Complex Localization in the Multiple Instance Learning Context

Răzvan-Alexandru Mariş, Dan-Ovidiu Graur, Rodica Potolea, Mihaela Dînşoreanu, Camelia Lemnaru

Technical University of Cluj-Napoca, Cluj-Napoca, Romania,
razvan.maris@student.utcluj.ro, dan.graur@student.utcluj.ro

Abstract. This paper introduces two techniques for solving multiple instance problems (MIP) where the instance localization assumption considered by classical MIP algorithms is not met. Our first technique applies a feature space transformation to meet the MIP localization assumption, while the second one identifies a region enclosing the majority class while excluding at least one instance from each positive bag (minority class). These new techniques are evaluated on synthetic datasets, as well as on a real-world dataset originated from a manufacturing process. The real-world dataset poses additional challenges: big data with noise, large imbalance and overlap.

Keywords: Multiple Instance Learning, Axis-Parallel Hyper-Rectangle, Feature Value Transformation, R-APR, Classification

1 Introduction

The aim of this work is to design a systematic strategy to detect faults in industrially manufactured entities. The real-world dataset originates from the traceability system of a *Printed Circuit Board* production line, therefore its dimension in considerably large ($\approx$ 320 gigabytes). Each entity is composed of a variable number of components. The characteristics of every component are known. After being manufactured, entities are labeled as functional or faulty by automatic inspection machines. An entity may be faulty due to one or more components. The task is to define a model that is able to identify non-functional entities. The difficulty arises due to the fact that faulty entities can contain both non-functional and functional components, without them being explicitly differentiated. This is known in literature as the *Multiple Instance Problem (MIP)*. In the current context, one expects the components rendering the entity to which they belong faulty (*positive instances*) to have atypical characteristics compared to functional components (*negative instances*). However, classical MIP algorithms attempt to find regularities amongst positive instances. In other words, MIP algorithms expect positive instances to be located in a small, dense region, while negative instances are supposed to be scattered around the feature space. In the studied problem, however, positive instances are situated in a large, less dense region (possibly scattered), while negative instances are located in a small, dense region. Although *outlier* or *novelty* detection techniques could be used under these circumstances, the region defining positive instances is not necessarily well determined. As such, our objective is to propose a novel set of methods through which such atypical MIP problems can be solved.

2 The Multiple Instance Problem

The MIP comes as a generalization to the classical Supervised Learning Problem [3], in that, training examples consist of groups of instances, where an instance is a feature vector. Each such group is known as a *bag*, and each such bag has an associated label. That is to say, labels are not directly associated to an instance, but rather to a group of instances. The concept which stands at the foundation of the MIP is known as *the standard MIP assumption* [12], or *linearity hypothesis* [5], which states that a *positive* bag has at least one positive instance, whilst a *negative* bag has no positive instances. The MIP is considerably more difficult than classical Supervised Learning [6,7], mainly due to the high degree of noise introduced in the learning process by the arbitrarily high number of positive instances a positive bag can have [7]. As such, specialized MIP algorithms need to be employed, to tackle the problem at hand.

Following, is a formal description of the standard MIP, based on the notation in [13]. Let $B = \{B_1, B_2, \ldots, B_m\}$ be a *set of* m *bags*, where $\forall B_i \in B\ \exists v_i \in \mathbb{N}^*$, such that $B_i = \{B_{i1}, B_{i2}, \ldots, B_{iv_i}\}$ is a bag containing v_i k-dimensional feature vectors. Let B_{ij} be the j^{th} instance of the i^{th} bag, such that $B_{ij} = \{B_{ij1}, B_{ij2}, \ldots, B_{ijk}\}$. Let $L = \{l_1, l_2, \ldots, l_m\}$ be the label set, and $l_i \in Y$ for $i = 1 \ldots m$. In the particular case of *binary classification*, which is the problem approached in this paper, $Y = \{\perp, \top\}$. Finally, let $D = \{\langle B_1, l_1\rangle, \langle B_2, l_2\rangle, \ldots, \langle B_m, l_m\rangle\}$ be the labeled data. The aforementioned standard MIP assumption can be formally represented as $l_i = l_{i1} \vee l_{i2} \vee \cdots \vee l_{iv_i}$, that is, a bag is positive if and only if it has at least one positive instance.

Whilst the *standard MIP* is arguably the most popular type of MIP, it is important to mention that the MIP context hosts a set of more complex challenges [12,1,2]. Weidmann et al. [12] produce a comprehensive taxonomy of the various types of MIPs, based on the existence of a multitude of underlying concepts, as opposed to a singular underlying concept which stands at the foundation of the *positive* class, as is the case in the standard MIP. These challenges are not to be further detailed here, since they are beyond the scope of this paper.

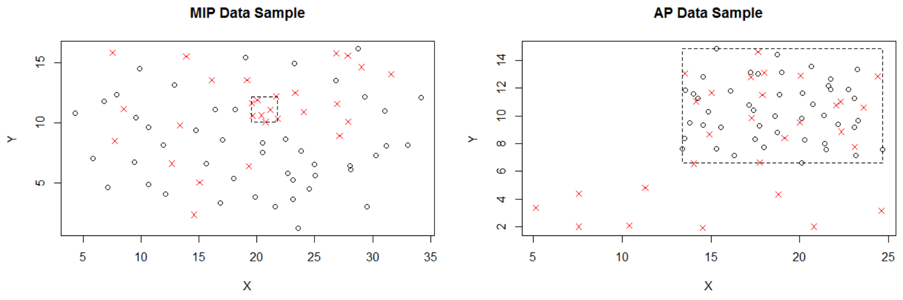
3 MIP Issues In The Current Context

The MIP de facto standard works under the assumption that positive instances converge towards a certain region, whilst negative instances are scattered around the feature space. However, in the given context, positive instances are scattered around the feature space, while negative instances cluster within a particular region. The ***A**ntisymmetry **P**roblem* (AP) is best described graphically by figure 1. It might be tempting to consider that a simple class label inversion solves this problem. However, this is not the case, since positive bags will now consist only of positive instances, while negative bags will contain both negative and positive instances. This goes against the assumptions made by existing MIP algorithms, and as a consequence, their learning process becomes biased. For instance, the Iterated Discrimination [6] algorithm requires only one instance from every positive bag to be included in the resulting Axis-Parallel Hyper-Rectangle (APR). However, after the label inversion, every instance belonging to a now positive bag is positive. Therefore, this algorithm would yield a high number of false negatives (or false positives, considering the initial labels). Another issue is the presence of positive instances in negative bags, as mentioned previously, which may prove problematic during the *feature selection* stage. Likewise, the DD metric [7], which stands at the foundation of the EM-DD algorithm [13], will require extensive modification in order to accommodate the existence of negative bag instances in high density areas. Thus, existing methods require either a preprocessing step or changes in their approach to allow them to tackle the AP.

3.1 A Feature-Value Transformation Based Approach

Our approach first transforms the feature space to meet the MIP instance localization assumption. Such a transformation is supposed to bring positive instances ”closer” together while scattering negative instances around the feature space. Such a transformation would apply a function $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, where n is the number of features, to all instances $\mathbf{x}$, replacing their feature vectors with $f(\overline{\mathbf{x}}, \mathbf{x})$, where $\overline{\mathbf{x}}$ is the mean of all instances belonging to negative bags:

$$\overline{\mathbf{x}} = \frac{\sum\limits_{\mathbf{x}\,\in\,N} \mathbf{x}}{|N|}, \text{where } N \text{ is the set of negative instances} \qquad (1)$$



(a) (b)

Fig. 1: × = positive bag instance and ○ = negative bag instance. (a) Positive instances converge. (b) Positive instances are scattered.

The function f must be chosen such that positive instances end up "closer" to $\overline{\mathbf{x}}$, while negative instances end up "further" from $\overline{\mathbf{x}}$, considering a metric for which a difference in only one dimension of the feature vectors is enough for the output to change considerably (e.g. the *Euclidean* metric). It must be noted that this specific transformation relies upon the fact that all negative instances are clustered. The dataset obtained after applying the transformation is then fed into the Iterated Discrimination algorithm [6].

It is worth mentioning, however, that the feature-value transformation employed here is *general purpose*, and as such, can be used on any antisymmetric dataset, whose initial structure is incompatible with the standard MIP algorithms, to convert it so that MIP learning methods can be applied.

3.2 An Axis-Parallel Hyper-Rectangle Based Approach

The second approach we propose towards solving the standard MIP in the Antisymmetric MIP Context is the *Reverse Axis-Parallel Hyper-Rectangle Algorithm* (R-APR). R-APR is inspired by the *Iterated Discrimination algorithm* [6]. It solves the standard MIP by finding an APR which, unlike the one resulted from the *Iterated Discrimination algorithm*, encloses all the negative bag instances and some of the instances of positive bags, leaving at least one positive bag instance outside. As such, a bag is classified as positive, if at least one of its instances falls outside of the APR along at least one dimension, whilst a bag is classified as negative if all its instances fall within the APR's bounds for all dimensions.

The algorithm consists of four major stages: All-Negative APR Generation, High Density Positive Instance Margin Expansion, Feature Selection, and finally, Statistical Margin Expansion. The R-APR algorithm attempts to solve the AP without employing any sort of feature-value transformations, other than normalization. Moreover, the APR produced by this algorithm yields valuable information in terms of what the normal value ranges for the relevant features are. Consequently, in certain contexts, such as that of industrial manufacturing, it provides potentially useful insight into the production process.

4 Solving The Antisymmetry Problem

This section provides a more in depth description of the two original approaches we propose towards solving the AP problem in the MIP.

4.1 The Transformation-Based Iterated Discrimination Algorithm

This approach requires the definition of a function as described in section 3.1. Every instance is then replaced with $f(\overline{\mathbf{x}}, \mathbf{x})$, with the purpose of bringing positive instances "closer" to $\overline{\mathbf{x}}$ while moving negative instances "further" from $\overline{\mathbf{x}}$. An example of such a function f is:

$$f(\overline{\mathbf{x}}, \mathbf{x}) = \overline{\mathbf{x}} + \frac{\mathbf{x} - \overline{\mathbf{x}}}{\|\mathbf{x} - \overline{\mathbf{x}}\|} \cdot g(\|\mathbf{x} - \overline{\mathbf{x}}\|), \tag{2}$$

where $\|\cdot\|$ is the *Euclidean* norm and $g : \mathbb{R} \to \mathbb{R}$ is a monotonically decreasing function. The function g can be defined independently of the number of features of the dataset, but then $\|\mathbf{x} - \overline{\mathbf{x}}\|$ must be scaled accordingly. Therefore $g(\|\mathbf{x} - \overline{\mathbf{x}}\|)$ from equation (2) should be replaced with $g\left(\frac{\|\mathbf{x}-\overline{\mathbf{x}}\|}{\sqrt{n}}\right)$. This is because the *Euclidean metric* of an n-dimensional vector, whose components are all equal to a, is $\sqrt{n} \cdot a$. That is, $\|(a, a, \ldots, a)\| = \sqrt{n} \cdot a$. The *Euclidean* norm is used so that one feature value being ”far” from that feature's mean suffices for the instance to be brought ”closer” to $\overline{\mathbf{x}}$.

Figure 3 contains plots of one family of functions which meet the above requirements, described by:

$$\mathbb{G} = \left\{ g : \mathbb{R} \to \mathbb{R} \mid g(x) = c \cdot a^{-b \cdot x} \right\}, \text{ where } \ a, b, c \in \mathbb{R}_{>0}. \tag{3}$$

An exponential family of functions was chosen because the absolute value of their derivative can be made large enough so as to achieve a substantial separation margin between positive and negative instances, regardless of the initial value of this margin. Furthermore, the behavior of these functions in the proximity of 0 can be constrained. The *fixed points* (x_0, y_0) of these functions are marked at the intersection of the vertical line $x = x_0$ with the functions' plots. Considering equation (2), these fixed points and the value of $\|\mathbf{x} - \overline{\mathbf{x}}\|$ determine whether $\mathbf{x}$ ends up closer or further from $\overline{\mathbf{x}}$.

Figure 2 shows the effect of applying (2) to a normally distributed two-dimensional dataset. The function g is replaced in (2), in turn, by the functions displayed in figure 3.

4.2 The R-APR Algorithm

The R-APR algorithm consists of the four steps shown in figure 4, excluding the data normalization stage, which is optional. The four steps are presented in the subsections that follow.

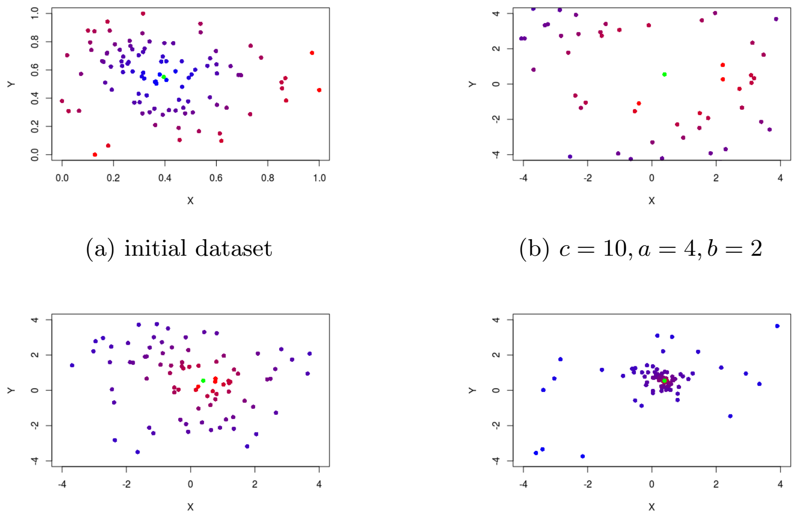
All-Negative APR Generation This APR defines a region in feature space which encloses all negative instances. The upper margins of the APR, along every relevant feature d, are defined as:

$$ub_d = \max_{B_i \in B^-, B_{ij} \in B_i} (B_{ijd}) \tag{4}$$

Respectively, the lower bounds are obtained using:

$$lb_d = \min_{B_i \in B^-, B_{ij} \in B_i} (B_{ijd}) \tag{5}$$

Due to the standard MI assumption, the generated APR is not yet ready to be used for classification, since positive bags still have negative instances, which may be outside the All-Negative APR. During this stage, only negative bags are processed.



(a) initial dataset

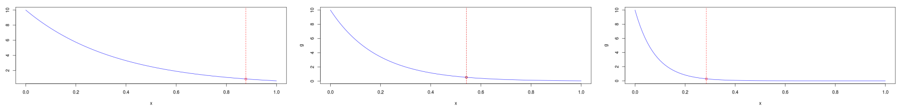
(b) $c = 10, a = 4, b = 2$

(c) $c = 10, a = 6, b = 3$

(d) $c = 10, a = 8, b = 6$

Fig. 2: Figure 2a represents a normally distributed, two-dimensional dataset. Instances that are "closer" to the *mean point* (green) are colored in *blue*, while instances that are "further" from it are colored in *red*. Figures 2b, 2c and 2d illustrate the original dataset transformed using equation (2), where the function g belongs to the function family described in equation (3).

High Density Positive Instance Margin Expansion To generalize better, the APR must be expanded in such a way as to include negative instances belonging to positive bags. However, due to the asymmetry [8,7,3] introduced by the bag level label, identifying them is not straightforward. We propose two solutions towards solving this problem, based on the assumption that negative instances from positive bags are gathered together, since they should have similar feature values. Both procedures are based on density and distance measurements.



(a) $c = 10, a = 4, b = 2$

(b) $c = 10, a = 6, b = 3$

(c) $c = 10, a = 8, b = 6$

Fig. 3: Plots of functions belonging to the family defined in 3. The vertical lines mark the fixed points of the functions. In this context, the fixed point discriminates between instances $\mathbf{x}$ which end up "closer" and "further" from $\overline{\mathbf{x}}$.

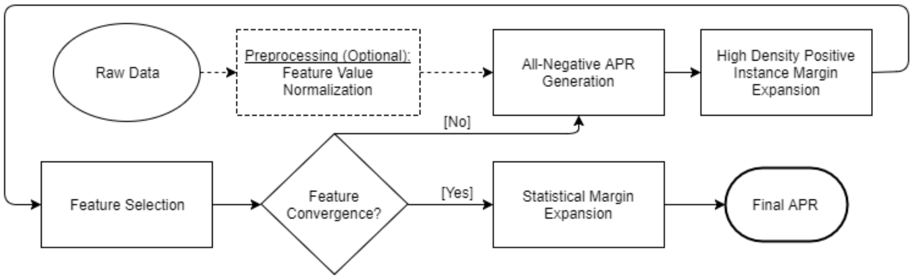


Fig. 4: The general execution flow of the R-APR algorithm.

The first approach refers to selecting one instance from each positive bag, thus constructing a set of instances which are used towards building an auxiliary APR. The new APR is used to expand the All-Negative APR, where necessary. The instance is chosen based on a Density measurement, which computes the instance's degree of proximity to the other instances belonging to the same bag:

$$HD_i = \max_{B_{ij} \in B_i} \left(\sum_{k, k \neq j} \frac{1}{\zeta + \|B_{ij} - B_{ik}\|^2} \right) \tag{6}$$

where $\|\cdot\|$ is the Euclidean Metric, $\zeta \in \mathbb{R}$ is an offset, and HD_i is the highest density instance of B_i. Additionally, the instance's distance from the All-Negative APR, is computed, using the Manhattan Distance:

$$dist_{ij} = \sum_{d} f(B_{ijd}, lb_d, ub_d) \tag{7}$$

where function f is computed as:

$$f(x, lb, ub) = \begin{cases} lb - x, & \text{if } x < lb \\ x - ub, & \text{if } x > ub \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

During this stage, the algorithm attempts to find regions towards which it expands the margins of the All-Negative APR, by essentially *speculating* which regions host a large number of negative instances belonging to positive bags. As such, one must ensure that the APR is not wrongly expanded towards regions of positive instances, as may be the case when positive bags have few negative instances. In order to avoid such cases, the algorithm requires two user-defined thresholds, concerning *density* and *distance*, empirically identified and tuned for every data set. An instance is only selected if its density is above the density threshold, and if its distance from the APR is below the distance threshold.

The second approach we propose comes as an extension to the previously described technique. After determining the highest density instance for a bag, using equation (6), an optimization algorithm is used to find the point which maximizes the density function. The search is bounded by a rectangular region defined by the bag's instances. Once more, the user-defined distance and density thresholds are used when selecting the instances. The set obtained as a result of the selection procedure is used to construct a new APR, with the aim of expanding the All-Negative APR's bounds where needed.

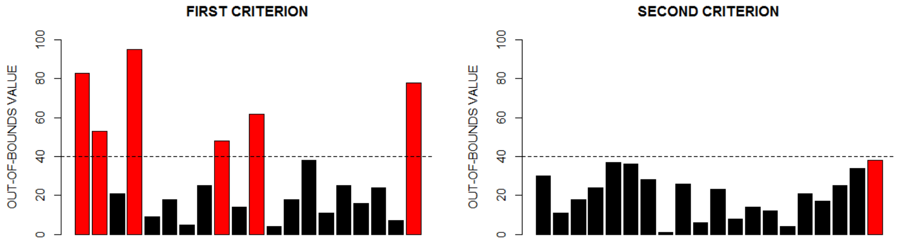
Feature Selection Similarly to the Iterated Discrimination algorithm [6], the R-APR algorithm attempts to select the relevant features in an *iterative* fashion. However, unlike the Iterated Discrimination algorithm, discrimination is performed on the positive bags and at the bag-level.

There are two criteria for establishing when a feature discriminates a bag, both dependent on a user-specified global out-of-bounds threshold $t \in \mathbb{R}_{\geq 0}$. Since discrimination is performed at the bag level, a bag's out-of-bounds value for a particular feature d is given by $val_d = \max_{B_{ij} \in B_i}(out\_of\_bounds(B_{ijd}, lb_d, ub_d))$. The *first criterion* specifies that a feature d discriminates a bag B_i if $val_d > t$. The *second criterion* specifies that a feature d discriminates a bag B_i if $val_d > val_k \ \forall k \in \mathcal{F}_r, d \neq k$, where $\mathcal{F}_r$ is the relevant feature set. Figure 5 describes these concepts visually.

Following is a formal description of the Feature Selection stage: let $\mathcal{F}_r^{old} = \{f_1, f_2, \ldots, f_n\}$ be the old set of relevant features. Let $\mathcal{F}_r^{new} = \emptyset$ be the new set of relevant features, initially empty, and let $B_{FS}^+ = B^+$ be the set of positive bags used in the current feature selection stage. As previously mentioned, the Feature Selection stage is iterative. Let f_i' be the most discriminating feature, i.e. the feature which discriminates the most bags in B_{FS}^+, as identified in iteration i of this stage. Let $B_{f_i'}^+$ be the set of positive bags discriminated by f_i'. It follows that $\mathcal{F}_r^{old} = \mathcal{F}_r^{old} \setminus \{f_i'\}$, and $\mathcal{F}_r^{new} = \mathcal{F}_r^{new} \cup \{f_i'\}$. Moreover, $B_{FS}^+ = B_{FS}^+ \setminus B_{f_i'}^+$. This stage will continue to loop, until either $B_{FS}^+ = \emptyset$ or $\mathcal{F}_r^{old} = \emptyset$. In the former case, the algorithm loops back to **All-Negative APR Generation**, with $\mathcal{F}_r^{new}$ as the set of relevant features. In the latter case, the feature set converges, and the algorithm moves on to the next, and final stage.

Statistical Margin Expansion The final stage of the R-APR algorithm refers to expanding the margins of the APR obtained so far, to generalize better. It is identical to that employed in the Iterated Discrimination algorithm [6], however, the *Kernel Density Estimation* (KDE) is built from negative bag instances, as opposed to positive bag instances. This stage is controlled by two user-defined constants: ε and τ. The τ constant specifies the amount of probability which should fall within the bounds of the APR, based on the KDE, if the negative bag instances were centered between its bounds. The value of τ is used to establish the deviation σ_d along relevant feature d, such that a normal distribution centered in $\mu_d = \frac{lb_d + ub_d}{2}$ with deviation σ_d hosts τ probability between the upper and lower bounds. This can be formally expressed as $\Pr(lb_d < X < ub_d) = \tau$.

A Gaussian KDE is built for each relevant feature d. Next, relative to the obtained KDEs, the margins of the APR along every relevant feature are expanded so as to ensure that $\frac{\varepsilon}{2}$ probability remains above the upper bound, and that $\frac{\varepsilon}{2}$ remains below the lower bound.



(a)

(b)

Fig. 5: An example of the two Feature Selection criteria. Each chart contains 20 features (vertical bars), the threshold t is set to 40. Bars represent a bag's maximal out-of-bounds value along that feature. Red bars represent features that discriminate the bag. Black bars represent features which do not discriminate the bag. (a) Example of the first criterion. (b) Example of the second criterion.

This step is, however, optional, and can be removed when the APR should be tighter. Such cases are largely identified empirically, and are typically due to a high number of positive instances being located near the bounds, outside of the unexpanded APR. Through expansion, these instances are included in the APR, which likely leads to an undesirably high rate of false-negatives.

Classification Classification assumes a series of comparisons against the bounds of the APR. As such, for an unseen bag, if at least one instance falls outside the APR, along at least one dimension, then that bag is classified as positive. Otherwise, the bag is classified as negative.

5 Experimental Procedures and Results

The APR algorithm, with and without applying the feature transformation described in section 4.1, as well as the R-APR algorithm were tested on both synthetic and real-world datasets. All synthetic datasets have 100 positive bags and 100 negative bags, while individual instances have 3 features. Negative bags contain 10 to 19 negative instances. Positive bags contain 10 to 16 negative instances and 3 to 5 positive instances. Negative instances belong to a 3-dimensional *Gaussian* distribution with $\mu = [10\ 10\ 10]$, with no feature correlation, each having a standard deviation $\sigma = 5$. Positive instances differ only by having $\mu = [\alpha\ \alpha\ 10]$, where α is different for each dataset, namely $\alpha \in \{13, 16, 19, 22.5, 27.5\}$.

The results on artificial data are shown in Table 1. They reveal that the Iterated Discrimination algorithm, with or without Feature Value Transformation, yields a high recall, regardless of the level of separation. This would suggest that

Iterated Discrimination is effective in discerning the minority class, even in cases of partial overlap. Precision and accuracy increase proportionally with the level of separation between positive and negative instances, for the version of the algorithm which uses the transformation. These values are upward of 90% for higher levels of separation. Iterated Discrimination, without transformation, produces relatively constant values for both precision and accuracy of around 50%. Its low precision is due to the fact that the generated APR is likely located in the region hosting the negative instances of the positive bags. Consequently, negative bags are expected to have instances located within this APR, thus resulting in a large number of negative bags being falsely classified as positive.

The R-APR algorithm produces increasingly better values for recall and accuracy, as the level of separation between the positive and negative instances increases. Its behavior is similar to that of Iterated Discrimination with transformation, requiring, however, a greater degree of distinction between instances in order to produce good results.

Table 2 presents the results obtained on real-world data, collected from a real-world industrial PCB manufacturing process, and consisting of 100 positive bags, and 800 negative bags. Each individual instance has 25 features. The Feature Value Transformation helps increase the precision, as well as the classification accuracy. The recall, however, remains relatively unchanged. We believe this to be a consequence of data overlapping along all dimensions, which continues to remain overlapped even after applying the transformation, thus maintaining the *false-negative* rate. This would suggest that supplementary features are needed.

All evaluations have been performed using a 5-fold *cross-validation*. The feature transformation, when employed, used a function g as described in equation (3), with $a = 10,\ b = 8,\ c = 6$.

Table 1: Artificial Dataset Results

Method	Mean (α)	TP	FN	TN	FP	Precision	Recall	Accuracy
Iter. Discrim. with Transf.	13.0	91	9	19	81	52.9%	91.0%	55%
Iter. Discrim. without Transf.	13.0	87	13	14	86	50.2%	87.0%	50.5%
R-APR	13.0	0	100	100	0	-	0%	50%
Iter. Discrim. with Transf.	16.0	89	11	37	63	58.5%	89.0%	63%
Iter. Discrim. without Transf.	16.0	95	5	6	94	50.2%	95.0%	50.5%
R-APR	16.0	0	100	100	0	-	0%	50%
Iter. Discrim. with Transf.	19.0	90	10	95	5	94.7%	90%	92.5%
Iter. Discrim. without Transf.	19.0	93	7	8	92	50.2%	93%	50.5%
R-APR	19.0	12	88	100	0	100%	12%	56%
Iter. Discrim. with Transf.	22.5	95	5	98	2	97.9%	95%	96.5%
Iter. Discrim. without Transf.	22.5	91	9	13	87	51.1%	91%	52%
R-APR	22.5	47	53	100	0	100%	47%	73.5%
Iter. Discrim. with Transf.	27.5	94	6	97	3	96.9%	94%	95.5%
Iter. Discrim. without Transf.	27.5	93	7	4	96	49.2%	93%	48.5%
R-APR	27.5	89	11	100	0	100%	89%	94.5%

Table 2: Real-World Dataset Results

Method	TP	FN	TN	FP	Precision	Recall	Accuracy
Iter. Discrim. with Transf.	77	23	784	16	82.7%	77%	95.6%
Iter. Discrim. without Transf.	81	19	644	156	34.1%	81%	80.5%

6 Existing Standard MIP Solutions

The Iterated Discrimination algorithm [6] is one of the fundamental means for tackling the MIP. It attempts to find an APR, such that any bag having an instance within its bounds, is classified as *positive*. Otherwise, the bag is considered *negative*. The method is based on the idea that positive instances have similar features, thus, converging towards a particular region in feature space.

The EM-DD algorithm [13] is another method employed towards solving the MIP. The algorithm is based on the same supposition that positive instances converge towards a particular region. Consequently, it searches for a target point in feature space around which the positive instances are assumed to gather. A bag is classified as positive if at least one of its instances neighbors the aforementioned target. The algorithm is based on the Diverse Density metric [7], which yields high values for hypothesized points in regions containing a large number of instances from diverse positive bags, and low values for regions containing instances from negative bags, or little diversity in terms of positive bags.

Numerous other approaches have been employed towards solving the standard MIP, including: Neural Networks [8], Support Vector Machines [3,4], density based approaches [7], Lazy Learning [11], Decision Trees, or Rule Sets [5]. Solutions to the standard MIP have also been explored in the context of real-valued labels through methods such as Multiple Instance Regression [10]. MIP algorithms can be applied in many areas, including: image classification, stock prediction, biochemistry, or text classification. However, empirical studies suggest that no particular Multiple Instance learning algorithm appears to perform successfully in every possible problem domain [9]. That is, MIP algorithms vary in performance, depending on the problem they attempt to solve. It is worth emphasizing that unlike the R-APR algorithm, these existing methods are unsuitable for directly solving the AP, without prior Feature Value Transformation.

7 Conclusions

The paper presents two strategies for tackling the MIP in an antisymmetric complex context, where the positive instances are located in a larger, less dense area, whilst the negative instances converge towards a particular region. The first technique refers to applying a feature-value transformation in order for the data to become compatible with standard MIP algorithms. The second technique refers to a novel algorithm, the Reverse Axis-Parallel Hyper-Rectangle (R-APR) algorithm, designed to identify a region in feature space which encloses all the negative bags, whilst excluding at least one instance from every positive bag. The strategies proved to be effective, with good performance on synthetic data (over 90% recall). The same performance measure (recall) on the real-world data is rather modest (77%). A parallel study we conducted showed that the real-world data suffers from large overlap, which makes the classes partly indistinguishable with the current set of available features. This would suggest that new features need to be identified, and extracted from the real-world manufacturing process. We are currently working on improvements of the two strategies.

References

1. Alpaydin, E., Cheplygina, V., Loog, M., Tax, D.M.: Single- vs. multiple-instance classification. Pattern Recogn. **48**(9) (September 2015) 2831–2838
2. Amores, J.: Multiple instance classification: Review, taxonomy and comparative study. Artif. Intell. **201** (August 2013) 81–105
3. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Proceedings of the 15th International Conference on Neural Information Processing Systems. NIPS'02, Cambridge, MA, USA, MIT Press (2002) 577–584
4. Bunescu, R.C., Mooney, R.J.: Multiple instance learning for sparse positive bags. In: Proceedings of the 24th International Conference on Machine Learning. ICML '07, New York, NY, USA, ACM (2007) 105–112
5. Chevaleyre, Y., Zucker, J.D.: Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. application to the mutagenesis problem. In: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence. AI '01, London, UK, UK, Springer-Verlag (2001) 204–214
6. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. **89**(1-2) (January 1997) 31–71
7. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10. NIPS '97, Cambridge, MA, USA, MIT Press (1998) 570–576
8. Ramon, J., Raedt, L.D.: Multi instance neural networks. In: Proceedings of ICML-2000, Workshop on Attribute-Value and Relational Learning. (2000)
9. Ray, S., Craven, M.: Supervised versus multiple instance learning: An empirical comparison. In: Proceedings of the 22Nd International Conference on Machine Learning. ICML '05, New York, NY, USA, ACM (2005) 697–704
10. Ray, S., Page, D.: Multiple instance regression. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 425–432
11. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proceedings of the Seventeenth International Conference on Machine Learning. ICML '00, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2000) 1119–1126
12. Weidmann, N., Frank, E., Pfahringer, B. In: A Two-Level Learning Method for Generalized Multi-instance Problems. Springer Berlin Heidelberg, Berlin, Heidelberg (2003) 468–479
13. Zhang, Q., Goldman, S.A.: EM-DD: An improved multiple-instance learning technique. In: In Advances in Neural Information Processing Systems, MIT Press (2001) 1073–1080