International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1483–1497
DOI: https://doi.org/10.2991/ijcis.d.200910.002; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Contextualizing Support Vector Machine Predictions

Marcelo Loor[1,2,*], Guy De Tré[1]

[1]*Department of Telecommunications and Information Processing, Ghent University, Sint-Pietersnieuwstraat 41 B-9000, Ghent, 9000, Belgium*
[2]*Department of Electrical and Computer Engineering, ESPOL Polytechnic University, Campus Gustavo Galindo V., Km. 30.5 Via Perimetral, Guayaquil, 09015863, Ecuador*

ARTICLE INFO

Article History
Received 10 May 2020
Accepted 06 Sep 2020

Keywords
Explainable artificial intelligence
Augmented appraisal degrees
Context handling
Support vector machine classification

ABSTRACT

Classification in artificial intelligence is usually understood as a process whereby several objects are evaluated to predict the class(es) those objects belong to. Aiming to improve the interpretability of predictions resulting from a *support vector machine* classification process, we explore the use of *augmented appraisal degrees* to put those predictions in context. A use case, in which the classes of handwritten digits are predicted, illustrates how the interpretability of such predictions is benefitted from their contextualization.

1. INTRODUCTION

As the ubiquity of *artificial intelligence* (AI) grows, computer applications like word processors that translate documents, or videoconference applications that generate transcripts of meetings, are thoroughly satisfying business or user needs. Nevertheless, AI applications like profiling tools that predict capabilities of people without providing any explanation have to be banned in situations where transparency and accountability are mandatory [1,2]. An existing challenge in this regard is to find suitable mechanisms by which the reasons and reasoning behind computer predictions involving complex techniques can be explained with ease [3,4].

To address that challenge in predictions made by a *support vector machine* (SVM) classification process [5,6], we explore the use of *augmented appraisal degrees* (AADs) [7] for the contextualization of the evaluations that yield such predictions. Since an AAD has been conceived as a mathematical representation of a connotative meaning in an *experience-based evaluation*, it can be used for recording not only the level to which an object belongs (or not) to a particular class, but also the object's features that support that level assignment. Hence, we propose a novel variant of an SVM classification process whereby the resulting predictions are augmented in such a way that those predictions are put in context and an explanation is provided. Our main motivation here is to obtain predictions that expose the aspects deemed to be relevant to the classification.

An important facet of the proposed variant is that, by explicitly representing context, it yields predictions that are better interpretable. Hence, our variant, named *explainable SVM classification* (XSVMC), can be used within an *explainable artificial intelligence* (XAI) system [8], by which users can take advantage of such interpretable predictions to make better informed decisions.

A key component of XSVMC is a novel evaluation procedure in which the most influential support vector (MISV) is used for identifying what has been relevant to the classification. This evaluation procedure, which is the main contribution of this work, contextualizes the evaluations in such a way that the forthcoming predictions can be explained with ease.

To describe how XSVMC works, we develop a process whereby handwritten numbers are evaluated to predict the class(es) those handwritten numbers belong to. A visual representation of a resulting prediction is shown in Figure 1: while the left side of this figure shows a handwritten number, which is used as input, the right side of the figure shows a representation of why the proposition "the handwritten number is a '3' " is true up to a specific level. The resulting prediction has also been used within an XAI system to produce the following output: "The green part suggests that the drawing is a '3' with a computed grade of 0.16; yet, the red part, which a '3' should have, and the gray part, which a '3' should not have, indicate that it is not a '3' with a computed grade of 0.64." Notice in this example that the output not only indicates why a proposition (or prediction) is true, but also why it is not. This provides the system and users with extra information and illustrates an advantage of including explainability into AI systems.

* *Corresponding author. Email: Marcelo.Loor@UGent.be*
This paper is an extended version of the work published in Marcelo Loor and Guy De Tré. Explaining Computer Predictions with Augmented Appraisal Degrees. *Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*, pages 158–165. Atlantis Press, 2019/08. https://doi.org/10.2991/eusflat-19.2019.24

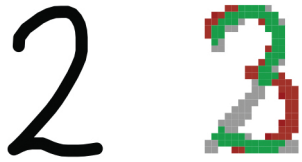

Figure 1 | Predicting handwritten numbers.

In the next section, we introduce the AAD concept and briefly describe how it can be integrated into the *intuitionistic fuzzy set* (IFS) concept. Then, we provide a comprehensive explanation of our novel XSVMC in Section 3 and illustrate in Section 4 how to use it. After that, other existing techniques for explaining individual predictions are reviewed in Section 5. We conclude the paper in Section 6.

2. PRELIMINARIES

As indicated previously, classification in AI is commonly understood as a process in which several objects are evaluated in order to predict the class(es) those objects belong to [9]. In this regard, a classification algorithm can look into the features of an object to evaluate the level to which this object is member of one or more well-known classes Using these evaluations, the algorithm can provide the best evaluated class(es) as a prediction. It is worth mentioning that herein by *'feature'* is meant a distinctive aspect that is relevant for the classification. For instance, the level of illumination of either one pixel or a group of pixels of the handwritten number shown on the left side of Figure 1 can be deemed to be relevant for the classification of this number.

In situations where an object, say x, has features suggesting a partial membership of this object in a given class, say A, the aforementioned classification algorithm can use the framework of *fuzzy set theory* [10] to model the evaluation of the level to which x belongs to A. In that framework, the evaluation of a proposition having the canonical form 'x BELONGS TO A', meaning x is member of A, can mathematically be denoted by a *membership grade*. A membership grade is a number $\mu_A(x)$ in the unit interval $[0, 1]$, where 0 and 1 represent respectively the lowest and the highest membership grades. For instance, if x represents the handwritten number shown on the left side of Figure 1 and A denotes (what has been learned about) the class of handwritten 3's, then $\mu_A(x) = 0.16$ indicates the level to which this handwritten number belongs to the class of handwritten 3's. Moreover, if B represents the class of handwritten 2's and $\mu_B(x)$ denotes the level to which x belongs to B, then $\mu_A(x) < \mu_B(x)$ means that the level to which x belongs to the class of handwritten 3's is less than the level to which x belongs to the class of handwritten 2's. In this manner, the classification algorithm can perform a numeric comparison to determine what class should be offered as a prediction.

As shown in Figure 1, the handwritten number can also have features suggesting that it does not belong to the class of handwritten 3's. – see, e.g., the right side of Figure 1 in which the gray and the red parts suggest the handwritten number is not a '3'. In this case, the evaluation of the proposition 'x BELONGS TO A' can be better described in the IFS framework [11,12] by means of an IFS element. An IFS element, say $\langle x, \mu_A(x), \nu_A(x)\rangle$, consists of the evaluated object x, a *membership grade* $\mu_A(x)$ and a *nonmembership grade* $\nu_A(x)$, where $\mu_A(x), \nu_A(x) \in [0, 1]$ must satisfy the consistency condition $0 \leqslant \mu_A(x) + \nu_A(x) \leqslant 1$. For example, the proposition "the handwritten number depicted on the left side of Figure 1 is a '3'" can be represented by the canonical form 'x BELONGS TO A', where x and A denote in that order the handwritten number and the class of handwritten 3's; thus, the evaluation of this proposition can be denoted by the IFS element $\langle x, \mu_A(x), \nu_A(x)\rangle = \langle x, 0.16, 0.64\rangle$. In addition, the *buoyancy* [13] of this IFS element, i.e., $\rho_A(x) = \mu_A(x) - \nu_A(x)$ can be used for comparing IFS elements to each other. For example, if the IFS element $\langle x, \mu_B(x), \nu_B(x)\rangle$ represents the evaluation of the proposition "the handwritten number depicted on the left side of Figure 1 is a '2'" then $\rho_A(x) < \rho_B(x)$ means that the level to which x belongs to the class of handwritten 3's is less than the level to which x belongs to the class of handwritten 2's. Such a comparison can be used by a classification algorithm for making a prediction.

As noticed above, while a membership grade and an IFS element make it possible to record the level(s) to which an object belongs (or not) to a given class, none of these representations enables the recording of the object's characteristics that lead to and hence explain this (these) level(s). To record those characteristics, the idea of AADs [7] has been introduced. An AAD of an object x, say $\hat{\mu}_{A@K}(x)$, can be seen as a pair $\langle \mu_{A@K}(x), F_{\mu_{A@K}}(x)\rangle$ that denotes the level $\mu_{A@K}(x)$ to which x belongs to the class A, as well as the particular collection $F_{\mu_{A@K}}(x)$ of x's features that are deemed to be relevant to the evaluation according to the knowledge K. For instance, the evaluation depicted on the right side of Figure 1 can be denoted by the AAD $\hat{\mu}_{A@K}(x) = \langle 0.16, F_{\mu_{A@K}}(x)\rangle$, where: (i) x and A represent the handwritten number on the left of Figure 1 and a class of handwritten 3's respectively; (ii) K symbolizes the knowledge about handwritten 3's used for the evaluation of x; and (iii) $F_{\mu_{A@K}}(x)$ represents a collection consisting of the green pixels that indicate why x should be a '3' according to K.[1]

To record the characteristics that indicate why the aforementioned handwritten number should not be a '3', the augmentation of IFS elements with AADs has been proposed [7]. An augmented IFS element, say $\langle x, \hat{\mu}_{A@K}(x), \hat{\nu}_{A@K}(x)\rangle$, consists of a membership AAD, $\hat{\mu}_{A@K}(x)$, and a nonmembership AAD, $\hat{\nu}_{A@K}(x)$: while $\hat{\mu}_{A@K}(x) = \langle \mu_{A@K}(x), F_{\mu_{A@K}}(x)\rangle$ indicates the level $\mu_{A@K}(x)$ to which x belongs to A and the collection $F_{\mu_{A@K}}(x)$ of x's features considered for quantifying this *membership* level, $\hat{\nu}_{A@K}(x) = \langle \nu_{A@K}(x), F_{\nu_{A@K}}(x)\rangle$ indicates the level $\nu_{A@K}(x)$ to which x does not belong to A and the collection $F_{\nu_{A@K}}(x)$ of x's features considered for quantifying this *nonmembership* level. For instance, keeping x, A, K and $F_{\mu_{A@K}}(x)$ as given in the previous example, one can represent the evaluation depicted in Figure 1 by $\langle x, \hat{\mu}_{A@K}(x), \hat{\nu}_{A@K}(x)\rangle = \langle x, \langle 0.16, F_{\mu_{A@K}}(x)\rangle, \langle 0.64, F_{\nu_{A@K}}(x)\rangle\rangle$,where $F_{\nu_{A@K}}(x)$ represents a collection consisting of the red and the gray pixels that indicate why x should not be a '3' according to K. In the next section, we explain how to use these concepts to explain predictions made by an SVM classification process.

[1] In this example, one can also say that $A@K$ represents what has been learned about a class of handwritten 3s after following a learning process that yields K as a result.

3. EXPLAINABLE SVM CLASSIFICATION

As was mentioned earlier, our aim is the contextualization of SVM predictions to make them better interpretable. For that purpose, in this section we describe our novel XSVMC process, by which SVM predictions are augmented with AADs. As depicted in Figure 2, the main components of XSVMC are a learning process, an evaluation process and a prediction step. Among these components, the fundamental contribution of this work is the novel evaluation process that makes use of the MISV to contextualize the evaluations. In what follows we give details of each component.

3.1. Learning Process

The aim of the learning process in XSVMC is to obtain a knowledge model for each class in a collection of well-known classes. To describe how it works, we make use of a process that mimics a learning behavior where a person learns about a concept (or class) by studying objects that satisfy or dissatisfy an evaluation criterion related to the concept. The process is based on the *feature-influence representational model* [15], which is summarized below.

3.1.1. Feature-influence representational model

Let $\mathcal{M}$ be a m-dimensional feature space in which each dimension corresponds to a feature f_j in a collection $\mathcal{F} = \{f_1, \cdots, f_m\}$. Let x be an object with a collection of features $\mathcal{F}_x \subseteq \mathcal{F}$. And let p_A be a proposition having the canonical form 'x BELONGS TO A' (see Section 2). Under these considerations, the influence of the features of x on the appraisal of p_A is modeled as follows:

- The *overall influence* $\mathbf{x}$ of the features of x on the classification is given by the vector

$$\mathbf{x} = \sum_{j=1}^{m} \beta_j \hat{\mathbf{f}}_j, \tag{1}$$

where β_j denotes the *overall importance* (or weight) on the classification of f_j among the features in $\mathcal{F}$, and $\hat{\mathbf{f}}_j$ is the unit vector representing the dimension related to f_j in $\mathcal{M}$. For instance, Figure 3 depicts the overall influence of x in a 2-dimensional feature space, where $\mathcal{F} = \{f_1, f_2\}$. In this case, if f_1 and f_2 represent, e.g., two pixels in a digitized image, β_1 and β_2 might represent their respective levels of illumination.

- A particular knowledge model about A, say K_A, is represented by a line in $\mathcal{M}$ and described by a pair $\langle \hat{\mathbf{u}}_A, t_A \rangle$ such that: (i) $\hat{\mathbf{u}}_A$ represents a unit vector that points to a location in $\mathcal{M}$ where the fulfillment of p_A is favored; and (ii) t_A is a point on the line defined by $\hat{\mathbf{u}}_A$ where the fulfillment of p_A is neither favored nor disfavored. For instance, Figure 4 shows a particular knowledge model K_A in the aforementioned 2-dimensional feature space. In this case, while the zone with the label '+' represents a location where the fulfillment of p_A is favored, the zone with the label '-' represents a location where the fulfillment of p_A is disfavored.

- The *specific influence* of the features of x on the appraisal of p_A is given by the vector

$$\mathbf{x}_A = (\mathbf{x} \cdot \hat{\mathbf{u}}_A)\hat{\mathbf{u}}_A = \sum_{j=1}^{m} \beta_{j_A} \hat{\mathbf{u}}_A, \tag{2}$$

where $\beta_{j_A} \hat{\mathbf{u}}_A$ denotes the *specific influence* of f_j on the appraisal of p_A, and '$\cdot$' denotes the inner product. Notice that $\mathbf{x}_A$ is the *vector projection* of the overall influence vector on the line that represents K_A, i.e., $\mathbf{x}_A$ corresponds to the vector projection of $\mathbf{x}$ on $\hat{\mathbf{u}}_A$. For instance, Figure 5 depicts the specific influence $\beta_1 \hat{\mathbf{f}}_1$ of f_1 on the appraisal of p_A according to the particular knowledge model about A characterized in Figure 4. In this case, if f_1 and β_1 represent respectively the aforementioned pixel and level of illumination, then β_{1_A} represents the specific influence of that pixel on the appraisal of p_A.

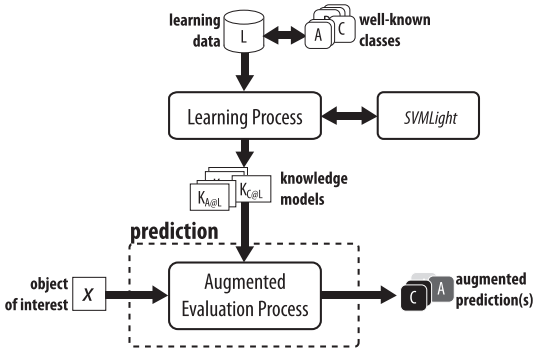

Figure 2 | A contextual view of XSVMC [14].

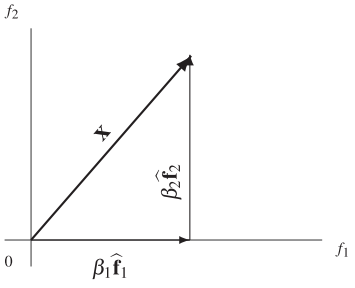

Figure 3 | Overall influence of the features of an object x.

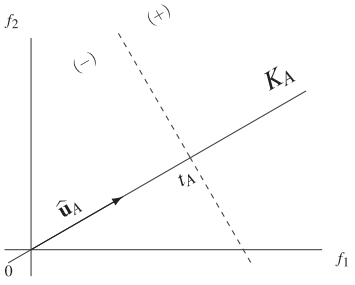

Figure 4 | Characterization of a particular knowledge about A.

- The *level to which* x satisfies (or dissatisfies) p_A is determined by the magnitude of the vector $\mathbf{l}_A$ defined by

$$\mathbf{l}_A = \mathbf{x}_A - t_A\hat{\mathbf{u}}_A, \tag{3}$$

i.e, this level is given by

$$||\mathbf{l}_A|| = \sqrt{\mathbf{l}_A \cdot \mathbf{l}_A}. \tag{4}$$

If the directions of $\mathbf{l}_A$ and $\hat{\mathbf{u}}_A$ are the same, x satisfies p_A to the extent $||\mathbf{l}_A||$. By the contrary, if the direction of $\mathbf{l}_A$ is opposite to the direction of $\hat{\mathbf{u}}_A$, x dissatisfies p_A to the extent $||\mathbf{l}_A||$. For example, Figure 6 shows the vector $\mathbf{l}_A$ that represents the resulting specific influence of x on the appraisal of p_A according to the particular knowledge model about A characterized in Figure 4. Since in this case the directions of $\mathbf{l}_A$ and $\hat{\mathbf{u}}_A$ are the same, x satisfies p_A to the extent $||\mathbf{l}_A||$.

3.1.2. Obtaining knowledge models

At this point, the feature-influence model can be used for explaining how to extract a model of the knowledge about a class A, say $K_A = \langle \hat{\mathbf{u}}_A, t_A \rangle$,[2] by looking into the features of each object x_i in a training collection, say $X_0 = \{x_1, \cdots, x_n\}$ (see Figure 7). Such a training collection consists of objects that satisfy the proposition p_A (*positive examples*), as well as objects that dissatisfy that proposition (*negative examples*). The main steps of the algorithm proposed in a previous work [15] to extract K_A are the following – the interested reader is referred to that work for a detailed description of this algorithm:

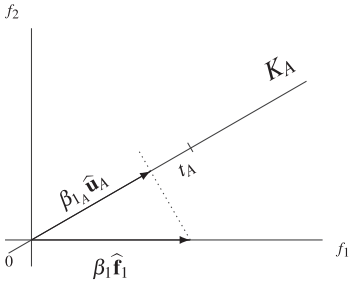

Figure 5 | Specific influence of one of the features of x.

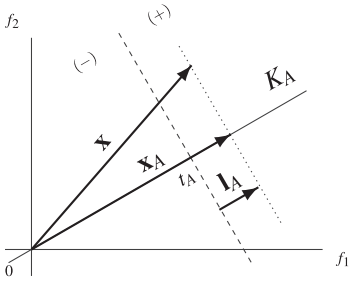

Figure 6 | Resulting specific influence $\mathbf{x}_A$ of the features of x.

[2]To be consistent with the notation introduced in Figure 2 where the *"source"* of the knowledge about A is explicitly denoted, we should say $K_{A@X_0} = \langle \hat{\mathbf{u}}_{A@X_0}, t_{A@X_0} \rangle$. For the sake of readability we use hereafter this simplified form of the notation.

1. For each $x_i \in X_0$, identify its features and put them into $\mathcal{F}_{X_0}$.
2. Assign an overall importance $\beta_{i,j}$ to each feature $f_j \in \mathcal{F}_{X_0}$ based on its overall influence on the appraisal of p_A for each $x_i \in X_0$.
3. Compute $\langle \hat{\mathbf{u}}_A, t_A \rangle$ in such a way that (i) the correspondence between each $x_i \in X_0$ satisfying or dissatisfying p_A and the resulting specific influence of its features is preserved, and (ii) both the aggregate of the specific influences that favor the fulfillment of p_A and the aggregate of the specific influences that disfavor such fulfillment are maximized.

In the first step, the objects' features that will be considered during the learning process are identified. It is worth mentioning that a feature can represent something about one or more characteristics of an object. For example, a feature can represent the presence of either one pixel or a group of pixels in a digitalized handwritten number.

In the second step, an overall weight for each of the features identified in the first step is assigned based on its relative influence on the classification. For example, the level of illumination can be considered as the overall weight of a feature consisting of one pixel.

In the third step, the components of K_A, i.e., $\hat{\mathbf{u}}_A$ and t_A, are adjusted in such a way that the following two conditions are (mostly) satisfied: (i) the resulting specific influence of the features of each object in the training collection is in agreement with the label assigned to the object (i.e., positive or negative example); and (ii) both the aggregate of the specific influence of positive examples and the aggregate of the specific influence of negative examples are maximized. For instance, Figures 8 and 9 illustrate, in that order, how the adjustments of t_A and $\hat{\mathbf{u}}_A$ can modify the resulting specific influence $\mathbf{x}_A$ of x shown in Figure 6.

The problem of finding an optimal couple $\langle \hat{\mathbf{u}}_A, t_A \rangle$ in the third step can be related to the problem of *finding an optimal separating hyperplane* with an SVM, which is stated as follows [5,6]:

- Suppose that a hyperplane H separates positive examples from negative ones. Let H^+ be a hyperplane that is parallel to H and contains the nearest positive example(s) and let H^- be another hyperplane that is also parallel to H and contains the nearest negative example(s). Find H such that the distance between H^+ and H^- is the largest.

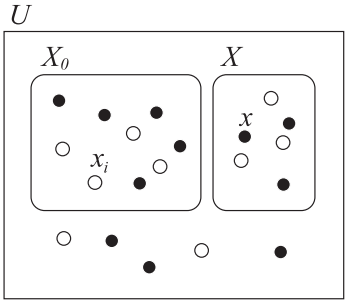

Figure 7 | Training and test collections consisting of positive examples, denoted by black circles, and negative examples, denoted by white circles.

The hyperplane H is defined by $\mathbf{w}\cdot\mathbf{x}_i+b=0$, where $\mathbf{w}$ and b represent in that order the normal vector to H and the intersect term, and $\mathbf{x}_i$ denotes any vector related to an object $x_i \in X_0$. An illustration of a hyperplane H is shown in Figure 10 along with the hyperplane H^+, defined by $\mathbf{w}\cdot\mathbf{x}_i+b=1$, and the hyperplane H^-, defined by $\mathbf{w}\cdot\mathbf{x}_i+b=-1$. Notice that, while the normal vector $\mathbf{w}$ and the directional vector (DV) $\hat{\mathbf{u}}_A$ are parallel to each other and point

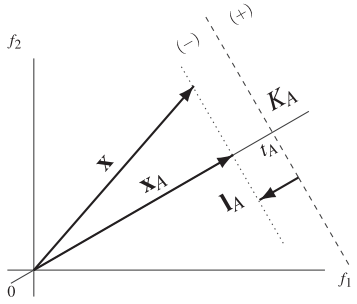

Figure 8 | Adjusting the component t_A of K_A.

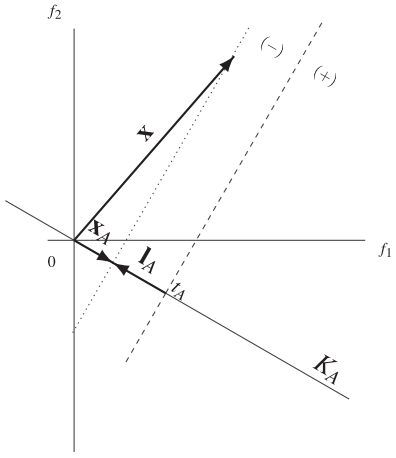

Figure 9 | Adjusting the component $\hat{\boldsymbol{u}}_A$ of K_A.

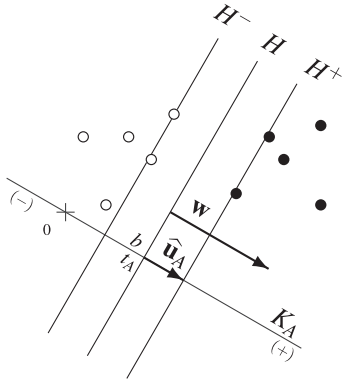

Figure 10 | An optimal couple <$\hat{\mathbf{u}}_A$, t_A> in relation to an optimal separating hyperplane H.

to the same side, the intersect term b corresponds to t_A. Hence, the equations

$$\hat{\mathbf{u}}_A = \frac{\mathbf{w}}{||\mathbf{w}||} \tag{5}$$

and

$$t_A = -\frac{b}{||\mathbf{w}||} \tag{6}$$

hold and (2), (3) and (4) can be rewritten as

$$\mathbf{x}_A = \frac{(\mathbf{x}\cdot\mathbf{w})}{||\mathbf{w}||}\hat{\mathbf{u}}_A, \tag{7}$$

$$\mathbf{l}_A = \frac{(\mathbf{x}\cdot\mathbf{w}+b)}{||\mathbf{w}||}\hat{\mathbf{u}}_A, \tag{8}$$

and

$$||\mathbf{l}_A|| = \left|\frac{(\mathbf{x}\cdot\mathbf{w}+b)}{||\mathbf{w}||}\right| \tag{9}$$

respectively.

To find the values of $\mathbf{w}$ and b, the Euclidean distance between H^+ and H^-, i.e., $d(H^+,H^-)=2/||\mathbf{w}||$, should be maximized subject to the following constraints: if x_i is a positive example, then $\mathbf{w}\cdot\mathbf{x}_i+b\geqslant 1$; and if x_i is a negative example, then $\mathbf{w}\cdot\mathbf{x}_i+b\leqslant -1$. However, minimizing $\frac{1}{2}||\mathbf{w}||^2$ is preferred. Thus, $\mathbf{w}$ and b are computed by the Lagrangian formulation of the *linearly separable case of an SVM classifier* [16], in which the value of Λ, given by equation

$$\Lambda = \frac{1}{2}\parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{n}\lambda_i\left(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1\right), \tag{10}$$

is minimized subject to $y_i(\mathbf{w}\cdot\mathbf{x_i}+b)-1\geqslant 0$ and $(\forall\lambda_i\in\{\lambda_1,\cdots,\lambda_n\})(\lambda_i>0)$. In this equation, $y_i\in\{-1,1\}$ is a label that indicates whether x_i is a positive example ($y_i=1$) or a negative one ($y_i=-1$); and $\lambda_1,\cdots,\lambda_n$ are Lagrange multipliers.

The previous problem is reformulated to an equivalent *dual problem* [16], which consists in finding the Lagrange multipliers such that the gradient of Λ with respect to $\mathbf{w}$ and b yields zero, and Λ is maximized. The conditions for the gradient of Λ, i.e., $\delta\Lambda/\delta\mathbf{w}=0$ and $\delta\Lambda/\delta b=0$, result in

$$\mathbf{w} = \sum_{i=1}^{n}\lambda_i y_i \mathbf{x}_i \tag{11}$$

and

$$\sum_{i=1}^{n}\lambda_i y_i = 0, \tag{12}$$

which are introduced in (10) to obtain

$$\Lambda = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1,k=1}^{n}\lambda_i\lambda_k y_i y_k(\mathbf{x_i}\cdot\mathbf{x}_k). \tag{13}$$

In this equation, Λ is formulated as a function of the Lagrange multipliers only, and is maximized subject to the constraints (12) and

$\lambda_i \geqslant 0, i = 1, \cdots, n$. In the *linearly non-separable case*, the last constraint is generalized to $0 \leqslant \lambda_i \leqslant C, i = 1, \cdots, n$, where C is called the *regularization parameter* [16]. The solution is given by (11) and

$$b = y_i - (\mathbf{w} \cdot \mathbf{x}_i) \tag{14}$$

for any vector $\mathbf{x}_i$ associated to $0 < \lambda_i < C, i = 1, \cdots, n$.[3] The objects related to these vectors are deemed to be crucial elements in X_0 since any of them can change the direction of H if removed. Because of this, these vectors are named *support vectors*.

In situations where the vectors are not linearly separable, those vectors can be mapped to another space in which they can be separated by a linear hyperplane. This means that a vector $\mathbf{x}_i$ in the feature space $\mathcal{M}$ can be mapped to a higher dimensional space, say $\mathcal{N}$, through a mapping $\phi : \mathcal{M} \mapsto \mathcal{N}$, such that (13) can be written as

$$\Lambda = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1,k=1}^{n} \lambda_i \lambda_k y_i y_k (\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x}_k)). \tag{15}$$

Instead of computing the inner product between $\phi(\mathbf{x_i})$ and $\phi(\mathbf{x}_k)$ in a higher dimensional space, the use of a *kernel function* $K(\mathbf{x_i}, \mathbf{x_k}) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x}_k)$ is preferred [5,6] – notice that K computes the inner product (or a reflection of similarity) between $\mathbf{x_i}$ and $\mathbf{x_k}$ in $\mathcal{N}$. Hence, (15) becomes

$$\Lambda = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1,k=1}^{n} \lambda_i \lambda_k y_i y_k (K(\mathbf{x_i}, \mathbf{x_k})). \tag{16}$$

Among others, the *Polynomial kernel* of degree d, defined by

$$K(\mathbf{x_i}, \mathbf{x_k}) = (\mathbf{x_i} \cdot \mathbf{x_k} + 1)^d, \tag{17}$$

and the *radial basis function (RBF) kernel* with parameter $\gamma > 0$, defined by

$$K(\mathbf{x_i}, \mathbf{x_k}) = \exp(-\gamma ||\mathbf{x_i} \cdot \mathbf{x_k}||^2), \tag{18}$$

are examples of such kernel functions.

As noticed, SVMs can be used for the computation of the optimal couple $K_A = \langle \hat{\mathbf{u}}_A, t_A \rangle$ even if the objects in the training collection are not linearly separable.

3.2. Augmented Evaluation Process

A conventional classification algorithm can use the knowledge model resulting from the above-described learning process to evaluate the level to which an object is member of a given class. For example, after obtaining a feature-influence model of the knowledge about class A, i.e., $K_A = \langle \hat{\mathbf{u}}_A, t_A \rangle$, the conventional classification algorithm can use K_A to evaluate, by means of (3) and (4), the level to which an object x is a member of A. In a similar way, the algorithm can use a model about class B, say $K_B = \langle \hat{\mathbf{u}}_B, t_B \rangle$, to evaluate the level to which x is a member of B. After that, the resulting levels can be used for making a prediction about the class of x: if the level to which x is member of A, i.e., $||\mathbf{l}_A||$, is greater than the level to which x is member of B, i.e., $||\mathbf{l}_B||$, A can be returned as the predicted class of x.

[3] To find the values of $\mathbf{w}$, b and all $\lambda_i \in \{\lambda_1, \cdots, \lambda_n\}$, the software package *SVMLight* [17] can be used.

If a user would like to know in the previous example why the predicted class is A, the conventional classification algorithm is limited to offer an answer like "x is more A than B because $||\mathbf{l}_A|| > ||\mathbf{l}_B||$". As noticed, nothing is mentioned in this answer about the *relevant features* of x that support that prediction. In this regard, the purpose of the evaluation process in XSVMC is to put those evaluations in context and, thus, help explaining the forthcoming predictions. Two procedures that put those evaluations in context are explained below.

Consider a class A, a collection $\mathcal{F}$ consisting of the features in a m-dimensional feature space, an object x in a test collection X (see Figure 7) and a proposition p_A : 'x BELONGS TO A'. Consider also a collection $\mathcal{F}_x \subseteq \mathcal{F}$ consisting of the features of x, as well as a collection $\mathcal{F}_{X_0} \subseteq \mathcal{F}$ consisting of the features identified after following the previous learning process with a training collection X_0 (see Figure 11). Assume that $K_A = \langle \hat{\mathbf{u}}_A, t_A \rangle$ is a feature-influence representation of a particular knowledge about A. Assume also that $\hat{\mathbf{u}}_A = \omega_1 \hat{\mathbf{f}}_1 + \cdots + \omega_m \hat{\mathbf{f}}_m$ and $\mathbf{x} = \beta_1 \hat{\mathbf{f}}_1 + \cdots + \beta_m \hat{\mathbf{f}}_m$ respectively represent the DV and the overall influence vector. Under these considerations, a procedure for performing an augmented evaluation of p_A that yields an augmented IFS element $\langle \hat{\mu}_A(x), \hat{\nu}_A(x) \rangle = \langle \langle \mu_A(x), F_{\mu_A}(x) \rangle, \langle \nu_A(x), F_{\nu_A}(x) \rangle \rangle$ as a result, consists of the following steps [14]:

1. For each feature f_j in $\mathcal{F}_x \cap \mathcal{F}_{X_0}$, compute its *specific influence* on the appraisal of p_A, i.e., compute $\mathbf{f}_{j_A} = \beta_{jA} \hat{\mathbf{u}}_A = \beta_j \omega_j \hat{\mathbf{u}}_A$ – recall from (2) that $\mathbf{x}_A = \beta_{1_A} \hat{\mathbf{f}}_{1_A} + \cdots + \beta_{m_A} \hat{\mathbf{f}}_{m_A}$. If $\beta_{jA} > 0$ include f_j in $F_{\mu_A}(x)$; else include f_j into $F_{\nu_A}(x)$ if $\beta_{jA} < 0$; otherwise, exclude f_j from both $F_{\mu_A}(x)$ and $F_{\nu_A}(x)$. For instance, if the specific influence of f_9 in Figure 11 is positive (i.e., $\beta_{9A} > 0$), f_9 will be included in $F_{\mu_A}(x)$; likewise, if the specific influence of f_7 is negative (i.e., $\beta_{7A} < 0$), f_7 will be included in $F_{\nu_A}(x)$; and if the specific influence f_5 is zero (i.e., $\beta_{5A} = 0$), f_5 will be excluded from both $F_{\mu_A}(x)$ and $F_{\nu_A}(x)$.
2. For each feature f_j in $\mathcal{F}_{X_0} - \mathcal{F}_x$, take into consideration the following rationale to decide whether or not f_j should be included into either $F_{\mu_A}(x)$ or $F_{\nu_A}(x)$: (i) if $\omega_j > 0$, it can be considered that the nonexistence of f_j in x will be against the membership of x in A and, thus, f_j should be included in $F_{\nu_A}(x)$; (ii) else, if $\omega_j < 0$, it can be considered that the nonexistence of f_j in x will favor the membership of x in A and, thus, f_j should be included in $F_{\mu_A}(x)$; (iii) otherwise, it can be considered that f_j does not favor nor disfavor the membership of x in A and, thus, f_j should be excluded from both $F_{\mu_A}(x)$ and $F_{\nu_A}(x)$. For instance, if $\omega_6 > 0$ and it is considered that the nonexistence

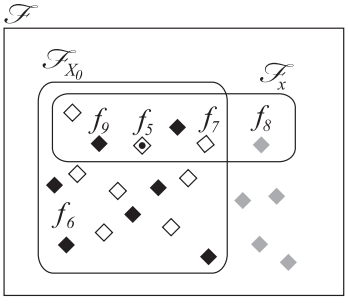

Figure 11 | Features collections.

of f_6 in x will be against the membership of x in A, f_6 should be included in $F_{\nu_A}(x)$. It is worth mentioning that, even though the features considered in this step are not part of x, their inclusion in $F_{\mu_A}(x)$ or $F_{\nu_A}(x)$ can help the users to be aware of what has been focused on during the evaluation of p_A.

3. For each feature f_j in $\mathcal{F}_x - \mathcal{F}_{X_0}$, if it is considered that the existence of f_j in x will be against the membership of x in A, f_j should be included in $F_{\nu_A}(x)$. Otherwise, f_j should be excluded from both $F_{\mu_A}(x)$ and $F_{\nu_A}(x)$. For instance, if it is considered that the existence of f_8 in $\mathcal{F}_x$ (see Figure 11) will disfavor the membership of x in A, f_8 should be included in $F_{\nu_A}(x)$. In a similar way to the previous step, the inclusion of the features considered in this step can help the users to get insights into what features of x have not been focused on during the evaluation of p_A due to the absence of those features from the model.

4. Compute $\mu_A(x)$ and $\nu_A(x)$ by means of the equations

$$\mu_A(x) = \breve{\mu}_A(x)/\eta_A(x) \tag{19}$$

and

$$\nu_A(x) = \breve{\nu}_A(x)/\eta_A(x) \tag{20}$$

respectively, where

$$\breve{\mu}_A(x) = \begin{cases} \dfrac{|t_A| + \sum_{j=1}^{m} \beta_{jA}}{\| \mathbf{x} \|} & \text{iff } (\forall \beta_{jA} > 0) \wedge (t_A < 0); \\ \dfrac{\sum_{j=1}^{m} \beta_{jA}}{\| \mathbf{x} \|} & \text{iff } (\forall \beta_{jA} > 0) \wedge (t_A \geqslant 0); \\ 0 & \text{otherwise;} \end{cases} \tag{21}$$

$$\breve{\nu}_A(x) = \begin{cases} \dfrac{t_A + \sum_{j=1}^{m} |\beta_{jA}|}{\| \mathbf{x} \|} & \text{iff } (\forall \beta_{jA} < 0) \wedge (t_A > 0) \\ \dfrac{\sum_{j=1}^{m} |\beta_{jA}|}{\| \mathbf{x} \|} & \text{iff } (\forall \beta_{jA} < 0) \wedge (t_A \leqslant 0); \\ 0 & \text{otherwise;} \end{cases} \tag{22}$$

and

$$\eta_A(x) = \max\left(1, \breve{\mu}_A(x) + \breve{\nu}_A(x)\right). \tag{23}$$

It is worth mentioning that (21) and (22) are obtained as follows. Using (2), (3) can be rewritten as

$$\mathbf{l}_A = \left(\sum_{j=1}^{m} \beta_{j_A} - t_A\right)\hat{\mathbf{u}}_A. \tag{24}$$

and, thus, (4) can be rewritten as

$$||\mathbf{l}_A|| = \left| \sum_{j=1}^{m} \beta_{j_A} - t_A \right|. \tag{25}$$

The first term of (25) can be split into the sum of positive specific influences and the sum of negative specific influences. Thus, (25) can be rewritten as

$$||\mathbf{l}_A|| = \left| \left(\sum_{j=1, \beta_{j_A} > 0}^{m} \beta_{j_A} + \sum_{j=1, \beta_{j_A} < 0}^{m} \beta_{j_A} \right) - t_A \right|. \tag{26}$$

Notice in (26) that, while the sum of positive specific influences will be increased by $|t_A|$ if $t_A < 0$, this sum will be decreased by $|t_A|$ if $t_A > 0$. Thus, if $t_A < 0$, the first term of (26) along with $|t_A|$ will be taken into account for the computation of $\breve{\mu}_A(x)$ in (21). Likewise, if $t_A > 0$, the second term of (26) along with $|t_A|$ will be taken into account for the computation of $\breve{\nu}_A(x)$ in (22). Since $\mu_A(x)$ and $\nu_A(x)$ are considered to be numbers in the unit interval $[0, 1]$, the sums of the specific influences are first divided by $\| \mathbf{x} \|$ in (21) and (22); then, $\breve{\mu}_A(x)$ and $\breve{\nu}_A(x)$ are divided by the result of (23) in (19) and (20) respectively.

The idea behind (21) and (22) is to quantify the levels to which each of the features of x favors or disfavors the membership of x in A. Notice in (21) that the membership level $\breve{\mu}_A(x)$ increases when a feature f_j has a positive specific influence β_{jA}. For instance, consider the specific influence of the feature f_1 depicted by $\mathbf{f}_{1_A} = \beta_{1_A}\hat{\mathbf{u}}_A$ in Figure 12. Since $\mathbf{f}_{1_A}$ and $\hat{\mathbf{u}}_A$ point to the same location, f_1 favors the fulfillment of p_A, i.e., f_1 has a positive specific influence on the appraisal of p_A. Likewise, notice in (22) that the nonmembership level $\breve{\nu}_A(x)$ increases when f_j has a negative specific influence. This case is illustrated in Figure 12 by the specific influence $\mathbf{f}_{2_A} = \beta_{2_A}\hat{\mathbf{u}}_A$ of the feature f_2. Since $\mathbf{f}_{2_A}$ points to the opposite direction of $\hat{\mathbf{u}}_A$, f_2 is against the fulfillment of p_A, i.e., f_1 has a negative specific influence on the appraisal of p_A. In this regard, the resulting specific influence vector $\mathbf{l}_A$ is given by $\mathbf{l}_A = \mathbf{f}_{1_A} + \mathbf{f}_{2_A} - t_A\hat{\mathbf{u}}_A = (\beta_{1_A} + \beta_{2_A} - t_A)\hat{\mathbf{u}}_A$, where $\beta_{1_A} > 0$, $\beta_{2_A} < 0$ and $t_A > 0$. Thus, the membership level $\breve{\mu}_A(x)$ and nonmembership level $\breve{\nu}_A(x)$ will be $\breve{\mu}_A(x) = \beta_{1_A}/\| \mathbf{x} \|$ and $\breve{\nu}_A(x) = (|\beta_{2_A}| + t_A)/\| \mathbf{x} \|$ respectively in this case.

In contrast to a conventional classification algorithm, XSVMC can use the above procedure to perform a contextualized evaluation of the membership (and nonmembership) of an object in a given class. For example, to evaluate the membership of an object x in a class A, XSVMC makes use of the evaluation procedure with a model of the knowledge about A, say $K_A = \langle \hat{\mathbf{u}}_A, t_A \rangle$, to obtain $\langle \hat{\mu}_A(x), \hat{\nu}_A(x) \rangle$ as a result. Likewise, XSVMC uses the procedure with

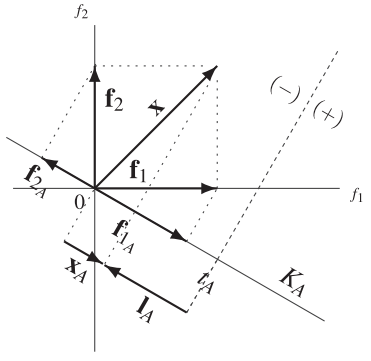

Figure 12 Specific influence of two features, f_1 and f_2, on the appraisal of a proposition p_A: 'x BELONGS TO A'.

the knowledge model about another class, say $K_B = \langle \hat{\mathbf{u}}_B, t_B \rangle$, to evaluate the membership of x in B and, so, obtain $\langle \hat{\mu}_B(x), \hat{\nu}_B(x) \rangle$ as a result. Then, XSVMC can compare those evaluations to predict whether the class of x is A or B: if the buoyancy of $\langle \hat{\mu}_A(x), \hat{\nu}_A(x) \rangle$, i.e., $\rho_A(x) = \mu_A(x) - \nu_A(x)$, (see Section 2) is greater than the buoyancy of $\langle \hat{\mu}_B(x), \hat{\nu}_B(x) \rangle$, i.e., $\rho_B(x) = \mu_B(x) - \nu_B(x)$, the predicted class will be A. In this case, if a user would like to know why the predicted class of x is A, an XAI system that incorporates XSVMC (see Section 1) can use the previous prediction to offer an answer such as "the features in $F_{\mu_A}(x)$ suggest that x belongs to A with a grade of $\mu_A(x)$; yet, the features in $F_{\nu_A}(x)$ indicate that x does not belong to A with a grade of $\nu_A(x)$."

In some situations, the collections $F_{\mu_A}(x)$ and $F_{\nu_A}(x)$ might include features having complex arrangements of attributes – e.g., when a polynomial kernel has been used for obtaining the knowledge model $K_A = \langle \hat{\mathbf{u}}_A, t_A \rangle$. To reduce that complexity, XSVMC includes the following alternative evaluation procedure in which the MISV is used for obtaining features having simplified arrangements of attributes.

Consider the next variant of (9)

$$||\mathbf{l}_A|| = \left| \frac{\sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + b}{|| \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i ||} \right|, \forall \lambda_i > 0, \tag{27}$$

in which $\mathbf{w}$ and $\mathbf{x}_i \cdot \mathbf{x}$ have been replaced by (11) and $K(\mathbf{x}_i, \mathbf{x})$ respectively. Recall that $\mathbf{x}_i$ denotes any of the support vectors since $\lambda_i > 0$, and y_i represents the value, -1 or 1, associated to it. Notice that the influence of $\mathbf{x}_i$ on the evaluation of $\mathbf{x}$ is given by $\lambda_i y_i K(\mathbf{x}_i, \mathbf{x})$. In this regard, the support vector having the greatest positive influence on the evaluation of $\mathbf{x}$ can be obtained by

$$\mathbf{v} = \arg\max_{\mathbf{x}_i} \{\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) | \forall \mathbf{x}_i \in S\}, \tag{28}$$

where S represents the collection of support vectors. From a semantic point of view, $\mathbf{v}$ represents the most similar support vector to $\mathbf{x}$. Hence, $\mathbf{v}$ can be used for the identification of the features that have been relevant to the evaluation. With this consideration and representing $\mathbf{v}$ and $\mathbf{x}$ by means of $\mathbf{v} = \sum_{j=1}^{m} \alpha_j \hat{\mathbf{f}}_j$ and $\mathbf{x} = \sum_{j=1}^{m} \beta_j \hat{\mathbf{f}}_j$ respectively, we compute the specific influence $\alpha_j \beta_j$ for each f_j in the collection of x's features, i.e., $f_j \in \mathcal{F}_x$. In a similar way to the first step of the previous evaluation procedure, we include f_j in $F_{\mu_A}(x)$ if $\alpha_j \beta_j > 0$; else, we include f_j into $F_{\nu_A}(x)$ if $\alpha_j \beta_j < 0$; otherwise we exclude f_j from $F_{\mu_A}(x)$ and $F_{\nu_A}(x)$. It might also be assumed that f_j is against the membership if $\alpha_j = 0$ or $\beta_j = 0$.

To obtain $\mu_A(x)$ and $\nu_A(x)$, we compute $\breve{\mu}_A(x)$ and $\breve{\nu}_A(x)$ by means of

$$\breve{\mu}_A(x) = \begin{cases} \dfrac{\sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + b}{\| \mathbf{x} \| \, \| \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \|} & \text{iff } (\forall \lambda_i > 0) \wedge (b > 0) \\ & \wedge \left(\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) > 0 \right); \\ \dfrac{\sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x})}{\| \mathbf{x} \| \, \| \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \|} & \text{iff } (\forall \lambda_i > 0) \wedge (b \leqslant 0) \\ & \wedge \left(\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) > 0 \right); \\ 0 & \text{otherwise;} \end{cases} \tag{29}$$

and

$$\breve{\nu}_A(x) = \begin{cases} \dfrac{\sum_{i=1}^{n} |\lambda_i y_i K(\mathbf{x}_i, \mathbf{x})| + |b|}{\| \mathbf{x} \| \, \| \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \|} & \text{iff } (\forall \lambda_i > 0) \wedge (b < 0) \\ & \wedge \left(\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) < 0 \right); \\ \dfrac{\sum_{i=1}^{n} |\lambda_i y_i K(\mathbf{x}_i, \mathbf{x})|}{\| \mathbf{x} \| \, \| \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i \|} & \text{iff } (\forall \lambda_i > 0) \wedge (b \geqslant 0) \\ & \wedge \left(\lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) < 0 \right); \\ 0 & \text{otherwise;} \end{cases} \tag{30}$$

and replace them in (19) and (20) respectively.

To obtain (29) and (30), we split (27) into the sum of positive specific influences and the sum of negative specific influences. Thus, we rewrite (27) as

$$||\mathbf{l}_A|| = \left| \frac{s^+ + s^- + b}{|| \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i ||} \right|, \forall \lambda_i > 0 \tag{31}$$

where

$$s^+ = \sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}), \forall \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) > 0 \tag{32}$$

and

$$s^- = \sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}), \forall \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) < 0. \tag{33}$$

Notice in (31) that, while the sum of positive specific influences increases if $b > 0$, this sum decreases if $b < 0$. Hence, (32) along with b are taken into account for the computation of $\breve{\mu}_A(x)$ in (29) if $b > 0$. In a similar way, the absolute value of (33) along with $|b|$ are taken into account for the computation of $\breve{\nu}_A(x)$ in (30) if $b < 0$. As was done with (21) and (22), to obtain $\mu_A(x)$ and $\nu_A(x)$ the sums of the specific influences are divided by $\| \mathbf{x} \|$ in (29) and (30) and, then, $\breve{\mu}_A(x)$ and $\breve{\nu}_A(x)$ are divided by the result of (23) in (19) and (20) respectively.

Notice in (29) that the membership level $\breve{\mu}_A(x)$ increases when a support vector $\mathbf{x}_i$ has a positive influence, which is computed by $\lambda_i y_i K(\mathbf{x}_i, \mathbf{x})$, or when the intersect term b is positive. Likewise, notice in (30) that the nonmembership level $\breve{\nu}_A(x)$ increases when a support vector has a negative influence or when the intersect term is negative. For instance, in Figure 13 while the support vector $\mathbf{x}_1$ has a positive specific influence $\mathbf{x}_{1A} = \frac{\lambda_1 y_1 K(\mathbf{x}_1, \mathbf{x})}{\| \lambda_1 y_1 \mathbf{x}_1 + \lambda_2 y_2 \mathbf{x}_2 \|} \hat{\mathbf{u}}_A$ on the appraisal of p_A, the support vector $\mathbf{x}_2$ has a negative specific influence $\mathbf{x}_{2A} = \frac{\lambda_2 y_2 K(\mathbf{x}_2, \mathbf{x})}{\| \lambda_1 y_1 \mathbf{x}_1 + \lambda_2 y_2 \mathbf{x}_2 \|} \hat{\mathbf{u}}_A$ on the appraisal. In this case, the resulting specific influence vector $\mathbf{l}_A$ is given by $\mathbf{l}_A = \mathbf{x}_{1A} + \mathbf{x}_{2A} + \mathbf{b}_A = \frac{\lambda_1 y_1 K(\mathbf{x}_1, \mathbf{x}) + \lambda_2 y_2 K(\mathbf{x}_2, \mathbf{x}) + b}{\| \lambda_1 y_1 \mathbf{x}_1 + \lambda_2 y_2 \mathbf{x}_2 \|} \hat{\mathbf{u}}_A$, where $\lambda_1 y_1 K(\mathbf{x}_1, \mathbf{x}) > 0$, $\lambda_2 y_2 K(\mathbf{x}_2, \mathbf{x}) < 0$ and $b > 0$. Hence, the membership level $\breve{\mu}_A(x)$ and nonmembership level $\breve{\nu}_A(x)$ will be $\breve{\mu}_A(x) = \frac{\lambda_1 y_1 K(\mathbf{x}_1, \mathbf{x}) + b}{\| \mathbf{x} \| \, \| \lambda_1 y_1 \mathbf{x}_1 + \lambda_2 y_2 \mathbf{x}_2 \|}$ and $\breve{\nu}_A(x) = \frac{|\lambda_2 y_2 K(\mathbf{x}_2, \mathbf{x})|}{\| \mathbf{x} \| \, \| \lambda_1 y_1 \mathbf{x}_1 + \lambda_2 y_2 \mathbf{x}_2 \|}$ respectively.

It is worth mentioning that the main difference between both aforementioned evaluation procedures lies in the strategy to identify which features have been relevant to the evaluation: while the first

evaluation procedure makes use of the DV $\hat{\mathbf{u}}_A$, which incorporates all the support vectors, the second procedure uses only the support vector with the greatest positive influence on the evaluation.

4. USE CASE

Aiming to illustrate how the novel XSVMC works, in this section we implement a use case where the classes of handwritten digits are predicted. This use case consists of a *successful scenario*, in which the right class is predicted, and an *unsuccessful scenario*, in which a wrong class is predicted.

To implement the use case, we use two collections of digitized handwritten numbers: the first one is a very small collection consisting of the handwritten numbers depicted in Figures 14–16, and the second one is the *MNIST collection* [18], which contains 70000 handwritten numbers.

As shown in Figure 17, such a digitized handwritten number consists of 784 pixels, each associated with a value between 0 and 1, where 0 and 1 denote, in that order, no strength and the maximum strength of a pen while handwriting on that pixel.

To use the learning and evaluation procedures included in XSVMC, each digitized handwritten number has been modeled in a 784-dimensional feature space $\mathcal{M}$ as a feature-influence vector $\mathbf{x} = \beta_1\hat{\mathbf{f}}_1 + \cdots + \beta_{784}\hat{\mathbf{f}}_{784}$, such that β_j denotes the strength of the pen in pixel f_j. For instance, while in Figure 17 the value of β_{58} is 0 since no strength has been put on pixel f_{58}, the value of β_{275} is 0.99 since the strength of the pen in this pixel is almost the maximum.

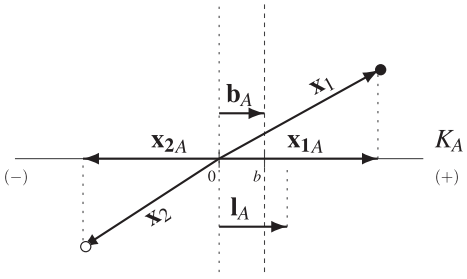

Figure 13 | Specific influence of support vectors x_1 and x_2 on the appraisal of a proposition p_A: 'x BELONGS TO A'.

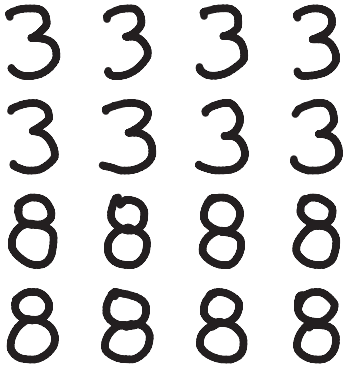

Figure 14 | User 1's training collection ($X_{0@usr1}$).

4.1. XSVMC on a Very Small Collection

An SVM classification process can be effective in cases where the dimension of the feature space $\mathcal{M}$ is greater that the number of samples [5,6]. To illustrate how XSVMC works in such cases, we use the collections $X_{0@usr1}$ (see Figure 14) and $X_{0@usr2}$ (see Figure 15), which include handwritten numbers given by two users, say *usr1* and *usr2*.

We first use $X_{0@usr1}$ as a training collection to obtain the knowledge models for the classes of handwritten '3's and handwritten '8's given by *usr1*. Then, to evaluate the level to which the handwritten number x depicted in Figure 16 satisfies the propositions "x BELONGS TO '8'" and "x BELONGS TO '3'", we use these models as input for the evaluation procedures described in Section 3.2, namely the procedure based on the DV and the procedure based on the MISV.

The results of those contextualized evaluations are listed in Table 1 and Figure 18. Notice in Table 1 that the levels computed with DV are the same levels computed with MISV. This observation

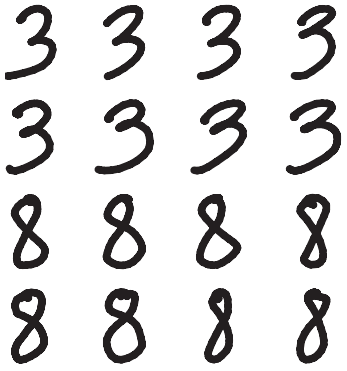

Figure 15 | User 2's training collection ($X_{0@usr2}$).

Figure 16 | User 1's test collection ($X_{@usr1}$).

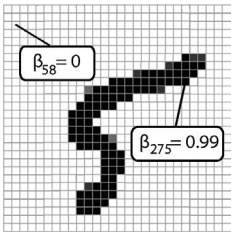

Figure 17 | Characterization of a handwritten '5'.

Table 1 Results of the evaluations of "x BELONGS TO '8' " and "x BELONGS TO '3' ", where '8' and '3' are two classes learned through $X_{0@usr1}$ and x is the handwritten number depicted in Figure 16.

	DV	MISV
$\check{\mu}_{\text{'8'}}(x)$	1.4317	1.4317
$\check{\nu}_{\text{'8'}}(x)$	1.2104	1.2104
$\eta_{\text{'8'}}(x)$	2.6422	2.6422
$\mu_{\text{'8'}}(x)$	0.5419	0.5419
$\nu_{\text{'8'}}(x)$	0.4581	0.4581
$\rho_{\text{'8'}}(x)$	0.0838	0.0838
$\check{\mu}_{\text{'3'}}(x)$	1.2104	1.2104
$\check{\nu}_{\text{'3'}}(x)$	1.4317	1.4317
$\eta_{\text{'3'}}(x)$	2.6422	2.6422
$\mu_{\text{'3'}}(x)$	0.4581	0.4581
$\nu_{\text{'3'}}(x)$	0.5419	0.5419
$\rho_{\text{'3'}}(x)$	−0.0838	−0.0838

Table 2 Results of the evaluations of "x BELONGS TO '8' " and "x BELONGS TO '3' ", where '8' and '3' are two classes learned through $X_{0@usr1} \cup X_{0@usr2}$ and x is the handwritten number depicted in Figure 16.

	DV	MISV
$\check{\mu}_{\text{'8'}}(x)$	1.4425	1.4425
$\check{\nu}_{\text{'8'}}(x)$	1.2318	1.2318
$\eta_{\text{'8'}}(x)$	2.6743	2.6743
$\mu_{\text{'8'}}(x)$	0.5394	0.5394
$\nu_{\text{'8'}}(x)$	0.4606	0.4606
$\rho_{\text{'8'}}(x)$	0.0788	0.0788
$\check{\mu}_{\text{'3'}}(x)$	1.2318	1.2318
$\check{\nu}_{\text{'3'}}(x)$	1.4425	1.4425
$\eta_{\text{'3'}}(x)$	2.6743	2.6743
$\mu_{\text{'3'}}(x)$	0.4606	0.4606
$\nu_{\text{'3'}}(x)$	0.5394	0.5394
$\rho_{\text{'3'}}(x)$	−0.0788	−0.0788

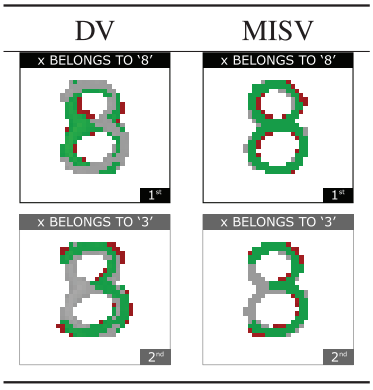

Figure 18 Visual results of the evaluations listed in Table 1.

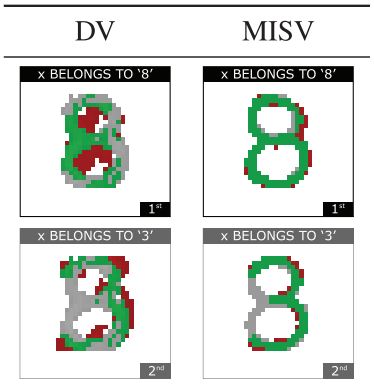

Figure 19 Visual results of the evaluations listed in Table 2.

makes the equivalence between (21) and (29), as well as the equivalence between (22) and (30) evident. In contrast, the visual representations listed in Figure 18 provide an evidence of the difference between the context of the evaluation obtained with DV and the context of the evaluation obtained with MISV. In these representations, while the green parts suggest that a handwritten number is part of a class, the red part, which a member of the class should have, and the gray part, which a member of the class should not have, indicate that the handwritten number is not part of the class. Notice in this case that, even though a difference between those contexts exists, this difference is rather small.

A potential explanation for such a small difference is that the DV, which incorporates all the support vectors, and the MISV are substantially similar since the knowledge models used for the previous evaluations have been obtained from a training collection consisting of numbers written only by one person, namely *user1*.

To obtain further insight in that regard, we use $X_{0@usr1} \cup X_{0@usr2}$ (see Figures 14 and 15) as a training collection to obtain the knowledge models for the classes of handwritten '3's and handwritten '8's given by both *usr1* and *usr2*. The resulting models were used as input of DV and MISV for the evaluation of the number depicted in Figure 16.

The results of those evaluations are listed in Table 2 and Figure 19. Notice that the equivalence between (21) and (29), as well as the equivalence between (22) and (30) are also visible in Table 2. Notice also that the difference between the contexts of the evaluations obtained with DV and the contexts of the evaluations obtained with MISV has increased.

An explanation for that increment is that in this case the DV incorporates features of the handwritten numbers given by *usr2* whose influence differs from the influence of the features included in the MISV, which includes features of one of the handwritten numbers given by *usr1*. For instance, the influence of the features of the handwritten '8"s given by *usr2* is reflected in the red part of the visual representation of the evaluation of "x BELONGS TO '8' " performed with DV (see Figure 19). In contrast, the visual representation of the evaluation of "x BELONGS TO '8' " performed with MISV only reflects the influence of the MISV, which is related to one of the '8"s given by *usr1* – recall that x represents the handwritten number depicted in Figure 16, which is given by *usr1*.

Regarding the prediction of the class the handwritten number x, the contextualized evaluations of the propositions "x BELONGS TO '3' " and "x BELONGS TO '8' " have been sorted in descending order according to the computed buoyancy. Since only two classes have been considered in this case, the 2 best evaluated classes have been presented as the 2 most optimistic predictions.

It is worth mentioning that, even though the computed buoyancy is used for sorting the evaluations, it might be considered optional while offering an explanation with a contextualized evaluation. A reason for this it that, compared to the context, the computed buoyancy might have a limited significance for an explanation since the buoyancy could be a very small number – notice in Tables 1

and 2 the effect of scaling the membership levels $\breve{\mu}_{`8'}(x)$ and $\breve{\mu}_{`3'}(x)$ and the nonmembership levels $\breve{\nu}_{`8'}(x)$ and $\breve{\nu}_{`3'}(x)$ in order to satisfy the consistency conditions $0 \leqslant \mu_{`8'}(x) + \nu_{`8'}(x) \leqslant 1$ and $0 \leqslant \mu_{`3'}(x) + \nu_{`3'}(x) \leqslant 1$ respectively. For this reason, offering the k most optimistic predictions instead of a unique prediction can help a user to make a better decision – cf. the work of Alonso and Bugarín [19] where additional classes are highlighted in case of ambiguity.

4.2. XSVMC on a Large Collection

To illustrate how XSVMC works in cases where the number of samples is greater that the dimension of the feature space $\mathcal{M}$, in this section we use the MNIST collection. This collection is composed of a training collection of 60000 samples and a test collection of 10000 samples, which have been used for benchmarking several classifiers [18].

In contrast to the binary classification performed in the previous section, in this section we use XSVMC to perform a multi-class classification of handwritten decimal digits. In this regard, we use an 'one-versus-the-rest' strategy to build each of the 10 training collections. For instance, to build the training collection for the class of handwritten 8's, the handwritten numbers with the tag '8' in the MNIST training collection were considered as positive examples while the other numbers without the tag '8' were considered as negative examples. These 10 collections were used as input of the XSVMC learning process to obtain knowledge models for the 10 handwritten decimal digits. The resulting models were used as input for the evaluation processes described in Section 3.2, i.e., DV and MISV, to evaluate each of the 10000 handwritten numbers included in the test collection.

The visual representations of two of those contextualized evaluations are shown in Figure 20. While the first column shows the visual representation of the evaluation of "x BELONGS TO '8'" performed with DV, the second column shows the visual representation of the same evaluation performed with MISV. In these representations, while the green parts suggest that the handwritten number is an '8', the red parts, which an '8' should have, and the gray part, which an '8' should not have, indicate that the handwritten number is not an '8'. Notice that the representation on the second column shows more plainly what has been relevant during the evaluation process.

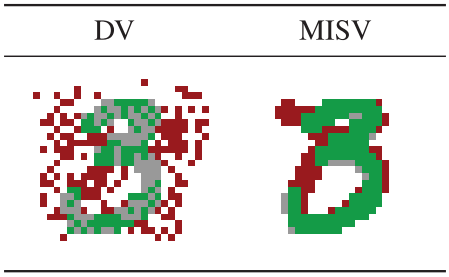

Figure 20 | Difference between the context of an evaluation based on the directional vector (DV) and the context of an evaluation based on the support vector with the greatest positive influence on the evaluation (MISV).

An explanation for the difference between the above representations is that in this case the DV incorporates features of several shapes of handwritten numbers '8"s whose influence differs from the influence of the features included in the MISV, which is related to the shape of a particular handwritten number '8' – cf. the visual representations listed in Figure 19.

To predict the class of a handwritten number, contextualized evaluations of the membership (and nonmembership) of this number to each of the 10 classes of handwritten decimal digits have been first performed. Then, these evaluations have been sorted in descending order according to the computed buoyancy. After that, the k best evaluated classes have been presented as the k most optimistic predictions. For the sake of illustration, Table 3 and Figure 21 show the results of the evaluations of the membership and nonmembership of a handwritten 4 in each of the 10 classes of handwritten decimal digits using a kernel $(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 2$. In this case, the three best

Table 3 | Results of the evaluations of the membership and nonmembership of a handwritten '4', denoted by x, in each of the classes of handwritten decimal digits.

A	$\mu_A(x)$	$\nu_A(x)$	$\rho_A(x)$	Rank
'0'	0.0181	0.0410	−0.0229	10^{th}
'1'	0.0280	0.0489	−0.0209	8^{th}
'2'	0.0299	0.0500	−0.0201	7^{th}
'3'	0.0434	0.0627	−0.0193	6^{th}
'4'	**0.1195**	**0.1083**	**0.0112**	1^{st}
'5'	0.0325	0.0518	−0.0193	9^{th}
'6'	0.0180	0.0398	−0.0218	7^{th}
'7'	0.0809	0.0971	−0.0162	4^{th}
'8'	**0.0711**	**0.0790**	**−0.0079**	3^{rd}
'9'	**0.1487**	**0.1562**	**−0.0075**	2^{nd}

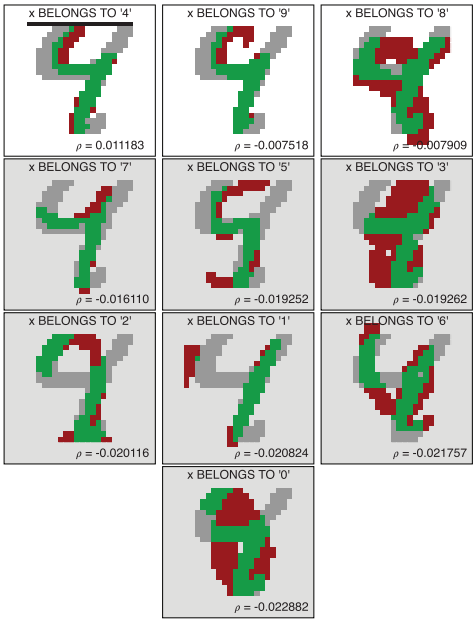

Figure 21 | Visual results of the evaluations of the membership and nonmembership of a handwritten '4' in each of the classes of handwritten decimal digits.

evaluated classes have been presented as the three most optimistic predictions.

The previous results were used as input of an XAI system that incorporates XSVMC to offer the following explanation of the most optimistic prediction: "*The green part suggests that your drawing is a '4' with a computed grade of* 0.1195*; however, the red part, which a '4' should have, and the gray part, which a '4' should not have, indicate that it is not a '4' with a computed grade of* 0.1083. Notice that not only the predicted class, but also the reasons behind that prediction are given.

A potential advantage of XSVMC over a conventional SVM classification process is shown in Table 4 and Figure 22. In this example, a conventional SVM classification process would offer '4' as a prediction since the best evaluated class is '4'. In contrast, since XSVMC offers the 3 most optimistic contextualized predictions in this case, based on the provided context, users might give preference to '7' which seems to be the class with the best credible justification.

To further illustrate the potential advantage of XSVMC over a conventional SVM classification process, we measure the number of right predictions included in the k best optimistic contextualized predictions. Table 5 shows the number of right predictions made by XSVMC with a polynomial kernel $(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 2$. Notice that 9985 out of 10000 predictions are included in the top 3 of the optimistic contextualized predictions, which represents an error rate of 0.15% – cf. the error rates reported for the MNIST collection [18].

It is worth mentioning that in situations where only the best evaluated class is presented as the most optimistic prediction, a conventional SVM classification process and an XSVMC process have the same performance. To prove that, the test collection of 10000 handwritten numbers has been used as input of both processes with several kernel configurations. The results are listed in Table 6. Notice that the error rate is the same for both classifiers.

Table 4 | Results of the evaluations of the membership and nonmembership of a handwritten '7' in each of the classes of handwritten decimal digits.

A	$\mu_A(x)$	$\nu_A(x)$	$\rho_A(x)$	Rank
'0'	0.5056	0.0532	−0.0260	10^{th}
'1'	0.0459	0.0681	−0.0222	5^{th}
'2'	0.0305	0.0564	−0.0259	9^{th}
'3'	0.0445	0.0670	−0.0225	6^{th}
'4'	**0.1370**	**0.1360**	**0.0010**	1^{st}
'5'	0.0406	0.0630	−0.0224	7^{th}
'6'	0.0231	0.0485	−0.0254	8^{th}
'7'	**0.1203**	**0.1227**	**−0.0024**	2^{nd}
'8'	0.0699	0.0852	−0.0153	4^{th}
'9'	**0.1750**	**0.1872**	**−0.0122**	3^{rd}

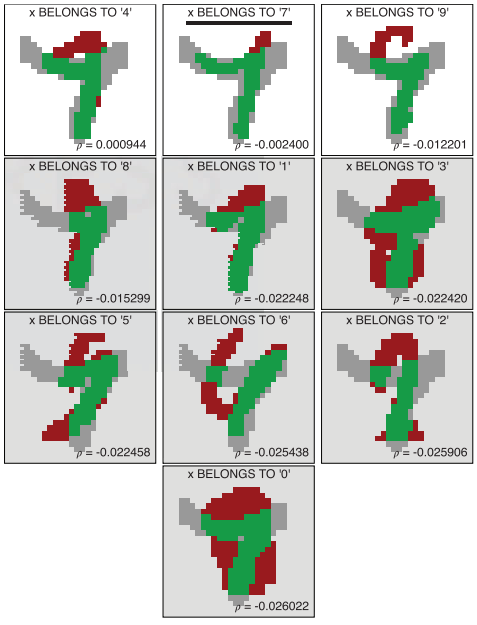

Figure 22 | Visual results of the evaluations of the membership and nonmembership of a handwritten '7' in each of the classes of handwritten decimal digits.

4.3. XSVMC versus Alternative Approaches

To illustrate potential advantages of XSVMC over alternative approaches, we use LIME [20] and ABELE [21] to perform the evaluations of the handwritten numbers considered in Figures 21 and 22.

LIME is a technique that tries to explain a prediction made by a classifier through an interpretable local model that is built around the prediction without knowing the details of the classifier. To produce the visual representations depicted in Figures 23 and 24, we use the source code of LIME (which is available in https://github.com/marcotcr/lime) with the same configuration of

Table 5 | Number of right predictions according to the ranking of the k best optimistic contextualized predictions (Kernel: $(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 2$).

Rank	Right Predictions	Freq. Acc.
1^{st}	**9790**	**9790**
2^{nd}	**169**	**9959**
3^{rd}	**26**	**9985**
4^{th}	8	9993
5^{th}	4	9997
6^{th}	2	9999
7^{th}	0	9999
8^{th}	1	10000
9^{th}	0	10000
10^{th}	0	10000

Table 6 | XSVMC versus Conventional SVM classification.

Kernel	Error Rate	
	XSVM	SVM
$(\mathbf{x_i} \cdot \mathbf{x_k}), C = 2$	8.19%	8.19%
$(\mathbf{x_i} \cdot \mathbf{x_k}), C = 4$	8.07%	8.07%
$(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 0$	2.46%	2.46%
$(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 2$	2.10%	2.10%
$(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 4$	2.11%	2.11%
$(\mathbf{x_i} \cdot \mathbf{x_k})^7, C = 2$	2.85%	2.85%

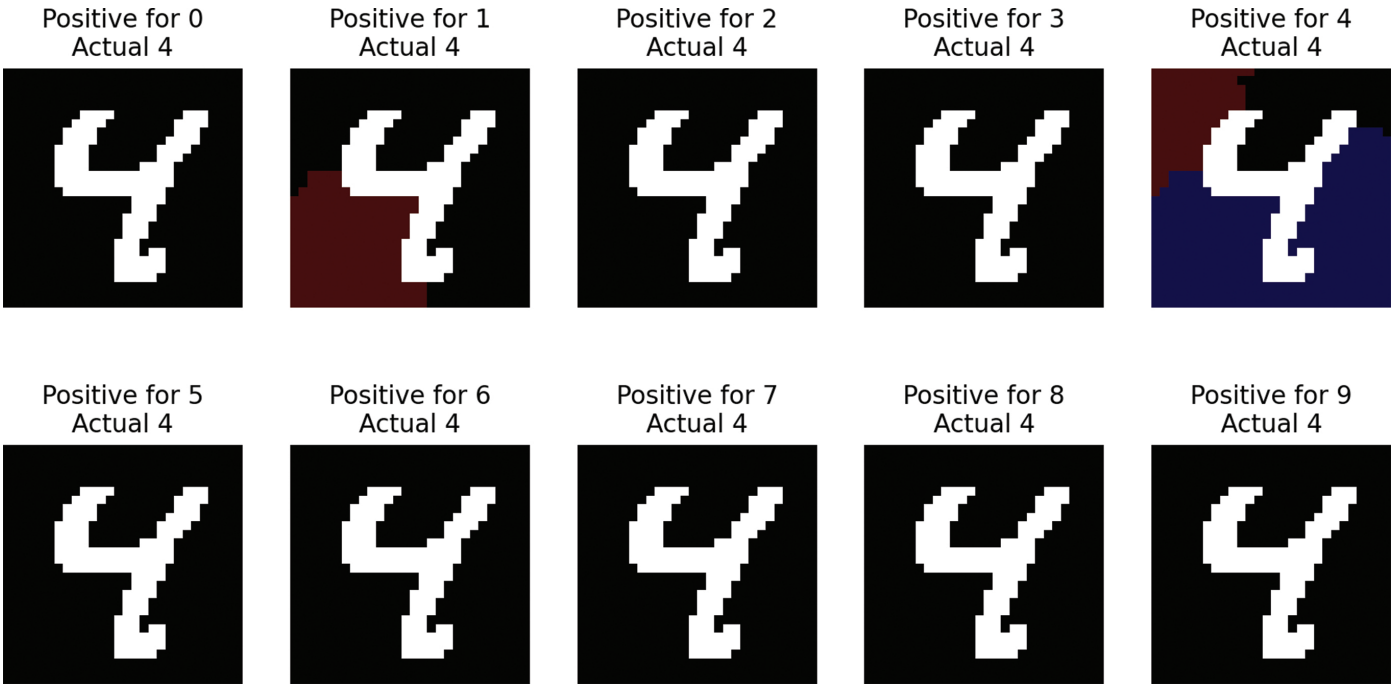

Figure 23 | Visual results of the evaluation of the handwritten number '4' depicted in Figure 21 using LIME.

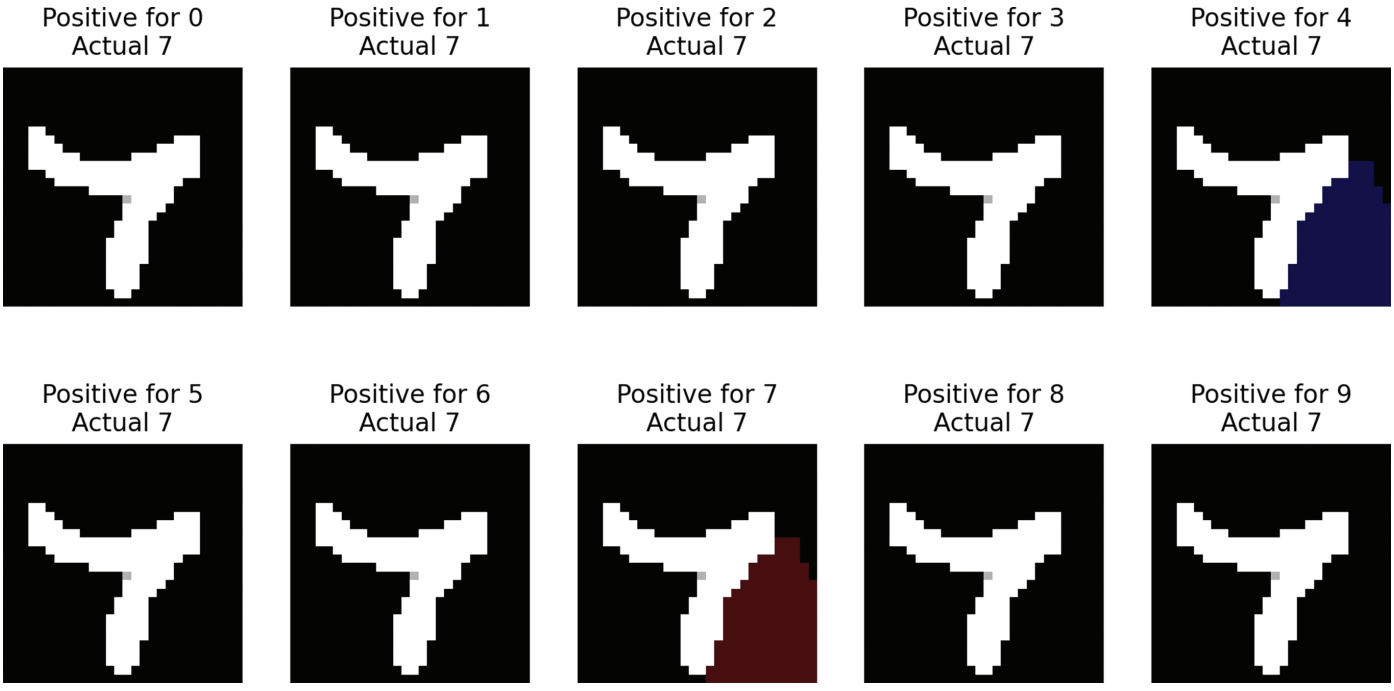

Figure 24 | Visual results of the evaluation of the handwritten number '7' depicted in Figure 22 using LIME.

the SVM classifier used in Figures 21 and 22 (i.e., a polynomial kernel $(\mathbf{x_i} \cdot \mathbf{x_k})^5, C = 2$). In both cases, the local model was built using 1000 synthetic samples. Notice that, in comparison to the visual representations produced with XSVMC, the visual representations produced with LIME show less plainly what has been relevant for the classifier during the evaluation process. In addition, while XSVMC needs only one evaluation to determine what has been relevant, LIME needs to evaluate all the generated synthetic samples.

Regarding ABELE, it is an extension of LORE [22] that, in a similar way to LIME, tries to explain a prediction by building an interpretable local classifier with a synthetic neighborhood of the handwritten number under evaluation, but, in addition, it takes into account existing relationships between the pixels of the handwritten number for building the synthetic neighborhood. To produce the visual representations depicted in Figure 25, we use the source code of ABELE (which is available in https://github.com/riccotti/ABELE) with the implemented

Figure 25 | Visual results of the evaluation of the handwritten numbers '4' and '7' depicted in Figures 21 and 22 respectively using ABELE.

Random Forest [23] classifier. Notice that, in comparison to LIME, the visual representations produced with ABELE show more clearly the approximations of what has been relevant to the classifier. However, ABELE needs more computational resources than LIME while evaluating all the generated synthetic samples.

5. RELATED WORK

An extensive survey of methods proposed for explaining computers predictions is presented in the work of Guidotti *et al.* [24]. In that survey, two main strategies have been identified: one is about the design of "transparent algorithms" that produce interpretable predictions, and the other is concerned with the interpretation of predictions without knowing the details of the algorithms that yield such predictions.

One of the methods aiming to interpret (and understand) predictions without knowing the internal details is a method proposed for decomposing a nonlinear image classification decision [25]. That method produces a heat map that highlights the relevant pixels, i.e., the pixels that have a significant influence on the classification decision. Another example is an explanation technique which tries to explain the predictions made by unknown classifiers by building interpretable local models that mimic the behavior of such classifiers [20]. In a similar way, the method proposed in the work of Baehrens *et al.* tries to extract a local model consisting of "explanation vectors," which contain features that are relevant to a given prediction [26]. The visualization method proposed by Zeiler and Fergus also tries to interpret the influence of the features (pixels) and the behavior of a specific knowledge model [27].

A particularity about the aforementioned methods is that they try to identify what has been relevant to the classification decision after a prediction is made. In contrast, XSVMC identifies what has been relevant before the prediction. This aspect is deemed to be a key advantage since the influence of the features can be taken into account to guide the classification decision. In this regard, XSVMC can be considered to be part of the "transparent algorithms" identified by the above-mentioned survey. The method proposed by Loor and De Tré to contextualize naive Bayes predictions [28] is another example of such transparent algorithms.

A classification process that generates visual explanations has been proposed by Hendricks *et al.* [29]. In that process, images with annotated features are used as input to train an explanation model that combines classification and sentence generation in natural language. The process yields sentences including discriminative features that justify why an object belongs to the predicted class. However, such sentences do not include features justifying why the object does not belong to the class as XSVMC does.

The contributions made by the fuzzy logic community to the development of the explainable AI research field have been analyzed by Alonso *et al.* [30]. The results of that work suggest that the contributions made by the fuzzy logic community seem to be distant with the efforts made by the non-fuzzy community. However, that study suggests that those contributions can be linked to address the challenges arising in that field. In this regard, while potential options to develop XAI systems with fuzzy modeling have been proposed by Mencar and Alonso [31], non-fuzzy options have been proposed by Adadi and Berrada [32].

6. CONCLUSIONS

In this paper, we have proposed a novel variant of an SVM classification process by which the resulting predictions are contextualized in order to improve their interpretability. In the proposed variant, named XSVMC, the membership and nonmembership of an object in a particular class are evaluated in such a way that the context of the evaluation is explicitly recorded. Hence, predictions resulting from such contextualized evaluations can be explained with ease. In this regard, a key component of XSVMC is a novel evaluation method that makes use of the MISV to contextualize the evaluations.

An important aspect of XSVMC is that users can take advantage of such contextualized predictions to give preference to the class(es) with the best credible justification. We have illustrated this aspect through the implementation of a use case where the classes of handwritten numbers are predicted.

Even though the results of the aforementioned implementation suggest that the contextualization of SVM predictions can favor the interpretability of them, qualitative attributes like coherence, naturalness and clearness that might be perceived by a person on those predictions are still subject to validation. In this regard, studies oriented to conduct such validations are considered (and suggested) as future work.

CONFLICT OF INTEREST

None.

AUTHORS' CONTRIBUTION

Marcelo Loor, main author Guy De Tré, PhD thesis promoter, advisor.

FUNDING STATEMENT

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

ACKNOWLEDGMENTS

The authors acknowledge the valuable and insightful comments given by the anonymous reviewers. The authors are also very grateful to the persons who have written the numbers contained in the small collection used in Section 4.1.

REFERENCES

[1] H. Hagras, Toward human-understandable, explainable AI, Computer. 51 (2018), 28–36.

[2] M.E. Kaminski, The right to explanation, explained, Berkeley Tech. LJ. 34 (2019), 189.

[3] A. Preece, Asking 'why' in AI: explainability of intelligent systems–perspectives and challenges, Intell. Syst. Account. Finance Manag. 25 (2018), 63–72.

[4] W. Samek, T. Wiegand, K.-R. Müller, Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models, ITU J. ICT Discov. 1 (2017), 39–48.

[5] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, NY, USA, 1995.

[6] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, vol. 1, Wiley, New York, NY, USA, 1998.

[7] M. Loor, G. De Tré, On the need for augmented appraisal degrees to handle experience-based evaluations, Appl. Soft Comput. 54 (2017), 284–295.

[8] D. Gunning, Explainable Artificial Intelligence (XAI), 2017. http://www.darpa.mil/attachments/XAIIndustryDay_Final.pptx

[9] F. Herrera, F. Charte, A.J. Rivera, M.J. del Jesus, Multilabel Classification, Springer International Publishing, Cham, Switzerland, 2016, pp. 17–31.

[10] L.A. Zadeh, Fuzzy sets, Inf. Control. 8 (1965), 338–353.

[11] K.T. Atanassov, Intuitionistic fuzzy sets, Fuzzy Sets Syst. 20 (1986), 87–96.

[12] K.T. Atanassov, On Intuitionistic Fuzzy Sets Theory, vol. 283 of Studies in Fuzziness and Soft Computing, Springer, Berlin, Heidelberg, Germany, 2012.

[13] M. Loor, A. Tapia-Rosero, G. De Tré, Usability of concordance indices in FAST-GDM problems, in Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI 2018), Seville, Spain, 2018, pp. 67–78.

[14] M. Loor, G. De Tré, Explaining computer predictions with augmented appraisal degrees, in Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019), Prague, Czech Republic. Atlantis Press, 2019, pp. 158–165.

[15] M. Loor, G. De Tré, Identifying and properly handling context in crowdsourcing, Appl. Soft Comput. 73 (2018), 203–214.

[16] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discov. 2 (1998), 121–167.

[17] T. Joachims, Making large-scale SVM learning practical, in: B. Schölkopf, C.J.C. Burges, A. Smola (Eds.), Advances in Kernel Methods - Support Vector Learning, chap. 11, MIT Press, Cambridge, MA, USA, 1999, pp. 169–184.

[18] Y. LeCun, C. Cortes, C.J.C. Burges, The MNIST database of handwritten digits. 1998. http://yann.lecun.com/exdb/mnist/

[19] J.M. Alonso, A. Bugarín, Expliclas: automatic generation of explanations in natural language for weka classifiers, in 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 2019, pp. 1–6.

[20] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 1135–1144.

[21] R. Guidotti, A. Monreale, S. Matwin, D. Pedreschi, Black box explanation by learning image exemplars in the latent feature space, in: U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, C. Robardet (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, Switzerland, 2020, pp. 189–205.

[22] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, IEEE Intell. Syst. 34 (2019), 14–23.

[23] L. Breiman, Random forests, Mach. Learn. 45 (2001), 5–32.

[24] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2018) 1–42.

[25] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLOS ONE. 10 (2015), 1–46.

[26] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.-R. MÃžller, How to explain individual classification decisions, J. Mach. Learn. Res. 11 (2010), 1803–1831.

[27] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in European Conference on Computer Vision, Springer, Zurich, Switzerland, 2014, pp. 818–833.

[28] M. Loor, G. De Tré, Contextualizing Naive Bayes predictions, in: M.J. Lesot, S. Vieira, M.Z. Reformat, J.P. Carvalho, A. Wilbik, B. Bouchon-Meunier, R.R. Yager (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer International Publishing, Cham, Switzerland, 2020, pp. 814–827.

[29] L.A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, T. Darrell, Generating visual explanations, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision - ECCV 2016, Springer International Publishing, Cham, Switzerland, 2016, pp. 3–19.

[30] J.M. Alonso, C. Castiello, C. Mencar, A bibliometric analysis of the explainable artificial intelligence research field, in: J. Medina, M. Ojeda-Aciego, J.L. Verdegay, D.A. Pelta, I.P. Cabrera, B. Bouchon-Meunier, R.R. Yager (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Theory and Foundations, Springer International Publishing, Cham, Switzerland, 2018, pp. 3–15.

[31] C. Mencar, J.M. Alonso, Paving the way to explainable artificial intelligence with fuzzy modeling, in: R. Fullér, S. Giove, F. Masulli (Eds.), Fuzzy Logic and Applications, Springer International Publishing, Cham, Switzerland, 2019, pp. 215–227.

[32] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on Explainable Artificial Intelligence (XAI), IEEE Access. 6 (2018), 52138–52160.