



# The semiproximal SVM approach for multiple instance learning: a kernel-based computational study

Matteo Avolio<sup>1</sup> · Antonio Fuduli<sup>1</sup>

Received: 13 February 2023 / Accepted: 25 May 2023 / Published online: 14 June 2023  
© The Author(s) 2023

## Abstract

The semiproximal Support Vector Machine technique is a recent approach for Multiple Instance Learning (MIL) problems. It exploits the benefits exhibited in the supervised learning by the Support Vector Machine technique, in terms of generalization capability, and by the Proximal Support Vector Machine approach in terms of efficiency. We investigate the possibility of embedding the kernel transformations into the semiproximal framework to further improve the testing accuracy. Numerical results on benchmark MIL data sets show the effectiveness of our proposal.

**Keywords** Multiple instance learning · Support vector machine · Semiproximal support vector machine · Kernel transformations

## 1 Introduction

Multiple Instance Learning (MIL) [20] deals with classification of point sets: such sets are named bags and the corresponding points inside the bags are called instances. The main peculiarity of a MIL problem, with respect to the classical supervised classification, resides in the fact that only the labels of each overall bag give a contribution to the learning phase, since the labels of the instances belonging to the bags are unknown.

The first MIL problem proposed in the literature is a drug design problem [12], aimed at discriminating between active and non-active molecules (bags) on the basis of the possible three-dimensional conformations (instances) they can assume. MIL applications can be found in various fields: text categorization, image recognition, medical diagnosis, bankruptcy prediction and so on.

---

✉ Antonio Fuduli  
antonio.fuduli@unical.it

Matteo Avolio  
matteo.avolio@unical.it

<sup>1</sup> Department of Mathematics and Computer Science, University of Calabria, Rende, Italy

In this work we focus on binary MIL problems, whose aim is to discriminate between positive and negative bags. For such problems a crucial question, related to the number of classes of instances, is what we mean by a positive bag in contrast to a negative one. In the case of two classes of instances, a very common assumption in the literature is the so-called *standard MIL assumption*, stating that a bag is positive if and only if it contains at least a positive instance. A typical application is in medical diagnosis by means of images [22]: a patient is considered positive if and only if his/her medical scan (bag) is characterized by at least an abnormal subregion (instance) and is negative if and only if all the subregions of his/her medical scan are normal.

For solving a MIL problem, in the literature there exist various approaches [1, 9], depending on the space where the classification process is performed. In particular, in the instance-space approaches the classification is carried out at the instance level and the class label of each bag is obtained as aggregation of the information coming from the labels of the corresponding instances. On the other hand, if the classification process is performed at the global level, i.e. considering each overall bag as a global entity, we say that the corresponding approach is of the bag-space type. A compromise between the instance-space and the bag-space approaches is given by the so-called embedding-space approaches, where each bag is represented by a specific instance and only such representative instances will successively contribute to the classification process. Some instance-space approaches are mi-SVM [2], MICA [21], MIC<sup>bundle</sup> [8] and, more recently, MIL-RL [5] and mi-SPSVM [7]. An embedding-space approach is MI-SVM [2], while some bag-space techniques can be found in [19, 26, 27]. Finally, very recently, a semi-embedding-space technique (Algorithm MI-MSph) has been designed in [3], where the positive bags are represented by the respective barycenters, maintaining the representation of the negative bags in terms of their original instances.

In this paper we propose a kernel version of the recent MIL semiproximal Support Vector Machine approach presented in [7] and based on the combination of two different philosophies designed for supervised classification: the well-established Support Vector Machine (SVM) technique, characterized by a good generalization capability (see for example [10]), and the Proximal Support Vector Machine (PSVM) approach, which has exhibited a reasonable compromise between efficiency and accuracy (see [17]).

The kernel trick is a well-known technique providing nonlinear separation surfaces in the SVM framework (see for example [23]). If, on one hand, using kernel functions substantially improves the final accuracy of the classifier, on the other hand the kernel techniques in general exhibit higher computational times, which could limit their use in solving large scale problems. Despite of that, our semiproximal kernel-based approach exhibits quite comparable CPU times and significantly better accuracy with respect to the linear version of the algorithm, whose computational efficiency has been amply shown in [7] by means of extensive numerical experiments.

The paper is organized in the following way. In the next section we formalize the binary MIL problem and we recall the mi-SPSVM algorithm presented in [7]. In Sect. 3 we present the corresponding kernelized version of mi-SPSVM and in

Sect. 4 some numerical experiments are proposed on a set of benchmark MIL problems. Finally in Sect. 5 some conclusions are drawn.

Throughout the paper we indicate by  $\|x\|$  the Euclidean norm of the vector  $x$  and by  $x^T y$  the scalar product between the vectors  $x$  and  $y$ .

## 2 The semiproximal SVM approach for MIL

Given  $m$  positive bags  $\mathcal{X}_i^+, i = 1, \dots, m$ , and  $k$  negative ones  $\mathcal{X}_i^-, i = 1, \dots, k$ , let  $J_i^+$  and  $J_i^-$  be the corresponding index sets such that

$$\mathcal{X}_i^+ = \{x_j \in \mathbb{R}^n \mid j \in J_i^+\} \quad i = 1, \dots, m$$

and

$$\mathcal{X}_i^- = \{x_j \in \mathbb{R}^n \mid j \in J_i^-\} \quad i = 1, \dots, k,$$

with  $x_j \in \mathbb{R}^n$  being a generic instance.

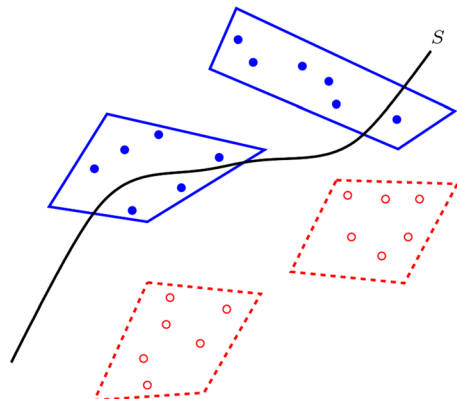
Under the standard MIL assumption, the objective is to find a separation surface  $S$  such that all the instances of the negative bags lie on one side with respect to  $S$  and, for each positive bag, at least an instance lies on the other side (see Fig. 1, with four bags: two positive and two negative).

The semiproximal SVM technique (Algorithm mi-SPSVM in [7]) is an instance-space approach which generates a separation hyperplane

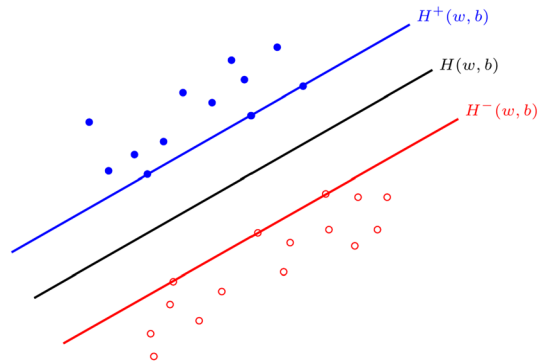
$$H(w, b) \triangleq \{x \in \mathbb{R}^n \mid w^T x + b = 0\},$$

by exploiting the nice properties exhibited by both the SVM (good accuracy) and PSVM (good efficiency) techniques for supervised classification. In particular, given two finite sets of points (instances)  $\mathcal{X}^+$  and  $\mathcal{X}^-$ , indexed respectively by  $J^+$  and  $J^-$ , in the standard SVM approach the quadratic programming problem

**Fig. 1** A surface  $S$  separating two positive bags (continuous polygons) and two negative ones (dashed polygons). The circles inside the bags are the instances



**Fig. 2** A separating hyperplane  $H$  provided by the SVM approach for supervised classification.  $H^+$  and  $H^-$  are the supporting hyperplanes



$$\begin{cases} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{j \in J^+} \xi_j + C \sum_{j \in J^-} \xi_j \\ \xi_j \geq 1 - (w^T x_j + b) & j \in J^+ \\ \xi_j \geq 1 + (w^T x_j + b) & j \in J^- \\ \xi_j \geq 0 & j \in J^+ \cup J^- \end{cases} \tag{1}$$

is solved, providing a separation hyperplane placed in the middle between the two supporting hyperplanes (see Fig. 2):

$$H^+(w, b) \triangleq \{x \in \mathbb{R}^n \mid w^T x + b = 1\}$$

and

$$H^-(w, b) \triangleq \{x \in \mathbb{R}^n \mid w^T x + b = -1\}.$$

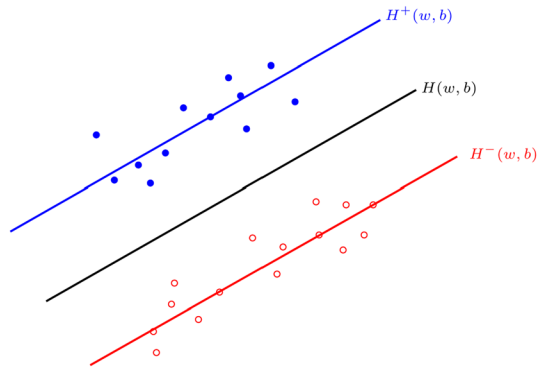
In problem (1), variables  $\xi_j$ s represent a measure of the misclassification error of the instances  $x_j$ s, while the positive constant  $C$  tunes the trade-off between the maximization of the margin (the area comprised between  $H^+$  and  $H^-$ ), obtained by minimizing the Euclidean norm of  $w$ , and the minimization of the misclassification error.

On the other hand the PSVM technique [17], for  $C > 0$ , solves the optimization problem

$$\begin{cases} \min_{w,b,\xi} \frac{1}{2} \left\| \frac{w}{b} \right\|^2 + \frac{C}{2} \sum_{j \in J^+} \xi_j^2 + \frac{C}{2} \sum_{j \in J^-} \xi_j^2 \\ \xi_j = 1 - (w^T x_j + b) & j \in J^+ \\ \xi_j = 1 + (w^T x_j + b) & j \in J^-, \end{cases} \tag{2}$$

providing a separation hyperplane placed in the middle between two proximal hyperplanes,  $H^+$  and  $H^-$ , which cluster the points of  $\mathcal{X}^+$  and  $\mathcal{X}^-$ , respectively (see Fig. 3).

**Fig. 3** A separating hyperplane  $H$  provided by the PSVM approach for supervised classification.  $H^+$  and  $H^-$  are the proximal (clustering) hyperplanes



Differently from problem (1), problem (2) is strictly convex with respect to both  $w$  and  $b$  and it can be solved in a closed form, improving significantly the efficiency with respect to the classical SVM technique.

Coming back to the solution of the MIL problem introduced at the beginning of this section, the main idea characterizing the semiproximal approach (Algorithm mi-SPSVM in [7]) takes into account the different roles played by the positive and the negative bags. In fact, considering the standard MIL assumption, a correct classification of the negative bags requires more effort with respect to the classification of the positive ones, the former needing the correct classification of all the instances and the latter of at least one. In other words, classifying correctly the positive bags is indeed easier than classifying the negative ones, allowing  $H^+$  to possibly be a proximal (clustering, instead of supporting) hyperplane for the instances of the positive bags.

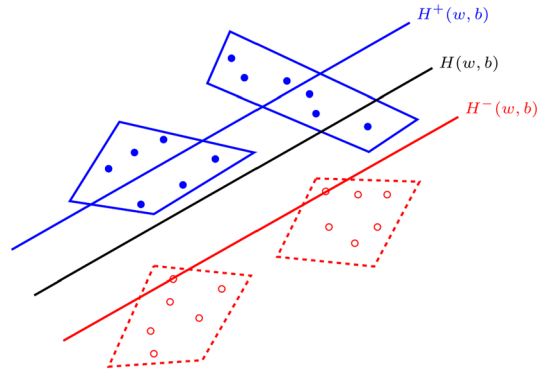
In particular, indicating by  $J^+$  the index set of all the instances of the positive bags and by  $J^-$  the index set of all the instances of the negative bags, the mi-SPSVM algorithm initially solves the quadratic program

$$\left\{ \begin{array}{l} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{j \in J^+} \xi_j^2 + C \sum_{j \in J^-} \xi_j \\ \xi_j = 1 - (w^T x_j + b) \quad j \in J^+ \\ \xi_j \geq 1 + (w^T x_j + b) \quad j \in J^- \\ \xi_j \geq 0 \quad j \in J^-, \end{array} \right. \quad (3)$$

providing a separation hyperplane located in the middle between a supporting hyperplane  $H^-$  for the instances of the negative bags and a proximal hyperplane  $H^+$  which clusters the instances of the positive bags (see Fig. 4).

The algorithm, named mi-SPSVM and described in Algorithm 1, proceeds by repeatedly solving problem (3) for different configurations of the sets  $J^+$  and  $J^-$ , in order to compute the current separation hyperplane  $H(\bar{w}, \bar{b})$ . In particular, the core of the procedure is constituted by steps 6-11, aimed at determining the sets  $J^*$  and

**Fig. 4** The semiproximal SVM hyperplane  $H$  separating two positive bags (continuous polygons) and two negative ones (dashed polygons). The circles inside the bags are the instances



$\bar{J}$  needed to update  $J^+$  and  $J^-$  at steps 13-14. While the satisfaction of the standard MIL assumption is guaranteed by means of the set  $J^*$  whose indexes remain in  $J^+$ , at each iteration the cardinality of  $J^-$  is increased by adding the indexes of the set  $\bar{J}$ : such indexes indeed correspond to the current positive instances which lie in the negative side with respect to the current hyperplane.

In [7] the finite termination of the algorithm has been shown, proving also that the sequence of the optimal objective function values of problem (3), generated by varying  $J^+$  and  $J^-$  at each iteration according to steps 13-14, is monotonically nonincreasing. Finally, the numerical experiments therein presented, in comparison with some other reimplemented approaches drawn from the literature, have proved a very good performance of mi-SPSVM, not only in terms of accuracy but also in terms of CPU time.

### 3 Embedding kernel transformations into the semiproximal approach

Kernel transformations constitute a very powerful trick useful to construct nonlinear classifiers, starting from a linear separation surface (hyperplane) obtained in a higher dimensional space.

---

**Algorithm 1: mi-SPSVM**

---

**Input:**  $C > 0$   
**Output:**  $\bar{w}, \bar{b}$

▷Initialization

- 1  $J^+ \leftarrow \cup_{i=1}^m J_i^+$
- 2  $J^- \leftarrow \cup_{i=1}^k J_i^-$
- 3  $\bar{J} \leftarrow J^+$
- 4 **repeat**
  - ▷Computing the separation hyperplane  $H(\bar{w}, \bar{b})$
  - 5 Solve problem (3) and compute  $\bar{w}$  and  $\bar{b}$ 
    - ▷Computing  $J^*$
  - 6  $J^* \leftarrow \emptyset$
  - 7 **for**  $i \leftarrow 1, \dots, m$  **do**
  - 8      $j_i^* \leftarrow \arg \max_{j \in (J_i^+ \cap J^+)} \{\bar{w}^T x_j + \bar{b}\}$
  - 9     **if**  $\bar{w}^T x_{j_i^*} + \bar{b} \leq -1$  **then**
  - 10          $J^* \leftarrow J^* \cup \{j_i^*\}$
  - ▷Updating  $\bar{J}$
  - 11  $\bar{J} \leftarrow \{j \in J^+ \setminus J^* \mid \bar{w}^T x_j + \bar{b} \leq -1\}$ 
    - ▷Updating  $J^+$  and  $J^-$
  - 12 **if**  $\bar{J} \neq \emptyset$  **then**
  - 13      $J^+ \leftarrow J^+ \setminus \bar{J}$
  - 14      $J^- \leftarrow J^- \cup \bar{J}$
- 15 **until**  $\bar{J} = \emptyset$

---

In particular, let  $I$  be the input space, such that  $x_j \in I \subseteq \mathbb{R}^n$ , for any  $j \in J_i^+$ ,  $i = 1, \dots, m$ , and for any  $j \in J_i^-, i = 1, \dots, k$ , and let  $F \subseteq \mathbb{R}^N$  be the feature space such that  $N > n$ . Given a map

$$\phi : I \mapsto F,$$

the kernel function  $\mathcal{K}$  is defined as:

$$\mathcal{K} : I \times I \mapsto \mathbb{R}$$

such that

$$\mathcal{K}(x, y) = \phi(x)^T \phi(y).$$

Analogously to the standard SVM, embedding the kernel functions into Algorithm mi-SPSVM is related to the optimal solution to the Wolfe dual of problem (3):

$$\left\{ \begin{array}{l} \min_{\lambda, \mu} \frac{1}{2} \lambda^T X^+ X^{+T} \lambda + \frac{1}{2} \mu^T X^- X^{-T} \mu - \lambda^T X^+ X^{-T} \mu \\ \quad + \frac{1}{2} (e^T \lambda - e^T \mu)^2 + \frac{\|\lambda\|^2}{2C} - \lambda^T e - \mu^T e \\ 0 \leq \mu \leq Ce, \end{array} \right. \tag{4}$$

where  $\lambda$  is the vector containing the Lagrangian multipliers introduced in correspondence to the constraints

$$\xi_j = 1 - (w^T x_j + b), \quad j \in J^+,$$

$\mu$  is the vector containing the Lagrangian multipliers defined in correspondence to the constraints

$$\xi_j \geq 1 + (w^T x_j + b), \quad j \in J^-,$$

$e$  is the vector of ones of appropriate dimension,  $X^+$  is the matrix whose  $j$ th row is the vector  $x_j$ ,  $j \in J^+$ , and  $X^-$  is the matrix whose  $j$ th row is the vector  $x_j$ ,  $j \in J^-$ .

From the primal-dual relationships, the optimal solution  $(w^*, b^*)$  obtained by solving problem (3) is also computable by means of the following formulae:

$$w^* = X^{+T} \lambda^* - X^{-T} \mu^* \tag{5}$$

and

$$b^* = e^T \lambda^* - e^T \mu^*, \tag{6}$$

where  $(\lambda^*, \mu^*)$  is the optimal solution to problem (4). Note that, differently from the classical SVM model, the optimal value of the bias  $b^*$  is uniquely determined by (6), due to the strict convexity of the primal problem (3) with respect to  $w$  and  $b$ .

It is easy to show that rewriting the Wolfe dual problem (4) in terms of the kernel function  $\mathcal{K}$  gives:

$$\left\{ \begin{array}{l} \min_{\lambda, \mu} \frac{1}{2} \lambda^T K^{++} \lambda + \frac{1}{2} \mu^T K^{--} \mu - \lambda^T K^{+-} \mu \\ \quad + \frac{1}{2} (e^T \lambda - e^T \mu)^2 + \frac{\|\lambda\|^2}{2C} - \lambda^T e - \mu^T e \\ 0 \leq \mu \leq Ce, \end{array} \right. \tag{7}$$

where the generic elements of the matrices  $K^{++}$ ,  $K^{--}$  and  $K^{+-}$  are of the type:

$$\begin{aligned} K_{pq}^{++} &= \mathcal{K}(x_p, x_q), & \text{with } p, q \in J^+, \\ K_{pq}^{--} &= \mathcal{K}(x_p, x_q), & \text{with } p, q \in J^-, \end{aligned}$$



and

$$K_{pq}^{+-} = \mathcal{K}(x_p, x_q), \quad \text{with } p \in J^+, q \in J^-.$$

Note that, in order to tackle the above problem (7), it is enough to know only the analytical form of the kernel function  $\mathcal{K}$ , without needing explicitly the map  $\phi$  (this is the reason of the expression “the kernel trick”, very common in the literature). Moreover, problem (7) coincides exactly with problem (4) in case the linear kernel function  $\mathcal{K}(x, y) = x^T y$  is adopted.

We conclude this section by observing that using the kernel trick into Algorithm mi-SPSVM involves also steps 8 and 11, where, in order to compute the sets  $J^*$  and  $\bar{J}$ , the quantity  $\bar{w}^T \phi(x_j) + \bar{b}$  should be evaluated for  $j \in J^+$ . Taking into account formulae (5) and (6), with  $x_j$  substituted by  $\phi(x_j)$  for any  $j \in J^+ \cup J^-$ , it is easy to see that also in this case there is no need to know explicitly the map  $\phi$ , since such evaluation corresponds to calculating the following vector:

$$K^{++} \lambda^* - K^{+-} \mu^* + e^T \lambda^* - e^T \mu^*,$$

where  $(\lambda^*, \mu^*)$  is the optimal solution to problem (7).

### 4 Computational study of the semiproximal kernelized version

The kernelized version of Algorithm mi-SPSVM, named mi-KSPSVM, has been implemented in Matlab (version R2019b) and run on a Windows 10 system, characterized by 16 GB of RAM and a 2.30 GHz Intel Core i7 processor. About the choice of the kernel function, we have used the RBF (Radial Basis Function) kernel [25]

$$\mathcal{K}(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \tag{8}$$

which is the most common one adopted in the literature.

The code has been tested on the twelve most commonly used (in the literature) benchmark MIL problems [2], listed in Table 1: the first three data sets are image recognition problems, the TST ones are text categorization problems, while the last two ones are drug design problems.

A crucial issue regards the choice of the hyperparameters  $C$  and  $\sigma$ : the former adjusts the trade-off between the maximization of the margin and the minimization of the misclassification error, while the latter tunes the kernel function values in formula (8).

We have adopted the following strategies. About the choice of  $C$  we have used a bi-level cross validation approach [4] of the same type adopted in [7], using the same grid of values  $2^i$ , with  $i = -7, \dots, 7$ , as in [21]. As for  $\sigma$ , we have preliminarily investigated on some potential values taken inside the grid  $2^i$ , with  $i = -4, \dots, 7$ , noting that the best testing accuracies are obtained in correspondence to the values of  $i$  belonging to the set  $\{-3, -2, 2, 3, 4\}$ . Then, in order to automate the choice, we

**Table 1** Data sets

Data set	Dimension ( $n$ )	Instances	Bags ( $m + k$ )	Positive bags ( $m$ )	Negative bags ( $k$ )
Elephant	230	1391	200	100	100
Fox	230	1320	200	100	100
Tiger	230	1220	200	100	100
TST1	6668	3224	400	200	200
TST2	6842	3344	400	200	200
TST3	6568	3246	400	200	200
TST4	6626	3391	400	200	200
TST7	7037	3367	400	200	200
TST9	6982	3300	400	200	200
TST10	7073	3453	400	200	200
Musk-1	166	476	92	47	45
Musk-2	166	6598	102	39	63

have embedded the computation of  $\sigma$  into the bi-level cross validation using the grid  $2^i$ , with  $i = -3, -2, 2, 3, 4$ .

In Table 2 we report the results in terms of average training correctness, average testing correctness (accuracy) and average CPU time, compared with the linear kernel semiproximal version mi-SPSVM [7] run on the same machine as mi-KSPSVM. Highlighting in bold the best correctness value for each data set, it appears clear that, while in terms of computational times the two algorithms are quite comparable,

**Table 2** mi-KSPSVM vs mi-SPSVM [7]: average training correctness, average testing correctness (accuracy) and average CPU time

Data set	mi-KSPSVM			mi-SPSVM		
	Training (%)	Testing (%)	CPU time (secs)	Training (%)	Testing (%)	CPU time (secs)
Elephant	<b>84.78</b>	<b>81.00</b>	0.68	82.00	76.50	0.83
Fox	<b>89.33</b>	<b>63.00</b>	0.42	71.39	59.00	1.64
Tiger	<b>91.56</b>	<b>83.00</b>	0.88	85.33	74.50	0.86
TST1	<b>99.53</b>	<b>95.25</b>	5.50	97.44	94.25	4.10
TST2	<b>99.75</b>	<b>81.25</b>	5.38	88.50	74.50	5.06
TST3	<b>98.75</b>	<b>88.50</b>	4.33	96.44	86.25	5.33
TST4	<b>98.69</b>	<b>85.75</b>	6.77	91.33	81.75	5.93
TST7	<b>96.81</b>	<b>83.25</b>	5.58	92.42	81.75	5.90
TST9	<b>97.61</b>	<b>70.75</b>	7.14	88.00	68.50	6.04
TST10	<b>97.69</b>	<b>83.50</b>	7.33	96.14	78.75	7.70
Musk-1	<b>99.52</b>	<b>86.67</b>	0.04	93.25	82.22	0.06
Musk-2	<b>94.67</b>	<b>79.00</b>	25.47	86.74	73.00	216.86

on the other hand mi-KSPSVM significantly overcomes mi-SPSVM not only in terms of accuracy, but also in terms of average training correctness.

For the sake of completeness, in Table 3 we report the average testing results of mi-KSPSVM and mi-SPSVM, expressed in terms of sensitivity, specificity and F-score. While the sensitivity and the specificity express the capability of the classifier to correctly identify the positive bags (sensitivity) and the negative ones (specificity), the F-score is the harmonic mean of sensitivity and precision, the latter expressing the percentage of the true positive bags among all the bags classified as positive. From Table 3 (where, for each data set and for each evaluation metric, the best value is highlighted in bold), mi-KSPSVM appears to be the best performant also in terms of specificity and F-score. We recall that, in case the accuracy is not equal to 100%, low values of sensitivity [resp. specificity] are generally a consequence of high values of specificity [resp. sensitivity].

In order to investigate the behaviour of the code with respect to the literature, in Table 4 we also present the comparison of our accuracy results with those ones obtained by the following approaches, for which we have reported the best published values obtained using indifferently the linear or the nonlinear kernel:

- mi-SVM [2]: it is an instance-space approach based on solving heuristically a SVM type mixed integer program, by means of a BCD (Block Coordinate Descent) method [24].
- MI-SVM [2]: it is an embedding-space approach, where each positive bag is represented by a single feature vector, the instance furthest from the current hyper-plane.
- MICA: [21]: it is an instance-space approach, where each positive bag is represented by the convex combination of its instances.

**Table 3** mi-KSPSVM vs mi-SPSVM [7]: average testing results in terms of sensitivity, specificity and F-score

Data set	mi-KSPSVM			mi-SPSVM		
	Sensitivity (%)	Specificity (%)	F-score (%)	Sensitivity (%)	Specificity (%)	F-score (%)
Elephant	98.12	<b>64.57</b>	<b>83.46</b>	<b>98.46</b>	55.99	80.43
Fox	82.42	<b>43.10</b>	68.48	<b>92.70</b>	24.24	<b>68.59</b>
Tiger	<b>94.35</b>	<b>71.87</b>	<b>84.42</b>	90.22	58.13	77.21
TST1	93.94	<b>94.46</b>	<b>95.09</b>	<b>94.37</b>	93.93	94.13
TST2	82.61	<b>78.92</b>	<b>81.83</b>	<b>83.20</b>	65.47	76.44
TST3	90.82	<b>86.74</b>	<b>88.88</b>	<b>91.16</b>	81.40	87.12
TST4	<b>85.80</b>	<b>85.73</b>	<b>84.26</b>	85.41	79.77	80.61
TST7	81.77	<b>84.07</b>	<b>82.64</b>	<b>83.18</b>	80.66	81.76
TST9	77.63	<b>64.61</b>	<b>72.49</b>	<b>78.21</b>	59.26	71.24
TST10	<b>83.65</b>	<b>83.67</b>	<b>81.91</b>	82.86	77.92	77.79
Musk-1	91.81	<b>85.42</b>	<b>83.85</b>	<b>96.90</b>	75.50	81.53
Musk-2	69.00	<b>85.44</b>	<b>69.11</b>	<b>77.17</b>	70.63	66.24

**Table 4** mi-KSPSVM vs mi-SVM [2], MI-SVM [2], MICA [21], MIC<sup>bundle</sup> [8] and MI-Kernel [19]: average testing correctness (accuracy)

Data set	mi-KSPSVM (%)	mi-SVM (%)	MI-SVM (%)	MICA (%)	MIC <sup>bundle</sup> (%)	MI-Kernel (%)
Elephant	81.00	<b>82.20</b>	81.40	80.50	80.50	–
Fox	<b>63.00</b>	58.20	59.40	58.70	58.30	–
Tiger	83.00	78.90	<b>84.00</b>	82.60	79.10	–
TST1	<b>95.25</b>	93.60	93.90	94.50	–	–
TST2	81.25	78.20	84.50	<b>85.00</b>	–	–
TST3	<b>88.50</b>	87.00	85.10	86.00	–	–
TST4	85.75	82.80	82.90	<b>87.70</b>	–	–
TST7	<b>83.25</b>	81.30	78.70	78.90	–	–
TST9	<b>70.75</b>	67.50	63.70	61.40	–	–
TST10	<b>83.50</b>	79.60	81.00	82.30	–	–
Musk-1	86.67	<b>87.40</b>	77.90	84.40	84.10	86.40
Musk-2	79.00	83.60	84.30	<b>90.50</b>	85.20	88.00

- MIC<sup>bundle</sup> [8]: it is an instance-space approach, solving a nonsmooth nonconvex optimization problem by means of bundle methods [14–16].
- MI-Kernel [19]: it is a bag-space kernel-based approach, where the similarities between the bags are measured by set kernels.

For each data set, the best value is in bold and the character “–” means that the corresponding result is not available. Looking at Table 4, we observe that our approach is the best on six data sets (Fox, TST1, TST3, TST7, TST9, TST10) out of 12, and it appears quite comparable on Elephant, Tiger and Musk-1.

To better analyze the accuracy and the CPU time results, we have also performed the nonparametric statistical Friedman test [13], used to compare different classifiers (see [11, 18]) and furnished by the Statistics and Machine Learning Toolbox of Matlab. Such test is based on providing, for each data set, any classifier with a rank, computed on the basis of its performance with respect to the other approaches. For example, on Elephant, mi-SVM is the winner and then it has rank 1, MI-SVM has rank 2, mi-KSPSVM has rank 3, while both MICA and MIC<sup>bundle</sup>, which present the same performance, have rank 4. The Friedman test provides in output the so-called *p*-value, which, in case it assumes small values (generally less than or equal to 0.05), suggests to reject the null hypothesis, implying that there is a significant difference among the algorithms.

We have applied the Friedman test to the following comparisons, on the basis of the accuracy and the CPU time results reported in Tables 2 and 4:

- The comparison of mi-SPSVM against mi-KSPSVM in terms of average testing correctness and average CPU time has provided  $p$ -values equal to 0.0005 and 0.5637, respectively.
- The comparison among mi-KSPSVM, mi-SVM, MI-SVM, MICA and MIC<sup>bundle</sup> on the image recognition problems (Elephant, Fox and Tiger) has furnished a  $p$ -value equal to 0.2069.
- The comparison among mi-KSPSVM, mi-SVM, MI-SVM and MICA on the TST problems has given a  $p$ -value equal to 0.0370.
- The comparison among all the algorithms (mi-KSPSVM, mi-SVM, MI-SVM, MICA, MIC<sup>bundle</sup> and MI-Kernel) on the two musk problems has output a  $p$ -value equal to 0.7220.

The above  $p$ -values confirm the following observations:

- The linear and the nonlinear versions of the semiproximal SVM approach are comparable in terms of CPU time, but not in terms of accuracy (in fact mi-KSPSVM always significantly overcomes mi-SPSVM).
- On the image recognition problems all the approaches present a comparable behaviour.
- On the TST problems, there is a significant difference among the classifiers (in fact mi-KSPSVM wins on 5 data sets out of 7).
- All the classifiers are comparable on the two musk data sets.

We conclude the section by highlighting that the numerical results presented in [7] have already shown that the main advantage of the semiproximal technique resides in the computational efficiency of the approach, without sacrificing the accuracy performance. Here we have shown that the kernelized version improves significantly the average testing correctness with respect to the linear one, maintaining the same computational advantage in terms of CPU time.

## 5 Conclusions

In this paper we have presented the kernelized version of the semiproximal SVM technique proposed in [7]. The numerical results, performed on a set of benchmark problems and supported by the statistical Friedman test, have shown that embedding the kernel trick into the semiproximal framework greatly improves the accuracy of the classifier, still preserving the good performance in term of computational effort.

Future research could be devoted to extend the kernel trick to other types of MIL classifiers, such as to the spherical separation approach adopted in [3], as done in [6] for the supervised case.

**Funding** Open access funding provided by Università della Calabria within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Amores, J.: Multiple instance classification: review, taxonomy and comparative study. *Artif. Intell.* **201**, 81–105 (2013)
2. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Becker, S., Thrun, S., Obermayer, K. (eds.) *Advances in Neural Information Processing Systems*, pp. 561–568. MIT Press, Cambridge (2003)
3. Astorino, A., Avolio, M., Fuduli, A.: A maximum-margin multisphere approach for binary multiple instance learning. *Eur. J. Oper. Res.* **299**, 642–652 (2022)
4. Astorino, A., Fuduli, A.: The proximal trajectory algorithm in SVM cross validation. *IEEE Trans. Neural Netw. Learn. Syst.* **27**, 966–977 (2016)
5. Astorino, A., Fuduli, A., Gaudioso, M.: A Lagrangian relaxation approach for binary multiple instance classification. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 2662–2671 (2019)
6. Astorino, A., Gaudioso, M.: The fixed-center spherical separation algorithm with kernel transformations for classification problems. *CMS* **6**, 357–372 (2009)
7. Avolio, M., Fuduli, A.: A semiproximal support vector machine approach for binary multiple instance learning. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 3566–3577 (2021)
8. Bergeron, C., Moore, G., Zaretzki, J., Breneman, C., Bennett, K.: Fast bundle algorithm for multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**, 1068–1079 (2012)
9. Carbonneau, M., Cheplygina, V., Granger, E., Gagnon, G.: Multiple instance learning: a survey of problem characteristics and applications. *Pattern Recogn.* **77**, 329–353 (2018)
10. Cristianini, N., Shawe-Taylor, J.: *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, UK (2000)
11. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006)
12. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **89**, 31–71 (1997)
13. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **32**, 675–701 (1937)
14. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optim. Method. Softw.* **19**, 89–102 (2004)
15. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *SIAM J. Optim.* **14**, 743–756 (2004)
16. Fuduli, A., Gaudioso, M., Nurminski, E.: A splitting bundle approach for non-smooth non-convex minimization. *Optimization* **64**, 1131–1151 (2015)
17. Fung, G., Mangasarian, O.: Proximal support vector machine classifiers. In: Provost, F., Srikant, R. (eds.) *Proceedings KDD-2001: Knowledge discovery and data mining*, ACM, pp. 77–86

18. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *J. Mach. Learn. Res.* **9**, 2677–2694 (2008)
19. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: *Proceedings of the Nineteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., pp. 179–186 (2002)
20. Herrera, F., Ventura, S., Bello, R., Cornelis, C., Zafra, A., Sánchez-Tarragó, D., Vluymans, S.: *Multiple instance learning: foundations and algorithms*. Springer International Publishing, USA (2016)
21. Mangasarian, O., Wild, E.: Multiple instance classification via successive linear programming. *J. Optim. Theory Appl.* **137**, 555–568 (2008)
22. Quéllec, G., Cazuguel, G., Cochener, B., Lamard, M.: Multiple-instance learning for medical image and video analysis. *IEEE Rev. Biomed. Eng.* **10**, 213–234 (2017)
23. Shawe-Taylor, J., Cristianini, N.: *Kernel methods for pattern analysis*. Cambridge University Press, UK (2004)
24. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* **109**, 475–494 (2001)
25. Vapnik, V.: *The nature of the statistical learning theory*. Springer Verlag, New York (1995)
26. 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*, pp. 1119–1126
27. Wen, C., Zhou, M., Li, Z.: Multiple instance learning via bag space construction and ELM, in: *Proceedings of SPIE - The International Society for Optical Engineering*, volume 10836

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.