold cross-validations partitions. The red line marks the zero level. Different colors are used to distinguish between the two `softmax` methods employed for variable selection prior to constructing the predictive `milr` model.

Appendix C. Disaggregated line plots by fold for the results in Subsection 4.3

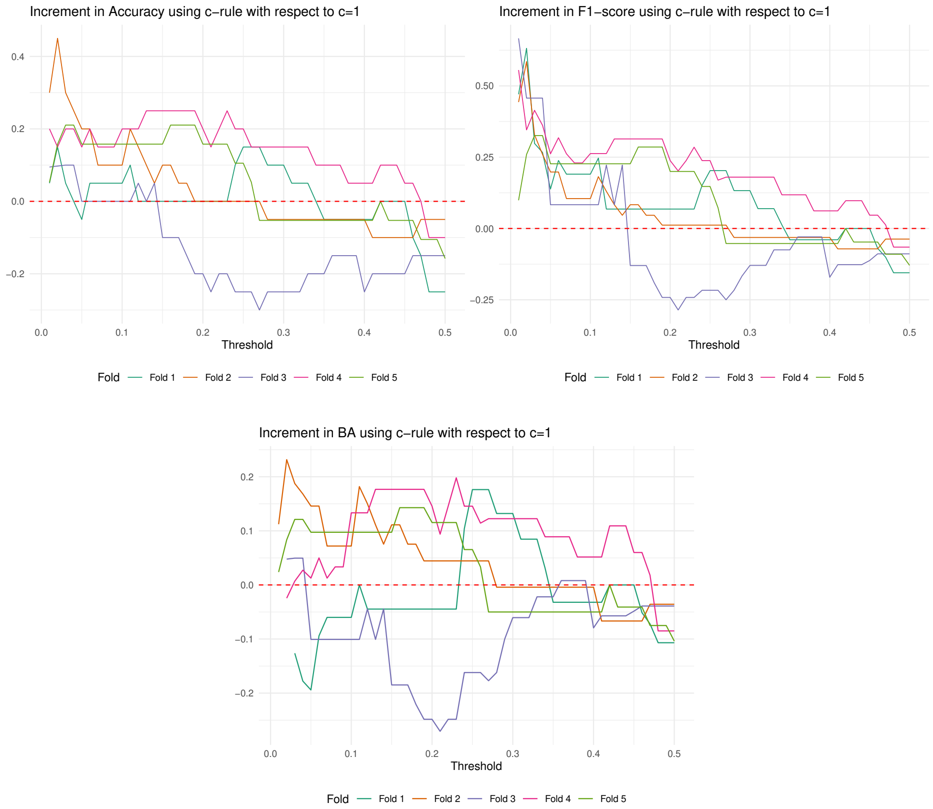


Figure Appendix C.1: Evolution of the increase in performance metrics when applying the "*c*-rule" compared to the standard rule across thresholds, disaggregated by folds. Each line represents one of the five folds, illustrating the trends observed in Figure 7 and emphasizing the less favorable behavior of fold 3.

References

[1] T. Dietterich, Lathrop, R., Lozano-Pérez, T. (1997) Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, 89, pp. 31–71. https://doi.org/10.1016/S0004-3702(96)00034-3

[2] E. Alpaydin, Cheplygina V., Loog M., Tax D.M.J. (2015) Single- vs. multiple-instance classification. Pattern Recognition, vol. 48, pp. 2831–2838. https://doi.org/10.1016/j.patcog.2015.04.006

[3] R. Delgado, Sánchez-Delgado, H. (2024) Multi-instance learning with application to the profiling of multi-victim homicides. Expert Systems with Applications, Volume 237, Part B, 2024, 121593, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2023.121593

[4] J. Amores (2013) Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence, vol. 201, pp. 81–105. https://doi.org/10.1016/j.artint.2013.06.003

[5] J. Foulds, Frank E. (2010) A review of multi-instance learning assumptions. The Knowledge Engineering Review, vol. 25(1), pp. 1–25. https://doi.org/10.1017/S026988890999035X

[6] M.-A. Carbonneau, Cheplygina, V., Granger, E., Gagnon, G. (2018) Multiple instance learning: A survey of problem characteristics and applications. Pattern Recognition, vol. 77, pp. 329–353. https://doi.org/10.1016/j.patcog.2017.10.00

[7] J. Li, Song, Y., Zhu, J., Cheng, L., Su, Y., Ye, L., Yuan. P., Han, S. (2021) Learning from large-scale noisy web data with ubiquitous reweighting for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43(5), pp. 1808–1814. https://ieeexplore-ieee-org.are.uab.cat/stamp/stamp.jsp?tp=&arnumber=8941250

[8] R. C. Bunescu, Mooney, R.J. (2007) Multiple Instance Learning for Sparse Positive Bags. In Proceedings of the 24th International Conference on Machine Learning (ICML), Corvallis, OR. pp 105–112. https://doi.org/10.1145/1273496.1273510

[9] C-K. Chiang, Sugiyama, M. (2023) Unified Risk Analysis for Weakly Supervised Learning (Preprint) 78 pages. https://doi.org/10.48550/arXiv.2309.08216

[10] S. Duffner, Garcia, D. (2020) Multiple Instance Learning for Training Neural Networks under Label Noise. In 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, pp. 1–7. https://doi.org/10.1109/IJCNN48605.2020.9206669

[11] I. Bhattacharya, Ye, Z., Pavani, K., Dasgupta, S. (2023) RT2S: A Framework for Learning with Noisy Labels. In Proceedings of the CIKM'23, pp. 5234–5235. https://doi.org/10.1145/3583780.3615996

[12] T. Leung, Song, Y., Zhang, J. (2011) Handling label noise in video classification via multiple instance learning. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, pp. 2056–2063. https://do.org/10.1109/ICCV.2011.6126479.

[13] X. Wang, Chi, X., Song, Y., Yang, Z. (2023) Active learning with label quality control. PeerJ Computer Science, vol. 9:e1480, 34 pages. https://doi.org/10.7717/peerj-cs.1480

[14] J. Luengo, Sánchez-Tarragó, D., Prati, R.C., Herrera, F. (2021) Multiple instance classification: Bag noise filtering from negative instance noise cleaning. Information Sciences, vol. 579, pp. 388–400. https://doi.org/10.1016/j.ins.2021.07.076

[15] C.G. Northcutt, Jiang, L., Chuang, I.L. (2021) Confident Learning: Estimating Uncertainty in Dataset Labels. Journal of Artificial Intelligence Research, vol. 70, pp. 1373–1411. https://www.jair.org/index.php/jair/article/view/12125/26676

[16] W. Li, Vasconcelos, N. (2015) Multiple instance learning for soft bags via top instances. In: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR'2015. https://ieeexplore-ieee-org.are.uab.cat/stamp/stamp.jsp?tp=&arnumber=7299056

[17] C.G. Northcutt, Athalye, A., Mueller, J. (2021) Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), 24 pages. https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/f2217062e9a397a1dca429e7d70bc6ca-Paper-round1.pdf

[18] W-C. Wang, Mueller, J. (2022) Detecting Label Errors in Token Classification Data (preprint), 15 pages. https://doi.org/10.48550/arXiv.2210.03920

[19] R.C. Mayo, Kent, D., Sen, L.C. et al. (2019) Reduction of False-Positive Markings on Mammograms: a Retrospective Comparison Study Using an Artificial Intelligence-Based CAD. J Digit Imaging, vol. 32, pp. 618–624. https://doi.org/10.1007/s10278-018-0168-6

[20] M. Imbriaco (2021) Reducing False-Positive Screening MRI Rates in Women with Extremely Dense Breasts. Radiology, vol. 301, pp. 293–294. https://doi.org/10.1148/radiol.2021211547

[21] T.H. Ho, Bissell, M.C.S., Kerlikowske, K, et al. (2022) Cumulative Probability of False-Positive Results After 10 Years of Screening With Digital Breast Tomosynthesis vs Digital Mammography. JAMA Netw Open. vol. 5(3):e222440. https://doi.org/10.1001/jamanetworkopen.2022.2440

[22] T. Amir, Hogan, M.P., Jacobs, St., Sevilimedu, V., Sung, J., Jochelson, M.S. (2022) Comparison of False-Positive Versus True-Positive Findings on Contrast-Enhanced Digital Mammography. American Journal of Roentgenology, vol. 218(5), pp. 797–808. https://doi.org/10.2214/AJR.21.26847

[23] K.R. Carter, Kotlyarov, E. (2007) Common Causes of False Positive F^{18} FDG PET/CT Scans in Oncology. Brazilian Archives of Biology and Technology, vol. 50, Special Number, pp. 29–35. ISSN 1516–8913. https://www.scielo.br/j/babt/a/w3gBQbXnLVMx6FmccqTyNnh/?lang=en#

[24] L.L. Plesner, Müller,F.C., Brejnebol, M.W., Laustrup, L.C., Rasmussen, F., Nielsen, O.W., Boesen, M., Andersen, M.B. (2023) Commercially Available Chest Radiograph AI Tools for Detecting Airspace Disease, Pneumothorax, and Pleural Effusion. Radiology, vol. 308(3). https://doi.org/10.1148/radiol.231236

[25] S. Yan, Zhu, X., Liu, G., Wu, J. (2016) Sparse multiple instance learning as document classification. Multimedia Tools and Appl., pp. 1–18. https://doi.org/10.1007/s11042-016-3567-z

[26] Q. Zang, Goldman, S.A., Yu, W., Fritts, J. (2002) Content-based image retrieval using multiple-instance learning. In Proceedings of 19th International Conference on Machine Learning (ICML-2002), pp. 682–689. https://dl.acm.org/doi/10.5555/645531.656002

[27] R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[28] Python Software Foundation. Python Language Reference. https://www.python.org

[29] P. Y. Chen, Chen, C. C., Yang, C. H., Chang, S. M., Lee, K. J. (2017) milr: Multiple-Instance Logistic Regression with Lasso Penalty. The R Journal, vol. 9(1), pp. 446–457. https://journal.r-project.org/articles/RJ-2017-013/RJ-2017-013.pdf

[30] X. Xu, Frank, E. (2004) Logistic Regression and Boosting for Labeled Bags and Instances. In H. Dai, R. Srikant, and Zang (Eds.): PAKDD 2004, LNAI 3056, pp. 272–281. https://link.springer.com/content/pdf/10.1007/978-3-540-24775-3_35.pdf

[31] S. Ray, Craven, M. (2005) Supervised Versus Multiple Instance Learning: An Empirical Comparison. In Proceedings of the 22th International Conference on Machine Learning, Bonn, Germany, pp. 697–704. https://icml.cc/Conferences/2005/proceedings/papers/088_Supervised_RayCraven.pdf

[32] W. Zhang, X. Zhang, X., Deng, H.-W., Zhang, M.-L. (2022) Multi-Instance Causal Representation Learning for Instance Label Prediction and Out-of-Distribution Generalization. Advances in Neural Information Processing Systems 35 (NeurIPS-2022).