Knowledge and Information Systems (2024) 66:307–338
https://doi.org/10.1007/s10115-023-01961-z

REGULAR PAPER

A linear primal–dual multi-instance SVM for big data classifications

Lodewijk Brand[1] · Hoon Seo[1] · Lauren Zoe Baker[1] · Carla Ellefsen[1] · Jackson Sargent[2] · Hua Wang[1]

Received: 29 January 2022 / Revised: 4 August 2023 / Accepted: 5 August 2023 /
Published online: 26 August 2023

Abstract
Multi-instance learning (MIL) handles data that is organized into sets of instances known as bags. Traditionally, MIL is used in the supervised-learning setting for classifying bags which contain any number of instances. However, many traditional MIL algorithms do not scale efficiently to large datasets. In this paper, we present a novel primal–dual multi-instance support vector machine that can operate efficiently on large-scale data. Our method relies on an algorithm derived using a multi-block variation of the alternating direction method of multipliers. The approach presented in this work is able to scale to large-scale data since it avoids iteratively solving quadratic programming problems which are broadly used to optimize MIL algorithms based on SVMs. In addition, we improve our derivation to include an additional optimization designed to avoid solving a least-squares problem in our algorithm, which increases the utility of our approach to handle a large number of features as well as bags. Finally, we derive a kernel extension of our approach to learn nonlinear decision boundaries for enhanced classification capabilities. We apply our approach to both synthetic and real-world multi-instance datasets to illustrate the scalability, promising predictive performance, and interpretability of our proposed method.

Keywords Multi-instance learning · Support vector machine · Alternating direction method of multipliers · Scalability

1 Introduction

Multi-instance learning (MIL) is a sub-area of machine learning in which training and testing data are organized in sets called *bags*. What makes MIL challenging is that labels associated with these data are frequently provided at the bag level, but not at instance level. This is also known as *weakly supervised* learning in the literature. Algorithms that adhere to this type

✉ Hua Wang
huawangcs@gmail.com

1 Department of Computer Science, Colorado School of Mines, Golden, CO, USA

2 Computer Science and Engineering Division, University of Michigan, Ann Arbor, MI, USA

of weakly supervised learning paradigm are naturally suited to a wide variety of real-world problems that contain limited labeled data. For example, images can be represented by a bag of patches, documents can be organized into sentences or paragraphs, and patients can be represented by a collection of medical records, to name a few. Because the labels are given at the bag level, a lot of resources spent on characterizing each instance are saved. For example, the clinicians only need to label/diagnose the bag or patient, but not each medical record. However, as illustrated in Fig. 1, since each bag can have an arbitrary number of instances, standard machine learning approaches that rely on fixed-length vector representations cannot be applied to such data directly. In this case, the multi-instance learning would be a better choice compared to single-instance learning, because in single-instance learning, predicting becomes difficult when the given instance does not contain any useful information for the prediction. Meanwhile, multi-instance learning enables correct classification from the key instances included in the bag. As a result, significant research efforts have been made to design algorithms that can handle this type of data in recent years.

1.1 Related works

In the past twenty years, a large number of MIL algorithms [1–8] have been proposed. These approaches have been applied to many different topics including drug activity prediction [9], content-based image and video retrieval [10, 11], medical image analysis [12], and document classification [13], among many other application areas [14]. Recently, deep learning-based MIL methods [15–17] have also been proposed to handle multi-instance data. While these methods have demonstrated their effectiveness in solving a variety of real-world problems, their limitations have also been discussed [18, 19]. For example, a recent survey paper [19] notes that current state-of-the-art MIL approaches are sensitive to the construction of instances within a bag. Specifically, they determine that the performance of MIL methods are sensitive to *witness rate*, *e.g*., the proportion of positive instances in positive bags, as well as whether the algorithm operates on the instance or bag level. This has also been observed in older MIL survey papers [18] and requires new algorithms to be tested on a range of different datasets and applications. In addition to dataset-specific performance, the authors of these survey papers highlight that performance improvements in the training time of MIL algorithms, especially those who rely on instance-level information, are necessary for further adoption.

Recently, the scalability for analyzing large amount of data has become another major issue of MIL studies with the development of data mining technologies. The large amount of bags and instances are involved in MIL problems, and MIL models require many parameters to analyze the complex patterns between instances. However, many existing MIL models are often tested on small or moderately sized dataset. To alleviate this scalability issue, We *et.al.,* [20] proposes to map the raw representation of a bag into simpler representation of a vector format which can be classified by the followed SVM model. Although effective, this method does not improve the scalability of the followed SVM model. Vatsavai [21] employs the divide-and-conquer strategy to scatter the large images into patches of predefined size which can be parallelized and processed by Citation-KNN MIL algorithms [22]. However, Citation-KNN MIL algorithm adapted in [21] scales quadratically with respect to the average number of instances per bag.

Besides the quantitative evaluation of MIL models, the qualitative evaluation is also gaining interest. For example, when multiple patches of a histopathological image are formulated as instances, the key instances identified as patches exhibiting the evidence of the

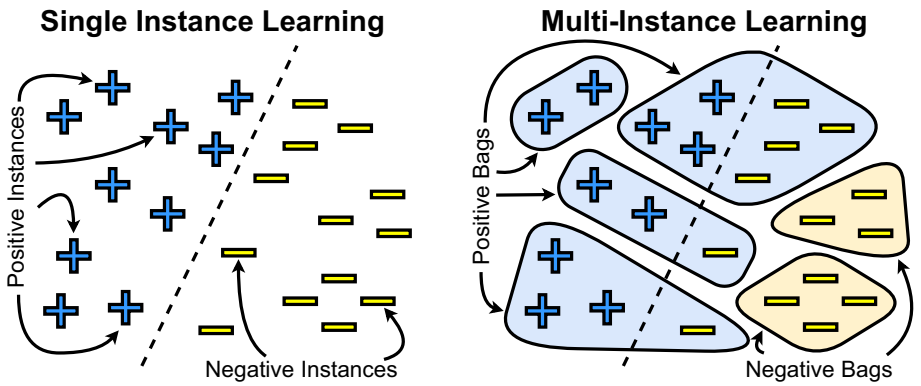


Fig. 1 An illustrative comparison of the single-instance and multi-instance learning paradigms. Algorithms that operate on multi-instance data must contend with the fact that instances are rarely individually labeled. Instead, labels are generally provided at the *bag* level. Thus, the goal of a multi-instance learning algorithm is to learn to identify instances, within a given bag, that indicate a particular class membership

disease would be useful for the doctors who need a reference for the corresponding medical decision [23]. At the same time, the aforementioned identified key instances can provide credibility to the predictions of MIL models. In an effort to interpret the outputs of MIL models, mi-SVM [23] provides pixel-wise abnormality scores of an X-ray image and these scores are plotted over the image constructing the heatmap to identify the regions exhibiting abnormalities. Another multi-instance learning framework [24] divides each instance into patches and calculates their similarities with respect to the positive and negative prototypical parts. As a result, the method in [24] explains which patches in an instance are responsible for the prediction. However the interpretation often requires the ground truth explanation or additional processes to generate the explanations, which sacrifices the scalability.

In this work, we focus specifically on scaling SVM-based MIL algorithms, as they have shown consistent performance and can be further extended to nonlinear decision boundaries via kernelization. Popular SVM-based MIL approaches such as miSVM/MISVM [1], NSK [25], and sMIL/sbMIL [2] have been proposed to handle multi-instance data and have demonstrated promising performance, even when compared against modern MIL deep-learning architectures such as miNet/MINet [26]. While these approaches have performed well and can be extended to solve a variety of real-world problems, they are not widely used in practice as they do not scale well to large datasets. Furthermore, many of these approaches are not equipped with capabilities to interpret the results of their predictions. These two shortcomings, *speed of model training* and *model interpretability* of multi-instance learning methods, are the focus of this work.

1.2 Our contributions

For the remainder of this manuscript, we present a novel method that extends a multi-instance SVM to large-scale data. Our approach uses the multi-block alternating direction method of multipliers (ADMM) to avoid iteratively solving the quadratic programming problems that arise from standard SVM-based MIL approaches. The scientific contributions of this work are as follows:

- A novel MIL algorithm derivation, named the *primal–dual multi-instance SVM* (pdMISVM) method, and an associated implementation that scales ***linearly*** as the number of bags increases.
- An inexact variation of our approach, based on the optimal line search method, that scales ***linearly*** as the number of features increases.
- Experimental results showcasing the promising predictive performance, scalability, and interpretability of our approach on baseline multi-instance data and real-world image data compared against other MIL algorithms.
- An extension of our approach that allows for the inclusion of an arbitrary kernel function and a proof-of-concept experiment on synthetic data verifying our derivation.

This paper is an extension of our recent work [27] originally reported in the Proceedings of the Twenty-First International Conference on Data Mining (ICDM 2021). In this extended journal manuscript, we provide the following expansions over its conference version:

- We expand our discussions on the previous works related with the scalability and interpretability of MIL problems (Sect. 1.1).
- We present the complete derivations of the kernel variation of our pdMISVM method that scales linearly against the number of bags.
  - While the scalability issue of the kernel version of our new method against the number of bags has been discussed in the conference version, it was not solved. In this journal extension, we systematically derive the objective of the kernel version of our pdMISVM method and its solution algorithms. (Sect. 2.5).
  - The experimental results of our two kernel variations are added to compare their performance and scalability (Sects. 3.2 and 3.3).
- We provide the detailed analyses on the computational complexities of our pdMISVM method and its kernel extension (Sects. 2.4 and 2.5). The codes to implement our method and the data used in our methods have been made publicly available online at: https://github.com/minds-mines/pdMISVM.jl.
- We expand our experimental evaluations with two additional benchmark datasets and two recently proposed attention-based deep learning models (Sect. 3.2).
- We include a case study on neuroimaging data to further evaluate the performance and interpretability of our method to solve real-world problems. We apply our method to identify the disease relevant brain regions from the neuroimaging perspective (Sect. 3.6).

2 Methods

In this section, we begin with a sketch for the steps required for the multi-instance SVM (MISVM) derivation initially presented by Andrews et.al., [1]. Then, following the multi-block ADMM framework [28–30], we construct the augmented Lagrangian that will be used to derive the solution to the proposed pdMISVM method, which is then followed by a step-by-step derivation to optimize the proposed objective. In addition, we extend our approach to handle a large number of features through an application of the optimal line search method [31]. Finally, we derive the solution algorithm of kernel pdMISVM.

2.1 Notation

In this manuscript we represent matrices as $\mathbf{M}$, vectors as $\mathbf{m}$, and scalars as m. The i-th row and j-th column of $\mathbf{M}$ are denoted as $\mathbf{m}^i$ and $\mathbf{m}_j$, respectively. Similarly, m^i_j is the scalar value indexed by the i-th row and j-th column of $\mathbf{M}$. The matrix $\mathbf{M}_p$ corresponds to the p-th column-block of $\mathbf{M}$. Given a $K \times N$ matrix $\mathbf{M}$, $\{m, i\} = \arg\max_{m',i'}(\mathbf{M})$ gives the row-by-column coordinates for the maximum element in $\mathbf{M}$. The row and column indices are given by $\arg\max_{m',i'}(\mathbf{M})^m$ and $\arg\max_{m',i'}(\mathbf{M})_i$, respectively.

2.2 Extending the MISVM to *K*-class classification

In the binary multi-instance classification problem, the MIL algorithm is presented with a collection of bags and labels represented by the set $\{\mathbf{X}_i, y_i\}_{i=1}^N$, where $y_i \in \{-1, 1\}$ and $\mathbf{X}_i \in \mathbb{R}^{d\times n_i}$ designates a bag containing n_i instances, and $\{\mathbf{x}_1, \ldots, \mathbf{x}_{n_i}\} \in \mathbf{X}_i$ represent each instance within the i-th bag. Following the *instance-centric* approach advocated by Andrews *et.al.,* [1] for the MISVM model, where a single "witness" instance determines the class of a bag, we define the decision function for a multi-instance binary classifier as

$$y_i = \text{sign}\left(\max_{\mathbf{x}_i\in\mathbf{X}_i}(\mathbf{w}^T\mathbf{x}_i + b)\right), \tag{1}$$

where $\mathbf{w}$ and b are the hyperplane and intercept for the MISVM model. The MISVM objective devised by Andrews *et.al.,* is [1]

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^N \xi_i \\ \text{subject to} \quad & \max_{\mathbf{x}_i\in\mathbf{X}_i}\left(\mathbf{w}^T\mathbf{x}_i + b\right) y_i \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \ldots, N, \end{aligned} \tag{2}$$

where C is a hyperparameter that determines the level of regularization on the learned hyperplane, and ξ_i are slack variables. The constraints in Eq. (2) can be incorporated into the objective via a Lagrangian function

$$\min_{\mathbf{w},b}\ \max_{\boldsymbol{\alpha}}\quad \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) \qquad \text{subject to} \qquad \alpha_i \geq 0, \tag{3}$$

which can be solved with respect to the dual variables (α_i) using off-the-shelf quadratic programming solvers or heuristic algorithms like sequential minimal optimization [32] that takes advantage of a limited number of support vectors. Although the MISVM formulation proposed by Andrews et.al., [1] is widely used in MIL literature, it is generally limited to binary classification problems.

In order to design a method suitable for multi-class multi-instance classification, we extend the decision function presented in Eq. (1) to K-classes via

$$\hat{y}_i = \arg\max_{m',i'}\left(\mathbf{W}^T\mathbf{X}_i + \mathbf{b}^T\mathbf{1}_i\right)^m, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{d\times K}$, $\mathbf{b} \in \mathbb{R}^K$, $\hat{y}_i \in \{1, \ldots, K\}$ represents the hyperplanes, intercepts, and labels for K classes. Motivated by the results of [33] where it is argued that all-in-one formulations for K-class SVMs provide superior predictive performance, when compared to

one-vs-all approaches, we construct the following Weston & Watkins [34] MISVM extension to Eq. (2)

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\boldsymbol{\xi}} \ & \frac{1}{2}\sum_{m=1}^{K}\|\mathbf{w}_m\|_2^2 + C\sum_{i=1}^{N}\sum_{m=1}^{K}\xi_i^m \\ \text{subject to} \ & \left(1 - \left[\max(\mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m) - \max(\mathbf{w}_y^T\mathbf{X}_i + \mathbf{1}b_y)\right]y_i^m\right)_+ \leq \xi_i^m, \\ & 0 \leq \xi_i^m, \ i = 1,\ldots,2, \ m = i,\ldots,K, \end{aligned} \tag{5}$$

where $\mathbf{w}_y$, b_y is the hyperplane-intercept pair associated with the i-th bag's class label, $(\cdot)_+ = \max(0, \cdot)$ is the hinge loss function, and $y_i^m \in \{-1, 1\}$ indicates if the i-th bag belongs to the m-th class. Similar to Eq. (2), the K-class formulation above can be transformed into a quadratic programming problem and solved, although this approach is known [33] not to scale well as the number of bags increases. To address this issue, we propose a novel primal–dual algorithm based on the multi-block ADMM [30] to optimize Eq. (5).

2.3 A primal–dual multi-instance SVM

Incorporating the constraints of Eq. (5) into the objective gives the unconstrained optimization

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b}} \ & \frac{1}{2}\sum_{m=1}^{K}\|\mathbf{w}_m\|_2^2 + C\sum_{i=1}^{N}\sum_{m=1}^{K}(1 - [\max(\mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m) \\ & - \max(\mathbf{w}_y^T\mathbf{X}_i + \mathbf{1}b_y)]y_i^m)_+, \end{aligned} \tag{6}$$

which is difficult to solve given the coupling across $\mathbf{w}_m$, b_m, and the $\max(\cdot)$ operations. Using the multi-block ADMM approach, we introduce the following constraints, inspired by [31, 35], and rewrite Eq. (6) as

$$\begin{aligned} \min_{\substack{\mathbf{W},\mathbf{b},\mathbf{E},\\ \mathbf{Q},\mathbf{R},\mathbf{T},\mathbf{U}}} \ & \frac{1}{2}\sum_{m=1}^{K}\|\mathbf{w}_m\|_2^2 + C\sum_{i=1}^{N}\sum_{m=1}^{K}\left(y_i^m e_i^m\right)_+ \\ \text{subject to} \ & e_i^m = y_i^m - q_i^m + r_i^m, \ q_i^m = \max\left(\mathbf{t}_i^m\right), \ \mathbf{t}_i^m = \mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m, \\ & r_i^m = \max\left(\mathbf{u}_i^m\right), \ \mathbf{u}_i^m = \mathbf{w}_y^T\mathbf{X}_i + \mathbf{1}b_y, \end{aligned} \tag{7}$$

to decouple the primal variables. Then, the augmented Lagrangian function of Eq. (7) is

$$\begin{aligned} \mathcal{L}_\mu = & \frac{1}{2}\sum_{m=1}^{K}\|\mathbf{w}_m\|_2^2 + C\sum_{i=1}^{N}\sum_{m=1}^{K}\left(y_i^m e_i^m\right)_+ \\ & + \frac{\mu}{2}\sum_{i=1}^{N}\sum_{m=1}^{K}\Bigg[\left(e_i^m - (y_i^m - q_i^m + r_i^m - \lambda_i^m/\mu)\right)^2 \\ & + \left(q_i^m - \max\left(\mathbf{t}_i^m\right) + \sigma_i^m/\mu\right)^2 + \left\|\mathbf{t}_i^m - \left(\mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m\right) + \boldsymbol{\theta}_i^m/\mu\right\|_2^2 \\ & + \left(r_i^m - \max\left(\mathbf{u}_i^m\right) + \omega_i^m/\mu\right)^2 + \left\|\mathbf{u}_i^m - \left(\mathbf{w}_y^T\mathbf{X}_i + \mathbf{1}b_y\right) + \boldsymbol{\xi}_i^m/\mu\right\|_2^2\Bigg], \end{aligned} \tag{8}$$

where $\mathbf{W}$, $\mathbf{b}$, $\mathbf{E}$, $\mathbf{Q}$, $\mathbf{T}$, $\mathbf{R}$, $\mathbf{U}$ are the primal variables, $\boldsymbol{\Lambda}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Theta}$, $\boldsymbol{\Omega}$, $\boldsymbol{\Xi}$ are the dual variables, and $\mu > 0$ is a tuning parameter. Equation (8) is then differentiated with respect to each

primal variable to derive Algorithm 2.1. The primal–dual updates terminate when the total difference between the constraints incorporated via the augmented Lagrangian terms are less than a predefined tolerance. In the following, we provide the details to derive each step of Algorithm 2.1.

Algorithm 2.1 The pdMISVM method to optimize Eq. (8)

Data: $\mathbf{X} \in \mathbb{R}^{D\times(n_1+\cdots+n_N)}$ and $\mathbf{Y} \in \{-1, 1\}^{K\times N}$.
Hyperparameters: $C > 0$, $\mu > 0$, $\rho > 1$ and $tolerance > 0$.
Initialize: primal variables $\mathbf{W}$, $\mathbf{b}$, $\mathbf{E}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{T}$, $\mathbf{U}$ and dual variables $\mathbf{\Lambda}$, $\mathbf{\Sigma}$, $\mathbf{\Theta}$, $\mathbf{\Omega}$, $\mathbf{\Xi}$.
while residual $> tolerance$ **do**
  for $m \in K$ **do**
    Update $\mathbf{w}_m \in \mathbf{W}$ **by** Eq. (10), or Eq. (25) (inexact)
    Update $b_m \in \mathbf{b}$ **by** Eq. (11)
  end for
  for $(p, m) \in \{N, K\}$ **do**
    Update $e_p^m \in \mathbf{E}$ **by** Eq. (13)
    Update $q_p^m \in \mathbf{Q}$ **by** Eq. (15)
    Update $r_p^m \in \mathbf{R}$ **by** Eq. (16)
    for $j \in n_p$ **do**
      Update $t_{p,j}^m \in \mathbf{T}$ **by** Eq. (19)
      Update $u_{p,j}^m \in \mathbf{U}$ **by** Eq. (20)
    end for
    Update $\lambda_p^m, \sigma_p^m, \omega_p^m, \boldsymbol{\theta}_p^m, \boldsymbol{\xi}_p^m$ **by** $\lambda_i^m = \lambda_i^m + \mu(e_i^m - (y_i^m - q_i^m + r_i^m))$;
    $\sigma_i^m = \sigma_i^m + \mu(q_i^m - \max(\mathbf{t}_i^m))$; $\omega_i^m = \omega_i^m + \mu(r_i^m - \max(\mathbf{u}_i^m))$;
    $\boldsymbol{\theta}_i^m = \boldsymbol{\theta}_i^m + \mu(\mathbf{t}_i^m - (\mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m))$; $\boldsymbol{\xi}_i^m = \boldsymbol{\xi}_i^m + \mu(\mathbf{u}_i^m - (\mathbf{w}_y^T\mathbf{X}_i + \mathbf{1}b_y))$.
  end for
  Update $\mu = \rho\mu$
end while
return $(\mathbf{w}_m, \ldots, \mathbf{w}_K) \in \mathbf{W}$ and $(b_1, \ldots, b_K) \in \mathbf{b}$.

Update W & b, exact. Removing all terms from Eq. (8) that do not include $\mathbf{W}$ and decoupling across columns of $\mathbf{W}$ gives the following K subproblems

$$\begin{aligned}\mathbf{w}_m = \arg\min_{\mathbf{w}_m} \frac{1}{2}\|\mathbf{w}_m\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{N}\left[\left\|\mathbf{t}_i^m - \left(\mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m\right) + \boldsymbol{\theta}_i^m/\mu\right\|_2^2\right] \\ + \sum_{i'=1}^{N'}\sum_{m'=1}^{K}\left[\frac{\mu}{2}\left\|\mathbf{u}_{i'}^{m'} - \left(\mathbf{w}_m^T\mathbf{X}_{i'} + \mathbf{1}b_m\right) + \boldsymbol{\xi}_{i'}^{m'}/\mu\right\|_2^2\right],\end{aligned} \tag{9}$$

where i' indicates the column blocks in $\mathbf{X}$ (and the corresponding columns of $\mathbf{U}$ and $\mathbf{\Xi}$) that belong to the m-th class and N' is the total number of bags belonging to the m-th class. Taking the derivative of Eq. (9) with respect to $\mathbf{w}_m$ and setting the result equal to zero gives the closed form solution

$$\begin{aligned}\mathbf{w}_m^T = \Big(\textstyle\sum_{i=1}^{N}\left[\left(\mathbf{t}_i^m - \mathbf{1}b_m + \boldsymbol{\theta}_i^m/\mu\right)\mathbf{X}_i^T\right] + \sum_{i'=1}^{N'}\sum_{m'=1}^{K}\left[\left(\mathbf{u}_{i'}^{m'} - \mathbf{1}b_m\right.\right. \\ \left.\left. + \boldsymbol{\xi}_{i'}^{m'}/\mu\right)\mathbf{X}_{i'}^T\right]\Big) * \left(\mathbf{I}/\mu + \textstyle\sum_{i=1}^{N}\mathbf{X}_i\mathbf{X}_i^T + K\sum_{i'=1}^{N'}\mathbf{X}_{i'}\mathbf{X}_{i'}^T\right)^{-1},\end{aligned} \tag{10}$$

which can be calculated via a least-squares solver to avoid an inverse calculation.

Similarly, differentiating Eq. (9) element-wise with respect to b_m and setting the result equal to zero gives the update

$$b_m = \frac{\sum_{i=1}^{N}\left[\mathbf{t}_i^m - \mathbf{w}_m^T\mathbf{X}_i + \boldsymbol{\theta}_i^m/\mu\right] + \sum_{i'=1}^{N'}\sum_{m'=1}^{K}\left[\mathbf{u}_{i'}^{m'} - \mathbf{w}_m^T\mathbf{X}_{i'} + \boldsymbol{\xi}_{i'}^{m'}/\mu\right]}{N + KN'}. \tag{11}$$

Update E. Dropping terms that do not contain **E** from Eq. (8), by performing element-wise decoupling of the problem, we end up with the following $K \times N$ subproblems

$$e_i^m = \arg\min_{e_i^m} C\left(y_i^m e_i^m\right)_+ + \frac{\mu}{2}\left(e_i^m - n_i^m\right)^2, \tag{12}$$

where $n_i^m = y_i^m - q_i^m + r_i^m - \frac{\lambda_i^m}{\mu}$. Equation (12) can be differentiated with respect to e_i^m via the sub-gradient method, and solved by the following three cases

$$e_i^m = \begin{cases} n_i^m - \frac{C}{\mu}y_i^m & \text{when} \quad y_i^m n_i^m > \frac{C}{\mu}, \\ 0 & \text{when} \quad 0 \le y_i^m n_i^m \le \frac{C}{\mu}, \\ n_i^m & \text{when} \quad y_i^m n_i^m < 0. \end{cases} \tag{13}$$

Update Q & **R**. Keeping only terms with **Q** in Eq. (8) and performing element-wise decoupling, we end up with the following $K \times N$ subproblems

$$\begin{aligned} q_i^m = \arg\min_{q_i^m} & \left(e_i^m - y_i^m + q_i^m - r_i^m + \lambda_i^m/\mu\right)^2 \\ & + \left(q_i^m - \max\left(\mathbf{t}_i^m\right) + \sigma_i^m/\mu\right)^2. \end{aligned} \tag{14}$$

Taking the derivative of Eq. (14) with respect to q_i^m and setting the result equal to zero, we can the problem for q_i^m by the following update

$$q_i^m = \frac{\left(y_i^m - e_i^m + r_i^m - \lambda_i^m/\mu + \max\left(\mathbf{t}_i^m\right) - \sigma_i^m/\mu\right)}{2}. \tag{15}$$

Following the same steps for each $r_i^m \in \mathbf{R}$, we derive the element-wise updates

$$r_i^m = \frac{\left(e_i^m - y_i^m + q_i^m + \lambda_i^m/\mu + \max\left(\mathbf{u}_i^m\right) - \omega_i^m/\mu\right)}{2}. \tag{16}$$

Update T & **U**. Keeping terms in Eq. (8) containing **T** and decoupling across K and N, we end up with the following subproblem

$$\mathbf{t}_i^m = \arg\min_{\mathbf{t}_i^m} \left(q_i^m - \max\left(\mathbf{t}_i^m\right) + \sigma_i^m/\mu\right)^2 + \left\|\mathbf{t}_i^m - \left(\mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m\right) + \theta_i^m/\mu\right\|_2^2, \tag{17}$$

which can be further decoupled into element-wise subproblems for each $t_{i,j}^m \in \mathbf{t}_i^m$, giving $K \times (n_1 + \cdots + n_N)$ problems

$$t_{i,j}^m = \arg\min_{t_{i,j}^m} \begin{cases} \left(q_i^m - t_{i,j}^m + \sigma_i^m/\mu\right)^2 + \left(t_{i,j}^m - \phi_{i,j}^m\right)^2, \\ \text{when } t_{i,j}^m = \max\left(\mathbf{t}_i^m\right), \\ \left(t_{i,j}^m - \phi_{i,j}^m\right)^2 \text{ else}, \end{cases} \tag{18}$$

where $\boldsymbol{\phi}_i^m = \mathbf{w}_m^T\mathbf{X}_i + \mathbf{1}b_m - \boldsymbol{\theta}_i^m/\mu$.

Taking the derivative of Eq. (18) with respect to $t_{i,j}^m$ and setting the result equal to zero, we solve the problem for $t_{i,j}^m$ by the following updates

$$t_{i,j}^m = \begin{cases} \frac{\max(\boldsymbol{\phi}_i^m)+q_i^m+\sigma_i^m/\mu}{2} & \text{if } j = \arg\max(\boldsymbol{\phi}_i^m), \\ \phi_{i,j}^m & \text{else.} \end{cases} \tag{19}$$

This same strategy is applied to derive the element-wise updates of $\mathbf{U}$, which gives

$$u_{i,j}^m = \begin{cases} \frac{\max(\boldsymbol{\psi}_i^m)+r_i^m+\omega_i^m/\mu}{2} & \text{if } j = \arg\max(\boldsymbol{\psi}_i^m), \\ \psi_{i,j}^m & \text{else.} \end{cases} \tag{20}$$

where $\boldsymbol{\psi}_i^m = \mathbf{w}_y^T \mathbf{X}_i + \mathbf{1}b_y - \boldsymbol{\xi}_i^m/\mu$.

The associated dual variable updates are provided in Algorithm 2.1.

2.4 Scaling to a large number of features

Although the updates derived in Sect. 2.3 provide a suitable algorithm as the number of bags increase, it does not scale well against the increasing number of features. To be specific, the calculation in Eq. (10) for the left parenthesis $\left(\sum_{i=1}^N \left[\left(\mathbf{t}_i^m - \mathbf{1}b_m + \boldsymbol{\theta}_i^m/\mu\right)\mathbf{X}_i^T\right] + \sum_{i'=1}^{N'}\sum_{m'=1}^{K}\left[(\mathbf{u}_{i'}^{m'} - \mathbf{1}b_m + \boldsymbol{\xi}_{i'}^{m'}/\mu)\mathbf{X}_{i'}^T\right]\right)$ has the computational complexity $O\left(\left(n_1 + \cdots + n_N\right)\cdot d\right)$ and the right parenthesis $(\mathbf{I}/\mu + \sum_{i=1}^N \mathbf{X}_i\mathbf{X}_i^T + K\sum_{i'=1}^{N'}\mathbf{X}_{i'}\mathbf{X}_{i'}^T)^{-1}$ can be efficiently calculated via the least-squares solver which has complexity $O\left(d^2\right)$. As a result, the updating $\mathbf{w}_m$ requires the time complexity $O\left((n_1 + \cdots + n_N + d)\cdot d\right)$ which scales quadratically as the number of features increase; this limits the scalability of our approach to *bags only*. Additionally, since μ is updated every iteration, the least-squares solver must be invoked at every iteration although $(\sum_{i=1}^N \mathbf{X}_i\mathbf{X}_i^T + K\sum_{i'=1}^{N'}\mathbf{X}_{i'}\mathbf{X}_{i'}^T)$ can be precomputed at the beginning of algorithm. The inversion calculation for $d \times d$ matrix is the bottleneck of SVM algorithms. To handle this issue, we propose an alternative optimal line search method [31] to update $\mathbf{W}$ for avoiding the inverse matrix calculation.

Update W, inexact. The partial derivative of Eq. (9) with respect to $\mathbf{w}_k$ gives

$$\begin{aligned} \nabla_{\mathbf{w}_m}^T = \mathbf{w}_m^T &- \mu\sum_{i=1}^{N}[\mathbf{t}_i^m - \mathbf{w}_m^T\mathbf{X}_i - \mathbf{1}b_m + \boldsymbol{\theta}_i^m/\mu]\mathbf{X}_i^T \\ &- \mu\sum_{i'=1}^{N'}\sum_{m'=1}^{K}[\mathbf{u}_{i'}^{m'} - \mathbf{w}_m^T\mathbf{X}_{i'} - \mathbf{1}b_m + \boldsymbol{\xi}_{i'}^{m'}/\mu]\mathbf{X}_{i'}^T, \end{aligned} \tag{21}$$

which can be used to create the following minimization problem

$$\begin{aligned} s_m = \arg\min_{s_m} \frac{1}{2}\left\|\mathbf{w}_m^T - s_m\nabla_{\mathbf{w}_m}^T\right\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{N}\Big[\Big\|\mathbf{t}_i^m - (\mathbf{w}_m^T - s_m\nabla_{\mathbf{w}_m}^T)\mathbf{X}_i - \mathbf{1}b_m \\ + \theta_i^m/\mu\Big\|_2^2\Big] + \sum_{i'=1}^{N'}\sum_{m'=1}^{K}\left[\frac{\mu}{2}\left\|\mathbf{u}_{i'}^{m'} - (\mathbf{w}_m^T - s_m\nabla_{\mathbf{w}_m}^T)\mathbf{X}_{i'} - \mathbf{1}b_m + \xi_{i'}^{m'}/\mu\right\|_2^2\right], \end{aligned} \tag{22}$$

in terms of s_m instead of $\mathbf{w}_m$.

Differentiating Eq. (22) with respect to s_m and setting the result equal to zero, we solve the problem for s_m as follows

$$s_m = \frac{\left(\mathbf{w}_m^T - \mu \sum_{i=1}^{N} \hat{\mathbf{t}}_i^m \mathbf{X}_i^T - \mu \sum_{i'=1}^{N'} \sum_{m'=1}^{K} \hat{\mathbf{u}}_{i'}^{m'} \mathbf{X}_{i'}^T\right) \nabla_{\mathbf{w}_m}}{\nabla_{\mathbf{w}_m}^T \left(\mathbf{I} + \mu \sum_{i=1}^{N} \mathbf{X}_i \mathbf{X}_i^T + \mu K \sum_{i'=1}^{N'} \mathbf{X}_{i'} \mathbf{X}_{i'}^T\right) \nabla_{\mathbf{w}_m}}, \tag{23}$$

where $\hat{\mathbf{t}}_i^m = \mathbf{t}_i^m - \mathbf{w}_m^T \mathbf{X}_i - \mathbf{1} b_m + \boldsymbol{\theta}_i^m / \mu$ and $\hat{\mathbf{u}}_{i'}^{m'} = \mathbf{u}_{i'}^{m'} - \mathbf{w}_m^T \mathbf{X}_{i'} - \mathbf{1} b_m + \xi_{i'}^{m'} / \mu$.

Because the denominator of Eq. (23) is equivalent to

$$\left\| \nabla_{\mathbf{w}_m} \right\|_2^2 + \mu \sum_{i=1}^{N} \left\| \nabla_{\mathbf{w}_m}^T \mathbf{X}_i \right\|_2^2 + \mu K \sum_{i'=1}^{N'} \left\| \nabla_{\mathbf{w}_m}^T \mathbf{X}_{i'} \right\|_2^2, \tag{24}$$

Equation (23) can be calculated efficiently in $O\left((n_1 + \cdots + n_N) \cdot d\right)$ time.

By combining Eq. (21) and Eq. (23), we can update $\mathbf{w}_m$ via

$$\mathbf{w}_m = \mathbf{w}_m - s_m \nabla_{\mathbf{w}_m}. \tag{25}$$

This "inexact" update option avoids solving the least squares problem present in Eq. (10) and is provided as an option on Line 6 of Algorithm 2.1 to extend our method to handle a large number of features.

2.5 A primal–dual multi-instance SVM with Kernel

While the exact and inexact formulations described in Algorithm 2.1 are computationally efficient and show promising performance on a variety of multi-instance datasets, they are limited to classification problems where instances within bags are linearly separable. In order for enabling our method to learn nonlinear decision boundaries, we derive an kernel extension of pdMISVM method in this subsection.

We begin by replacing all bags $\mathbf{X}_i$ in Eq. (6) by their corresponding feature matrices $\phi(\mathbf{X}_i) = \boldsymbol{\Phi}_i \in \mathbb{R}^{d_\phi \times n_i}$, which gives

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b}} \frac{1}{2} \sum_{m=1}^{K} \|\mathbf{w}_m\|_2^2 + C \sum_{i=1}^{N} \sum_{m=1}^{K} (1 - [\max(\mathbf{w}_m^T \boldsymbol{\Phi}_i + \mathbf{1} b_m) \\ - \max(\mathbf{w}_y^T \boldsymbol{\Phi}_i + \mathbf{1} b_y)] y_i^m) \end{aligned} \tag{26}$$

where ϕ is an arbitrary kernel function. Then, by introducing constraints to decouple $\mathbf{w}_m$ and b_m in Eq. (7) and incorporating them into the objective, we derive the following Lagrangian formulation

$$\begin{aligned} \mathcal{L}_\mu = & \frac{1}{2} \sum_{m=1}^{K} \|\mathbf{w}_m\|_2^2 + \sum_{i=1}^{N} \sum_{m=1}^{K} C \left(y_i^m e_i^m\right)_+ \\ & + \frac{\mu}{2} \sum_{i=1}^{N} \sum_{m=1}^{K} \Bigg[\left(e_i^m - (y_i^m - q_i^m + r_i^m - \lambda_i^m / \mu)\right)^2 \\ & + \left(q_i^m - \max\left(\mathbf{t}_i^m\right) + \sigma_i^m / \mu\right)^2 + \left\| \mathbf{t}_i^m - \left(\mathbf{w}_m^T \boldsymbol{\Phi}_i + \mathbf{1} b_m\right) + \boldsymbol{\theta}_i^m / \mu \right\|_2^2 \\ & + \left(r_i^m - \max\left(\mathbf{u}_i^m\right) + \omega_i^m / \mu\right)^2 + \left\| \mathbf{u}_i^m - \left(\mathbf{w}_y^T \boldsymbol{\Phi}_i\right) + \mathbf{1} b_y \right) + \boldsymbol{\xi}_i^m / \mu \Big\|_2^2 \Bigg]. \end{aligned} \tag{27}$$

We pause here to recognize that, while the number of columns in each $\mathbf{\Phi}_i$ is equal to the number of instances inside the original bag $\mathbf{X}_i$, the number of rows, *i.e.*., d_ϕ, can be arbitrarily (even *infinitely*) large. Motivated by [36], we work to derive each update of our algorithm with respect to each primal variable in Eq. (27) without explicitly calculating $\mathbf{w}_m$.

2.5.1 The Kernel extension of our method with exact solutions

We update $\mathbf{W}$ by discarding all terms in Eq. (27) that do not contain $\mathbf{W}$ and decoupling across columns, which gives the following K problems

$$\begin{aligned}\mathbf{w}_m = \arg\min_{\mathbf{w}_m} \; & \frac{1}{2}\|\mathbf{w}_m\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{N}\left[\left\|\mathbf{t}_i^m - \mathbf{w}_m^T\mathbf{\Phi}_i + \mathbf{1}b_m + \boldsymbol{\theta}_i^m/\mu\right\|_2^2\right] \\ & + \sum_{i'=1}^{N'}\sum_{m'=1}^{K}\left[\frac{\mu}{2}\left\|\mathbf{u}_{i'}^{m'} - \mathbf{w}_m^T\mathbf{\Phi}_{i'} + \mathbf{1}b_m + \boldsymbol{\xi}_{i'}^{m'}/\mu\right\|_2^2\right],\end{aligned} \tag{28}$$

where i' indicates the corresponding column blocks of $\mathbf{\Phi} = [\mathbf{\Phi}_1 \; \ldots \; \mathbf{\Phi}_N] \in \mathbb{R}^{d_\phi \times (n_1+\cdots+n_N)}$ that belong to the m-th class. Equation (28) can be differentiated with respect to $\mathbf{w}_m$, set equal to zero, and solved for each $\mathbf{w}_m$

$$\begin{aligned}\mathbf{w}_m^T = & \left(\left[\left(\mathbf{t}^m - \mathbf{1}b_m + \boldsymbol{\theta}^m/\mu\right)\mathbf{\Phi}^T\right] + \textstyle\sum_{m'=1}^{K}\left[\left(\mathbf{u}_{\prime}^{m'} - \mathbf{1}b_m + \boldsymbol{\xi}_{\prime}^{m'}/\mu\right)\mathbf{\Phi}_{\prime}^T\right]\right) \\ & * \left(\mathbf{I}/\mu + \mathbf{\Phi}\mathbf{\Phi}^T + K\,\mathbf{\Phi}_{\prime}\mathbf{\Phi}_{\prime}^T\right)^{-1},\end{aligned} \tag{29}$$

where $\mathbf{\Phi}_{\prime} = [\mathbf{\Phi}_{1'} \; \ldots \; \mathbf{\Phi}_{N'}]$. Equation (29) can be written in the matrix form

$$\mathbf{w}_m^T = \mathbf{v}^m\mathbf{D}\hat{\mathbf{\Phi}}^T * \left(\mathbf{I}/\mu + \hat{\mathbf{\Phi}}\mathbf{D}\hat{\mathbf{\Phi}}^T\right)^{-1}, \tag{30}$$

where $\mathbf{v}^m = [\mathbf{t}^m - \mathbf{1}b_m + \boldsymbol{\theta}^m/\mu \quad 1/K\sum_{m'=1}^{K}(\mathbf{u}_{\prime}^{m'} - \mathbf{1}b_m + \boldsymbol{\xi}_{\prime}^{m'}/\mu)]$, $\mathbf{D} = [\mathbf{I}\;\mathbf{0};\;\mathbf{0}\;K\mathbf{I}]$ and $\hat{\mathbf{\Phi}} = [\mathbf{\Phi}\;\mathbf{\Phi}_{\prime}]$. Since the kernel function applied to each $\mathbf{X}_i$ may return feature vectors that are infinitely long, it may be impossible to calculate the inverse required to express $\mathbf{w}_m$ in Eq. (30). In order to solve this issue we use the following method introduced in [36]

$$(\mathbf{P}^{-1} + \mathbf{m}^T\mathbf{R}^{-1}\mathbf{m})^{-1}\mathbf{m}^T\mathbf{R}^{-1} = \mathbf{P}\mathbf{m}^T(\mathbf{m}\mathbf{P}\mathbf{m}^T + \mathbf{R})^{-1}$$

to rewrite $\mathbf{w}_m^T$ equivalently, as

$$\mathbf{w}_m^T = \mathbf{v}_m(\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}} + \mathbf{D}^{-1}/\mu)^{-1}\hat{\mathbf{\Phi}}^T. \tag{31}$$

The updated expression in Eq. (31) can then be used to update $\mathbf{w}_m^T\phi(\mathbf{X}_i)$

$$\mathbf{w}_m^T\mathbf{\Phi} = \mathbf{v}_m(\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}} + \mathbf{D}^{-1}/\mu)^{-1}\hat{\mathbf{\Phi}}^T\mathbf{\Phi}, \tag{32}$$

and calculate $\|\mathbf{w}_m\|_2^2 = \mathbf{tr}\left(\mathbf{w}_m^T\mathbf{w}_m\right)$ by

$$\|\mathbf{w}_m\|_2^2 = \mathbf{tr}\left(\mathbf{v}_m(\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}} + \mathbf{D}^{-1}/\mu)^{-1}\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}}(\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}} + \mathbf{D}^{-1}/\mu)^{-1}\mathbf{v}_m^T\right), \tag{33}$$

without directly computing $\mathbf{w}_m$. These two expressions are computationally tractable as the kernel expressions occur as an inner product in both cases. The updates for the other primal variables $\mathbf{b}$, $\mathbf{E}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{U}$ are the same as Algorithm 2.1, except each instance of $\mathbf{w}_m\mathbf{X}_i$ and $\mathbf{w}_y\mathbf{X}_i$ are replaced by the corresponding columns of Eq. (32). We outline the

update steps for the pdMISVM method with kernel in Algorithm 2.2. The time complexity of Algorithm 2.2 $O\left((n_1 + \cdots + n_N)^2\right)$. The complexity comes from the multiplication between $\mathbf{v}_m \in \mathbb{R}^{n_1+\cdots+n_N+n_{1'}+\cdots+n_{N'}}$ and matrix and calculating $(\hat{\mathbf{\Phi}}^T \hat{\mathbf{\Phi}} + \mathbf{D}^{-1}/\mu)^{-1}$ in Eq. (32) and Eq. (33).

Algorithm 2.2 The pdMISVM method with kernel to optimize Eq. (27)

Data: $\mathbf{X} \in \mathbb{R}^{D\times(n_1+\cdots+n_N)}$ and $\mathbf{Y} \in \{-1, 1\}^{K\times N}$.
Hyperparameters: $C > 0$, $\mu > 0$, $\rho > 1$ and $tolerance > 0$, and kernel function ϕ.
Initialize: primal variables $\mathbf{W}^T\mathbf{\Phi}$, $\mathbf{b}$, $\mathbf{E}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{T}$, $\mathbf{U}$, dual variables $\mathbf{\Lambda}$, $\mathbf{\Sigma}$, $\mathbf{\Theta}$, $\mathbf{\Omega}$, $\mathbf{\Xi}$ and calculate $\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}}$, $\hat{\mathbf{\Phi}}^T\mathbf{\Phi}$ for each class using ϕ.
while residual $> tolerance$ **do**
  for $m \in K$ **do**
    Update $\mathbf{w}_m^T\mathbf{\Phi} \in \mathbf{W}^T\mathbf{\Phi}$ **by** Eq. (32), or $\mathbf{w}_m \in \mathbf{W}$ by Eq. (34) (inexact).
  end for
  Update $\mathbf{b}$, $\mathbf{E}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{T}$, $\mathbf{U}$, $\mathbf{\Lambda}$, $\mathbf{\Sigma}$, $\mathbf{\Omega}$, $\mathbf{\Theta}$, $\mathbf{\Xi}$ **by** Algorithm 2.1 where each $\mathbf{X}_i$ and $\mathbf{X}_{i'}$ are replaced by $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_{i'}$, respectively.
  Update $\mu = \rho\mu$
end while
return $(\mathbf{w}_m^T\mathbf{\Phi}, \ldots, \mathbf{w}_K^T\mathbf{\Phi}) \in \mathbf{W}^T\mathbf{\Phi}$ and $(b_1, \ldots, b_K) \in \mathbf{b}$.

2.5.2 The Kernel extension of our method with inexact solutions

Our exact kernel method in Eqs. (32) and (33) scales quadratically as the number of instances increases. To effectively deal with the large number of instances in the dataset, we can use the optimal line search method in Eq. (25) for the kernel version of our method by replacing $\mathbf{X}_i$ with $\mathbf{\Phi}_i$:

$$\mathbf{w}_m = \mathbf{w}_m - s_m \nabla_{\mathbf{w}_m}, \tag{34}$$

where

$$\begin{aligned}\nabla_{\mathbf{w}_m}^T = \mathbf{w}_m^T &- \mu \sum_{i=1}^{N} [\mathbf{t}_i^m - \mathbf{w}_m^T\mathbf{\Phi}_i - \mathbf{1}b_m + \boldsymbol{\theta}_i^m/\mu]\mathbf{\Phi}_i^T \\ &- \mu \sum_{i'=1}^{N'} \sum_{m'=1}^{K} [\mathbf{u}_{i'}^{m'} - \mathbf{w}_m^T\mathbf{\Phi}_{i'} - \mathbf{1}b_m + \boldsymbol{\xi}_{i'}^{m'}/\mu]\mathbf{\Phi}_{i'}^T,\end{aligned} \tag{35}$$

and

$$s_m = \frac{\left(\mathbf{w}_m^T - \mu\sum_{i=1}^{N}\hat{\mathbf{t}}_i^m\mathbf{\Phi}_i^T - \mu\sum_{i'=1}^{N'}\sum_{m'=1}^{K}\hat{\mathbf{u}}_{i'}^{m'}\mathbf{\Phi}_{i'}^T\right)\nabla_{\mathbf{w}_m}}{\left\|\nabla_{\mathbf{w}_m}\right\|_2^2 + \mu\sum_{i=1}^{N}\left\|\nabla_{\mathbf{w}_m}^T\mathbf{\Phi}_i\right\|_2^2 + \mu K\sum_{i'=1}^{N'}\left\|\nabla_{\mathbf{w}_m}^T\mathbf{\Phi}_{i'}\right\|_2^2}. \tag{36}$$

The computational complexity of inexact kernel method in Eq. (34) is $O((n_1 + \cdots + n_N)\cdot d_\phi)$, which is apparently more computationally efficient than Eq. (32) when the number of instances is larger than the number of kernel features d_ϕ. However, the inexact kernel pdMISVM is not applicable to the kernel function of an infinite number of features such as radial basis function. Therefore in our experiments, in contrast to using the radial basis function (RBF) for the exact method, the kernel extension of our method with exact solution

employs the degree-2 polynomial function (poly) kernel. In case of degree-2 polynomial kernel $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + c)^2$, the complexity becomes $O((n_1 + \cdots + n_N) \cdot d^2)$, since the feature map ϕ is given by:

$$\begin{aligned}\phi(\mathbf{x}) = [&x_n^2, \ldots, x_1^2, \sqrt{2}x_n x_{n-1}, \cdots, \sqrt{2}x_n x_1, \sqrt{2}x_{n-1}x_{n-2}, \cdots, \sqrt{2}x_{n-1}x_1, \\ &\cdots, \sqrt{2}x_2 x_1, \sqrt{2c}x_n, \cdots, \sqrt{2c}x_1, c].\end{aligned} \tag{37}$$

Apparently, our inexact method that scales to the number of features is especially useful to kernel features, because the kernel feature map usually increases the dimensionality.

3 Experiments

In this section we explore the performance of our linear/kernel and exact/inexact pdMISVM implementations. We first test our method against an array of standard MIL benchmark datasets to explore how our implementations compare against other recent MIL methods. We follow the baseline experiments with an investigation into increasingly complex natural scene data to determine the performance characteristics of our approach. Then, we conduct experiments with synthetic data to illustrate the scalability of our approach and experimentally verify the expected computational complexity/performance characteristics of our approach compared to others. We follow with a discussion of the interpretability of our method on three multi-instance datasets derived from two well-known baseline datasets.

3.1 Experimental settings and datasets

As discussed in Sect. 2.5.1, because it is not possible to directly access the kernel features of radial basis function (RBF), we use the degree-2 polynomial kernel (poly) for our inexact kernel method. We compare our methods of linear/kernel and exact/inexact versions against ten recent MIL learning algorithms: (1) a single-instance learning (SIL) approach that assigns the bags' labels to all instances during training and returns the maximum response for each bag/class-pair at test time for the testing bags' instances; (2) the miSVM and (3) MISVM algorithms [1] that assume that at least one instance per bag is positive to classify a bag as positive; (4) the NSK algorithm [25], a bag-based method, that maps the entire bag to a single-instance by way of a kernel function; (5) the sMIL and (6) sbMIL [2] algorithms which expect that only a small number of instances within a bag are classified as positive and combine instance-level and bag-level relationships to make a prediction. We also compare our approach against two end-to-end MIL algorithms, (7) miNet and (8) MINet [26], based on deep neural-networks (DNN). Finally, the two DNN MIL models using the attention mechanism are compared: (9) Attention-based deep multiple instance learning (AMIL) [16] calculates the parameterized attention (importance) score for each instance to generate the probability distribution of bag labels; (10) loss-based attention for deep multiple instance learning (LAMIL) [37] proposes to learn the instance scores and predictions jointly by integrating the attention mechanism with the loss function. These two attention-based DNN methods have demonstrated the state-of-the-art classification performance in MIL.

These methods are compared against the proposed pdMISVM (Ours) method, and the inexact variation, described in Algorithm 2.1 and Algorithm 2.2. The grid search and performance calculations for each method-dataset pair are conducted using the MLJ library [38] and

are included with our code.[1] All experiments were run on an Intel Xeon processor running at 2.20GHz using 126GB of RAM, running Ubuntu 18.04.4 LTS. The competing SVM-based methods are implemented using a library[2] written by Doran et.al. [39], while the DNN methods are implemented using the code[3] provided as a companion to the paper [16, 26, 37]. Methods that take longer than one-thousand seconds to train during a single cross-validation are considered "timed-out" (T/O) and their performance metrics are not provided.

Each method is compared against a synthetic dataset and ten multi-instance datasets that are normalized to have zero mean and unit variance. The synthetic dataset contains 10 to 1,000 bags with three to five instances per bag and 10 to 1,000 features per instance. The first instance per bag is constructed from two normally distributed clusters with a standard deviation of one; the second to fifth instances per bag contain uniform random noise.

The MUSK-2 [9], Elephant, Fox, Colon [40], and Tiger [1] datasets are standard small-scale MIL evaluation datasets and are widely cited in MIL literature as benchmarks. The MUSK-2 dataset is designed to classify chemical compounds as either "musk" or "non-musk" which describes the chemical properties of a given compound; bags within this dataset are representative of the possible conformations of the labeled compound. The MUSK-2 dataset contains 39 positive and 63 negative bags with 166 features per instance. The Elephant, Fox and Tiger datasets are derived from the Corel image dataset [41] and each contain 100 positive and 100 negative bags with 143 non-zero features per instance. The Colon [40] consists of 25,000 histopathological images (instances) generated from 750 lung tissue images (bags). There are 5 classes of lung benign tissue, lung adenocarcinoma, lung squamous cell carcinoma, colon adenocarcinoma, and colon benign tissue in Colon dataset.

The MNIST-bags [16] dataset contains 100 positive and 100 negative bags where a bag is made up of a random number of 28×28 greyscale images taken from the MNIST dataset. A bag is given a positive label if it contains a '9' and negative label if it does not. For our experiments the average number of bags is ten, thus the witness rate for positive bags is 10%, on average. This low witness rate makes this a challenging dataset for the chosen MIL algorithms.

The SIVAL dataset was specifically designed for content-based image retrieval (CBIR) and contains natural scene images consisting of 25 categories with 60 images per category for a total of 1500 bags. In this work, we use the processed dataset provided in the initial work of Rahmani et.al. [42] and create a new dataset derived from the raw SIVAL images. In the original processed SIVAL dataset, the images are segmented into 30 or 31 instances, depending on the picture, consisting of 30 features each. In total, there are 47,414 instances across the entire SIVAL dataset. In order to explore the prediction and runtime performance of the compared methods, we construct a few subsets of this dataset containing a predetermined number of classes. Specifically, we construct the SIVAL-3, SIVAL-5, SIVAL-10, SIVAL-15, and SIVAL-25 datasets each containing three, five, ten, fifteen, and twenty-five classes from the SIVAL dataset, of 180, 300, 600, 900, and 1500 bags.

In addition, we construct the "SIVAL-25-deep" dataset, which is inspired by the "hybrid" approach detailed by Zheng *et.al.,* [43], which investigates the ongoing shift from SIFT-based descriptors [44] to convolutional neural networks for generating image descriptors. To create this multi-instance dataset, we extract patches from the raw SIVAL images using the EdgeBox [45] proposal generator (`eta=0.2`, `minScore=0.04`, `maxBoxes=200`)

[1] https://github.com/minds-mines/pdMISVM.jl.

[2] https://github.com/garydoranjr/misvm.

[3] https://github.com/yanyongluan/MINNs.

provided in the OpenCV library. [4] These extracted patches are fed into a pre-trained AlexNet [46] convolutional neural network where the second to last fully connected layer (F10) is used to represent each instance by 4,096 features. We note that more complex/newer deep neural architectures, and other proposal generators, c ould be used to create this patch-level embedding but leave this to future work. This process is repeated for every image (where object proposals are detected by EdgeBox) and results in 1,463 bags for a total of 80,561 instances. The SIVAL-25-deep dataset is our attempt at a modernization of the standard SIVAL dataset; the pipeline used to generate this benchmark is provided with our code.

3.2 Classification performance

In Tables 1 and 2 we provide the classification performance of our approach compared against the other competing MIL algorithms described above in Sect. 3.1. Our goal is to verify that our approach matches the performance of the other MIL algorithms. For each dataset-method pair we report the accuracy (ACC) and balanced accuracy (BACC) results across ten sixfold cross validation experiments. We can see from Table 1 that our method gives comparable performance on the MUSK-2, Elephant, and Tiger datasets; this applies for both the exact and inexact implementations. Interestingly, the inexact linear version of our approach outperforms all other methods on both the Fox and MNIST-Bags datasets. That can be naturally explained by the previous study [47] which has shown some implementations of SVM obtain the highest accuracy before their objectives reach their minimum. In the Colon dataset, our kernel versions show superior performance in classifying the different shapes of tissues, which shows the benefits of kernel functions in our model.

In Table 2 our exact linear pdMISVM only slightly outperforms the next best performing method on the SIVAL-3 dataset; this impressive performance result does not hold for the inexact version. Although, the inexact method performs better (in comparison) when the number of classes/bags increase. The inexact linear methods shows surprisingly impressive results on SIVAL-25-deep dataset which are recorded just within the time-budget; this significant performance improvement can be seen very clearly in the comparison between the confusion matrices in Figs. 2 and 3. In comparison of training times (TT) between our exact and inexact kernel pdMISVM, we can clearly see that the inexact version scales to the increasing number of bags (the number of bags increases in the order of SIVAL-3, 5, 10, 15, and 25) better than the exact version. This empirically verifies the analytical complexity discussed in Sect. 2.5. It is clear from these results that the exact/inexact and linear/kernel methods are capable of providing competitive performance results on a variety of multi-instance datasets.

3.3 Bag/feature scalability

The *key contribution* of this work is that the derived algorithms described in Sects. 2.3 and 2.4 scale to large datasets. This can be clearly seen in the SIVAL-25 column of Table 2 where our methods are the only ones that are able to fit a model within one-thousand seconds. In order to further validate this finding, in Fig. 4 we report training time results on a synthetic multi-instance dataset where we increase the number of bags as described in Sect. 3.1. In this timing experiment, we use the degree-2 polynomial function for both of exact/inexact kernel methods to compare them fairly. Our linear-exact, linear-inexact, and poly-inexact methods scale well with respect to the number of bags which shows the importance of our primal–

[4] https://github.com/opencv/opencv.

Table 1 Classification performance and train time (seconds) of our method and ten other MIL learning methods on six benchmark datasets

	MUSK-2			Elephant		
Model	ACC	BACC	TT	ACC	BACC	TT
SIL	0.657 ± 0.136	0.711 ± 0.105	17.55	0.695 ± 0.056	0.694 ± 0.063	1.67
miSVM	0.657 ± 0.136	0.696 ± 0.103	278.07	0.790 ± 0.043	0.784 ± 0.036	13.72
MISVM	0.794 ± 0.081	0.768 ± 0.107	252.16	0.840 ± 0.052	0.844 ± 0.045	21.11
NSK	0.814 ± 0.058	0.808 ± 0.063	1.38	0.854 ± 0.081	0.855 ± 0.072	1.55
sMIL	0.725 ± 0.127	0.732 ± 0.128	21.09	0.500 ± 0.057	0.500 ± 0.000	1.44
sbMIL	0.657 ± 0.141	0.592 ± 0.130	27.82	0.665 ± 0.036	0.663 ± 0.055	2.13
miNet	0.853 ± 0.104	0.847 ± 0.121	17.79	0.844 ± 0.081	0.849 ± 0.068	23.62
MINet	**0.882 ± 0.064**	0.864 ± 0.070	19.61	0.860 ± 0.053	0.864 ± 0.050	23.88
AMIL	0.876 ± 0.083	**0.869 ± 0.131**	196.14	0.829 ± 0.092	0.835 ± 0.057	26.39
LAMIL	0.865 ± 0.091	0.865 ± 0.083	225.71	**0.883 ± 0.106**	**0.872 ± 0.064**	30.84
Ours (linear, exact)	0.794 ± 0.152	0.802 ± 0.160	**0.72**	0.825 ± 0.053	0.822 ± 0.062	*0.23*
Ours (linear, inexact)	0.804 ± 0.080	0.811 ± 0.085	*0.83*	0.830 ± 0.047	0.837 ± 0.042	**0.14**
Ours (RBF, exact)	0.835 ± 0.071	0.854 ± 0.076	1.92	0.843 ± 0.062	0.854 ± 0.075	0.47
Ours (poly, inexact)	0.829 ± 0.083	0.843 ± 0.081	0.95	0.852 ± 0.088	0.841 ± 0.06	0.62
	Colon			Fox		
Model	ACC	BACC	TT	ACC	BACC	TT
SIL	0.631 ± 0.095	0.639 ± 0.108	15.06	0.575 ± 0.050	0.579 ± 0.064	1.77
miSVM	0.629 ± 0.103	0.632 ± 0.097	249.41	0.595 ± 0.087	0.602 ± 0.089	28.03
MISVM	0.647 ± 0.098	0.651 ± 0.081	224.61	0.570 ± 0.037	0.569 ± 0.044	31.92
NSK	0.720 ± 0.064	0.741 ± 0.092	2.81	0.535 ± 0.099	0.539 ± 0.102	0.99
sMIL	0.682 ± 0.082	0.690 ± 0.093	22.19	0.529 ± 0.115	0.529 ± 0.086	1.25
sbMIL	0.593 ± 0.077	0.624 ± 0.084	29.86	0.599 ± 0.093	0.587 ± 0.090	3.42
miNet	0.780 ± 0.097	0.765 ± 0.101	39.10	0.561 ± 0.056	0.563 ± 0.060	23.85

Table 1 continued

Model	Colon			Fox		
	ACC	BACC	TT	ACC	BACC	TT
MINet	0.796 ± 0.076	0.807 ± 0.093	45.78	0.585 ± 0.120	0.585 ± 0.089	24.20
AMIL	0.782 ± 0.091	0.797 ± 0.092	293.45	*0.639* ± 0.039	0.637 ± 0.087	29.53
LAMIL	0.823 ± 0.041	0.816 ± 0.085	325.07	0.628 ± 0.052	0.633 ± 0.091	34.72
Ours (linear, exact)	0.783 ± 0.085	0.762 ± 0.075	**1.54**	0.590 ± 0.103	0.586 ± 0.097	*0.30*
Ours (linear, inexact)	0.807 ± 0.104	0.812 ± 0.053	*1.96*	**0.640 ± 0.047**	**0.647 ± 0.045**	**0.20**
Ours (RBF, exact)	**0.829 ± 0.143**	0.831 ± 0.121	23.42	0.628 ± 0.085	0.64 ± 0.081	0.82
Ours (poly, inexact)	0.825 ± 0.087	**0.835 ± 0.094**	2.26	0.631 ± 0.072	0.635 ± 0.082	0.64
Model	Tiger			MNIST-Bags		
	ACC	BACC	TT	ACC	BACC	TT
SIL	0.695 ± 0.070	0.697 ± 0.053	0.47	0.420 ± 0.156	0.495 ± 0.078	0.88
miSVM	0.760 ± 0.082	0.762 ± 0.083	12.25	0.401 ± 0.104	0.502 ± 0.074	1.50
MISVM	0.790 ± 0.091	0.792 ± 0.092	17.09	0.499 ± 0.099	0.489 ± 0.073	8.99
NSK	0.795 ± 0.113	0.787 ± 0.118	0.87	0.640 ± 0.166	0.641 ± 0.167	1.21
sMIL	0.670 ± 0.065	0.660 ± 0.068	0.79	0.609 ± 0.069	0.510 ± 0.073	**0.21**
sbMIL	0.626 ± 0.064	0.627 ± 0.046	1.64	0.539 ± 0.066	0.501 ± 0.066	0.63
miNet	0.805 ± 0.067	0.805 ± 0.058	24.33	0.608 ± 0.137	0.556 ± 0.109	15.05
MINet	**0.820 ± 0.083**	0.805 ± 0.087	24.48	0.591 ± 0.139	0.511 ± 0.075	15.35
AMIL	0.798 ± 0.062	0.793 ± 0.071	44.26	0.603 ± 0.112	0.59 ± 0.073	29.61
LAMIL	0.803 ± 0.081	**0.812 ± 0.048**	29.71	0.629 ± 0.093	0.604 ± 0.034	24.91
Ours (linear, exact)	0.795 ± 0.084	0.798 ± 0.081	*0.21*	0.616 ± 0.180	0.582 ± 0.135	3.30
Ours (linear, inexact)	0.780 ± 0.063	0.780 ± 0.064	**0.16**	**0.672 ± 0.105**	**0.645 ± 0.100**	*0.45*
Ours (RBF, exact)	0.812 ± 0.091	0.802 ± 0.067	1.21	0.63 ± 0.071	0.627 ± 0.092	2.41
Ours (poly, inexact)	0.807 ± 0.104	0.806 ± 0.082	0.92	0.626 ± 0.089	0.619 ± 0.093	6.83

The reported accuracy and standard deviations are calculated across ten sixfold cross-validation experiments. Best results are marked in bold, second best in italics

Table 2 Classification and train-time (seconds) performance of our method and ten other MIL learning methods on variants of the SIVAL dataset across a different number of classes (and bags) and preprocessing pipelines

	SIVAL-3			SIVAL-5		
Model	ACC	BACC	TT	ACC	BACC	TT
SIL	0.433 ± 0.167	0.474 ± 0.144	26.19	0.140 ± 0.057	0.000 ± 0.000	72.65
miSVM	0.856 ± 0.062	0.858 ± 0.053	469.83	–	–	T/O
MISVM	0.909 ± 0.058	0.908 ± 0.066	457.65	–	–	T/O
NSK	0.633 ± 0.070	0.634 ± 0.085	1.44	0.547 ± 0.088	0.551 ± 0.089	6.27
sMIL	0.544 ± 0.119	0.541 ± 0.121	7.52	0.443 ± 0.112	0.441 ± 0.131	68.51
sbMIL	0.767 ± 0.123	0.775 ± 0.128	44.08	–	–	T/O
miNet	0.644 ± 0.066	0.672 ± 0.038	72.49	0.203 ± 0.104	0.240 ± 0.101	72.49
MINet	0.589 ± 0.066	0.558 ± 0.043	78.91	0.253 ± 0.043	0.261 ± 0.063	78.91
AMIL	0.846 ± 0.037	0.856 ± 0.054	592.38	–	–	T/O
LAMIL	0.853 ± 0.071	0.863 ± 0.101	794.52	–	–	T/O
Ours (linear, exact)	**0.911 ± 0.058**	**0.917 ± 0.059**	**0.48**	**0.840 ± 0.051**	**0.846 ± 0.047**	**1.02**
Ours (linear, inexact)	0.767 ± 0.067	0.763 ± 0.069	*0.74*	0.730 ± 0.065	0.733 ± 0.057	*1.67*
Ours (RBF, exact)	0.713 ± 0.048	0.726 ± 0.048	1.20	0.793 ± 0.065	0.812 ± 0.059	10.71
Ours (poly, inexact)	0.698 ± 0.052	0.704 ± 0.035	2.93	0.746 ± 0.053	0.797 ± 0.103	9.13
	SIVAL-10			SIVAL-15		
Model	ACC	BACC	TT	ACC	BACC	TT
SIL	–	–	T/O	–	–	T/O
miSVM	–	–	T/O	–	–	T/O
MISVM	–	–	T/O	–	–	T/O
NSK	0.462 ± 0.042	0.469 ± 0.053	50.2	0.432 ± 0.056	0.452 ± 0.074	196.28
sMIL	–	–	T/O	–	–	T/O
sbMIL	–	–	T/O	–	–	T/O
miNet	0.100 ± 0.020	0.102 ± 0.024	203.93	0.093 ± 0.021	0.097 ± 0.025	607.42

Table 2 continued

Model	SIVAL-10			SIVAL-15		
	ACC	BACC	TT	ACC	BACC	TT
MINet	0.135 ± 0.037	0.135 ± 0.019	226.43	0.121 ± 0.033	0.125 ± 0.031	584.65
AMIL	–	–	T/O	–	–	T/O
LAMIL	–	–	T/O	–	–	T/O
Ours (linear, exact)	**0.732 ± 0.057**	**0.742 ± 0.057**	**7.88**	0.628 ± 0.062	0.634 ± 0.035	**15.28**
Ours (linear, inexact)	0.577 ± 0.061	0.581 ± 0.055	*13.57*	0.492 ± 0.049	0.504 ± 0.060	*20.34*
Ours (RBF, exact)	0.601 ± 0.049	0.597 ± 0.037	71.09	0.701 ± 0.052	0.713 ± 0.053	348.57
Ours (poly, inexact)	0.583 ± 0.052	0.589 ± 0.051	21.42	**0.713 ± 0.067**	**0.716 ± 0.049**	42.09
Model	SIVAL-25			SIVAL-25-deep		
	ACC	BACC	TT	ACC	BACC	TT
SIL	–	–	T/O	–	–	T/O
miSVM	–	–	T/O	–	–	T/O
MISVM	–	–	T/O	–	–	T/O
NSK	–	–	T/O	–	–	T/O
sMIL	–	–	T/O	–	–	T/O
sbMIL	–	–	T/O	–	–	T/O
miNet	–	–	T/O	–	–	T/O
MINet	–	–	T/O	–	–	T/O
AMIL	–	–	T/O	–	–	T/O
LAMIL	–	–	T/O	–	–	T/O
Ours (linear, exact)	0.487 ± 0.041	0.489 ± 0.039	**24.01**	–	–	T/O
Ours (linear, inexact)	0.388 ± 0.045	0.383 ± 0.044	*36.17*	**0.888 ± 0.040**	**0.887 ± 0.036**	**977.66**
Ours (RBF, exact)	–	–	T/O	–	–	T/O
Ours (poly, inexact)	**0.526 ± 0.04**	**0.530 ± 0.051**	121.08	–	–	T/O

The reported accuracy and standard deviations are calculated across ten sixfold cross-validation experiments. Best results are marked in bold, second best in italics

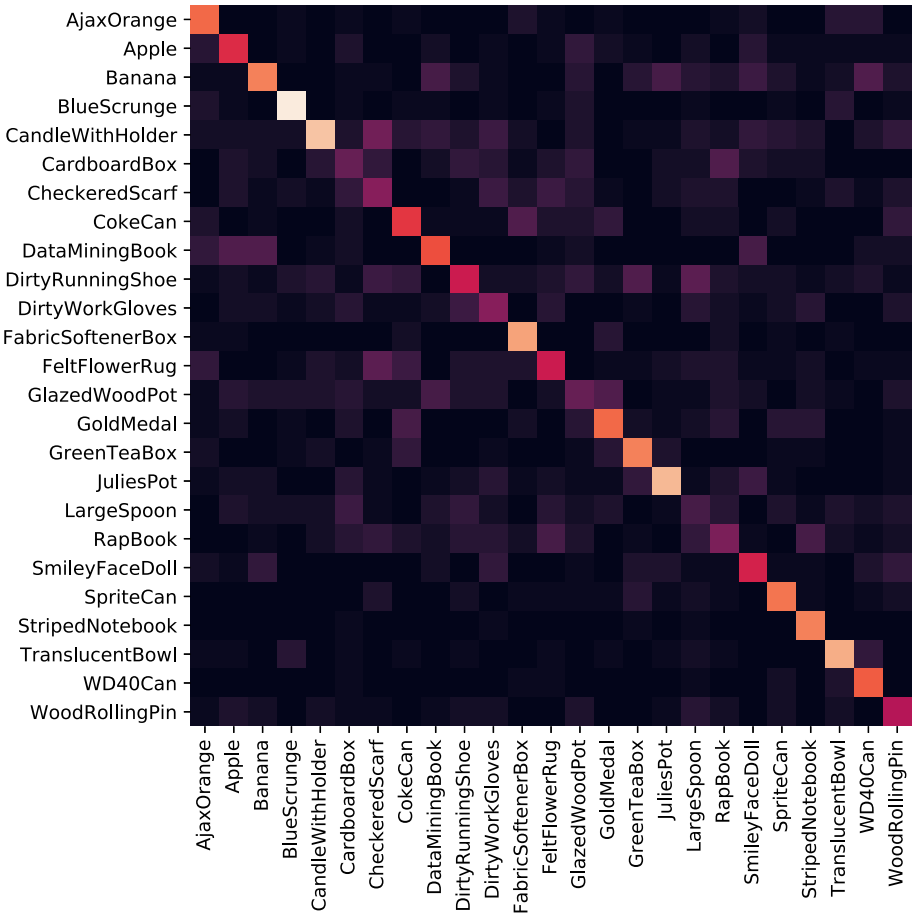


Fig. 2 Confusion matrix of the exact linear pdMISVM tested on the original SIVAL-25 dataset with 30 features per-instance. Results are derived from a sixfold cross-validation experiment across all 1500 bags

dual derivation. However, the training time of our poly-exact model increases rapidly as the number of instances increases, demonstrating the usefulness of our inexact kernel method. This conclusion is especially clear when our method is compared against SVM-based MIL methods which rely on repeatedly solving quadratic programming problems.

Although the initial pdMISVM derivation scales well with respect to bags, it does not scale to the number of features when it is large. This is due to the fact that the update for each $\mathbf{w}_k$ in Eq. (10) requires solving a least squares problem which scales quadratically as the number of features increase. To address this limitation, we proposed an optimal line search method to improve the scalability of our approach in Eq. (25). We conduct a timing experiment using synthetic data where the number of features is increased to see if our methods provide improved runtime performance. We can see in Fig. 5 that the proposed linear-inexact and poly-exact methods significantly reduce the training time of our approach as the number of features increase. We note that in our exact kernel method, we don't need to directly access the kernel features as discussed in Sect. 2.5.2.

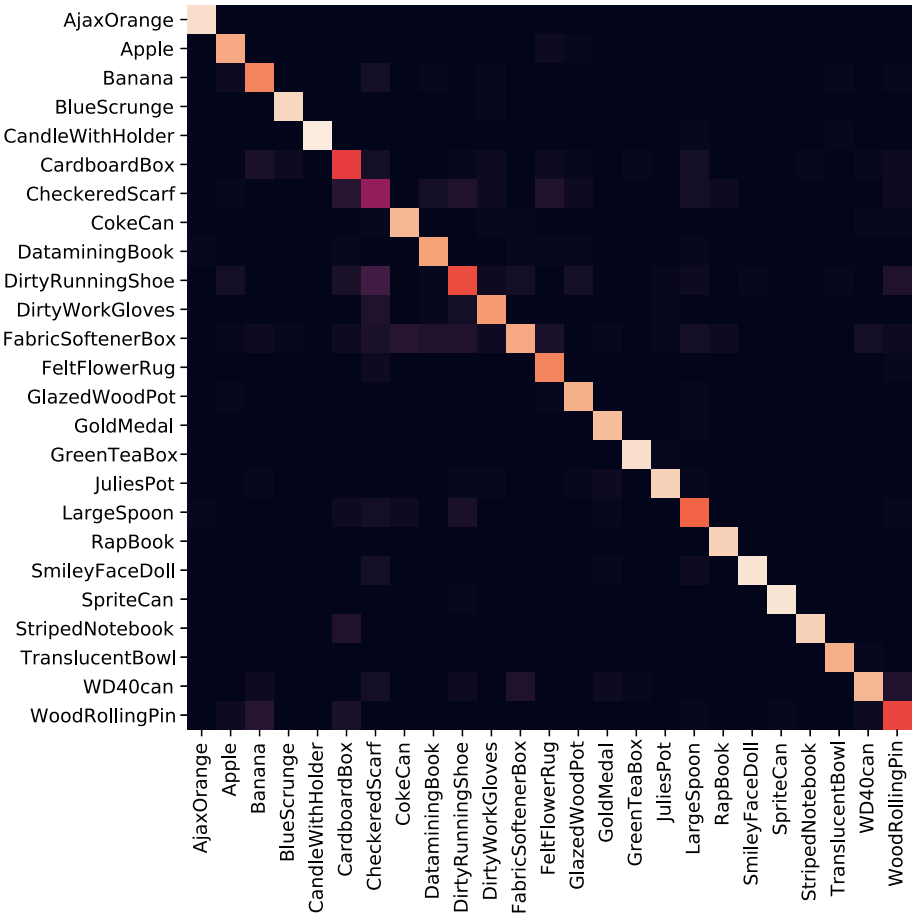


Fig. 3 Confusion matrix of the inexact linear pdMISVM approach tested on the SIVAL-25-deep dataset created from the patch-wise application of a convolutional neural network as a pre-processing step. Results are derived from a sixfold cross-validation experiment across all 1463 bags

3.4 Model interpretability

In addition to the promising predictive performance and scalability of our method, we note that instance-based methods such as ours come with an additional benefit: *interpretability*. Instance-based methods such as miSVM, MISVM, and the proposed pdMISVM method, identify an explicit instance within a bag that is responsible for the predicted label. We use this phenomenon to explore the limitations of our method on the MNIST-bags dataset and showcase patches identified during the SIVAL experiment across a number of different classes.

For the MNIST-bags dataset we plot the learned positive and negative class coefficients associated with the two learned hyperplanes in Fig. 6 (*e.g.*, $\mathbf{w}_1$ and $\mathbf{w}_2$). In addition, we plot four randomly chosen testing bags and what instance was chosen by the multi-instance decision function for the positive class hyperplane in Fig. 7. On the left-hand side of Fig. 6, we can see that our method can roughly detect the loop at the top of the '9' although it is clear

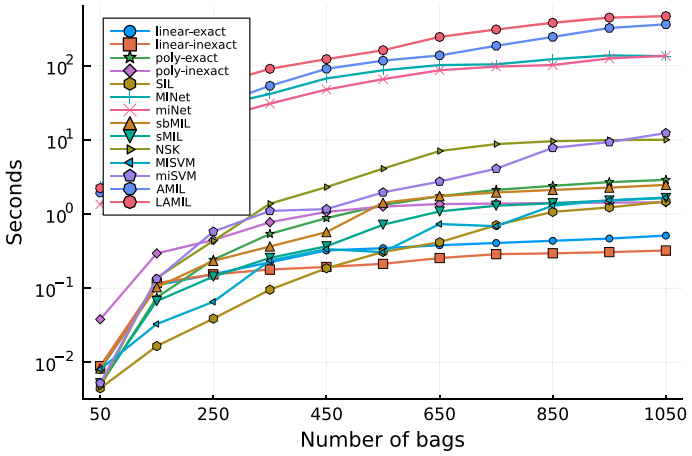


Fig. 4 Time to train our method and other MIL methods on synthetic multi-instance data where the number of bags increase. Both the linear-exact and linear-inexact methods end up training faster than the competing methods once the number of synthetic bags is greater than eight-hundred

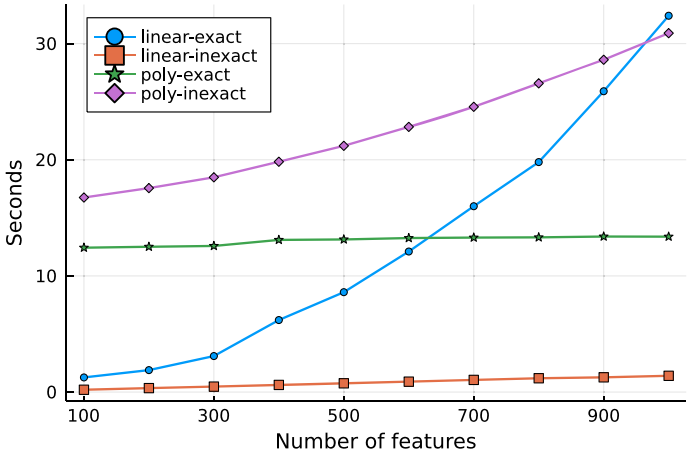


Fig. 5 Elapsed time to train the linear/kernel and exact/inexact methods on synthetic multi-instance data as the number of features is varied. As expected, the linear-exact and poly-inexact methods perform poorly as the number of features increases, but the linear-inexact and poly-exact method continues to scale linearly and almost constantly

from this interpretation that our approach will not be able to handle even moderate translation or rotation if it is only provided with raw-pixel values. Additionally, even though our method correctly classifies the first bag, it incorrectly identifies the '4' as the witness instance; we can see that a '4' appears to be contaminating the learned coefficients displayed in Fig. 6. In order to solve this problem it is likely that additional preprocessing will be required to extract more descriptive features from instances within the MNIST-bags dataset beyond raw pixels for our method to be effective.

In order to illustrate how effective feature extraction can aid in the interpretability of our method, we extend our discussion to the SIVAL-25 and SIVAL-25-deep datasets. In Fig. 8, we provide image patches identified by our approach on images chosen from the SIVAL-25

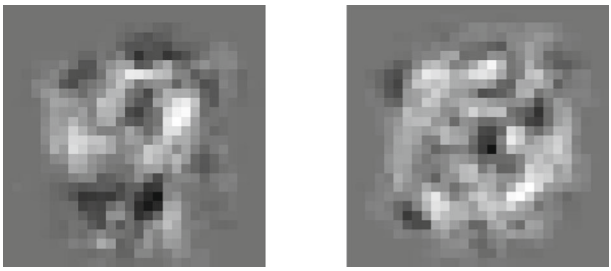


Fig. 6 Learned class-specific hyperplanes of the pdMISVM method on the MNIST-bags dataset plotted in a 28×28 grid. Left: Learned coefficients for predicting whether a bag contains the MNIST handwritten digit '9'. Right: The learned coefficients for predicting whether a bag *does not* contain the MNIST digit '9'

dataset. We can see that our method identifies distinctive visual characteristics in each of the classes. For example, the bag representing a "Banana" is identified by a distinctive patch along the length of the fruit while in the "Apple" image our approach identifies the round patch on top of the fruit. Similarly, in Fig. 9 we present the neural-network embedded patches extracted via the EdgeBox detection algorithm and the identified patches. We can clearly see in Fig. 9 that our approach is able to accurately localize the most distinctive parts of the object, at the patch level, within the image. For example, the medal is recognized by the "gold" part while the "bowl" of the spoon is recognized.

Remarkably, the results in Figs. 8 and 9 show that when our method is given a set of sufficiently descriptive object proposals/patches, paired with a bag-level label, our method can accurately locate objects within an image. This is one of the significant advantages of instance-based MIL methods over traditional single-instance learning methods that require *all* instances to be labeled. We anticipate that this framework could be extended to investigate and interpret the effectiveness of pre-trained neural networks on an assortment of datasets that can be formulated as MIL problems. We plan to further investigate different aspects of our approach under different object proposal methods [48, 49], neural architectures [50], and applications [13, 14, 51].

3.5 Capability to learn nonlinear decision boundaries

As an important extension of our pdMISVM method to deal with nonlinearly separable data, we introduced the kernel versions of our exact and inexact methods. In this subsection, we evaluate their classification performances.

We implement the kernel version of the pdMISVM method and compare it against the linear version in Fig. 10. We can see this extension successfully extends our approach to correctly classify data that is not linearly separable. In this paper, we propose two versions of kernel pdMISVM which scales to the number of instances or features, which enable to efficiently learn the nonlinear decision boundaries from the large dataset.

3.6 A case study on neuroimaging data

While we have demonstrated the effectiveness of our new pdMISVM method from a number of perspectives in the previous subsections, in this subsection we apply our new methods on a medical imaging dataset to verify its capability to solve real-world problems.

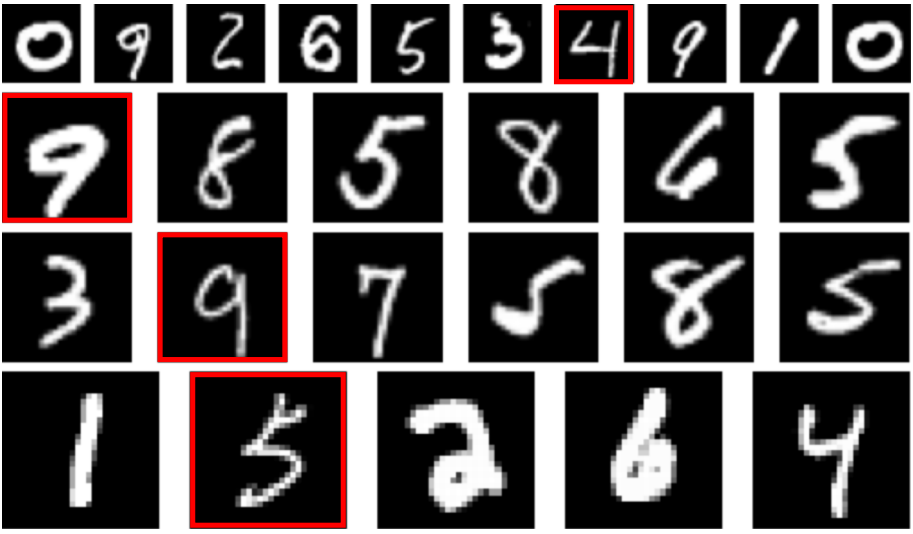


Fig. 7 Instance identification results on the first five testing bags of our method on the MNIST-bags dataset with the detectors in Fig. 6. Our approach correctly classifies the first, second and third bags. Although the first bag is classified correctly the "9"s are not properly identified

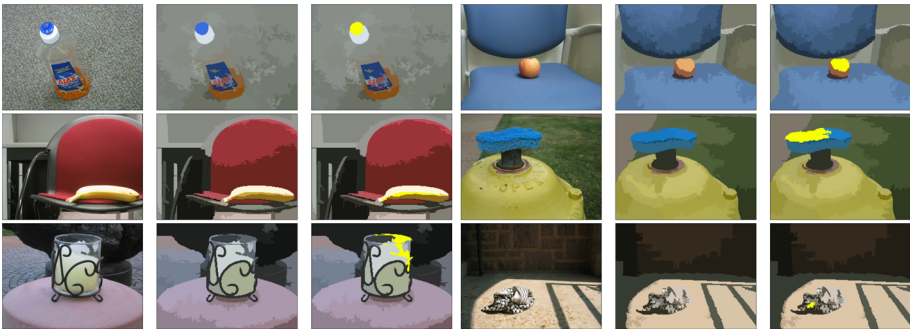


Fig. 8 Instance identification on the SIVAL-25 dataset across different classes. In each set of three pictures the leftmost is the original image, the middle shows the bag of patches extracted by the original authors, and the final image highlights the patch identified by our approach for classifying the image

3.6.1 Comparisons of the classification capabilities

Alzheimer's disease (AD) is a serious neurodegenerative condition in which people suffer from the deterioration of cognitive functions. To verify the capability of our new methods to solve real-world problems, we apply them to the Alzheimer's Disease Neuroimaging Initiative (ADNI) [52] dataset which provides the comprehensive neuroimagings such as voxel-Based Morphometry (VBM) and FreeSurfer (FS). We collect the magnetic resonance imaging (MRI) scans and their diagnosis in Alzheimer's disease, mild cognitive decline, and healthy condition of 821 participants from ADNI database[5]. We perform VBM and FS automated parcellation on the MRIs following [53] and extract mean modulated gray matter measures for 90 target regions of interest (ROI). Because the different number of MRI

[5] https://adni.loni.usc.edu.

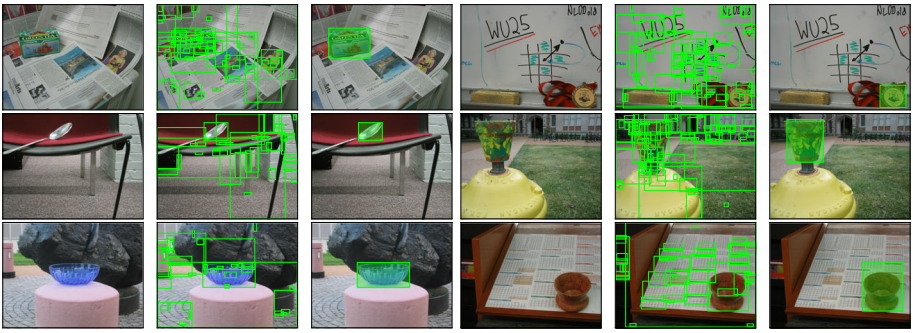


Fig. 9 Instance identification on the SIVAL-25-deep dataset across different classes. In each set of three pictures the leftmost is the original image, the middle shows the bag of patches extracted by the EdgeBox detector, and the final image highlights the patch identified by our approach for classifying the image

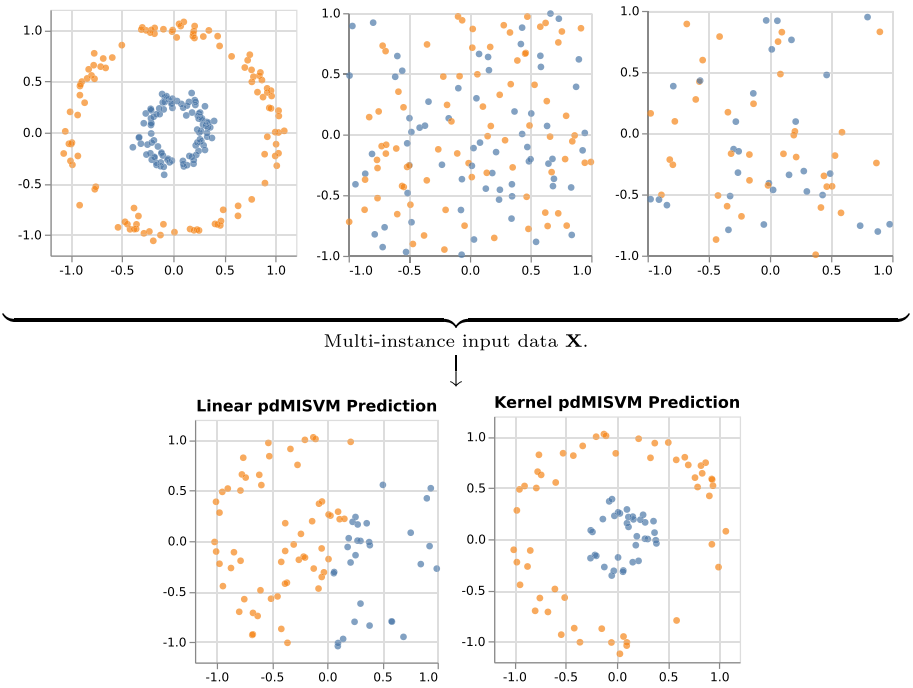


Fig. 10 The predictions of linear and RBF kernel (exact) pdMISVM on synthetic multi-instance data. Each bag in the training dataset $\mathbf{X}$ has up to three instances, where only the first instance determines the correct classification. The kernel extension of our approach is able to correctly learn a nonlinear decision boundary to separate the two classes

scans have been captured across the participants, it is difficult to directly apply the standard statistical methods to all of the neuroimagings provided. In this case study, we formulate each neuroimaging as an instance and each participant as a bag to predict their AD diagnosis.

In Table 3, we report the classification performances of ours and the other competing models. Although the deep learning models (miNet, MINet, AMIL, and LAMIL) have shown the promising performances, they require the significant amount of training time. On the other

Table 3 Classification and train-time (seconds) performance of our method and ten other MIL learning methods on ADNI dataset

	FS		
Model	ACC	BACC	TT
SIL	0.456 ± 0.076	0.493 ± 0.091	14.07
miSVM	0.668 ± 0.076	0.68 ± 0.071	93.17
MISVM	0.776 ± 0.058	0.761 ± 0.036	51.83
NSK	0.484 ± 0.083	0.481 ± 0.073	**0.93**
sMIL	0.406 ± 0.075	0.435 ± 0.078	7.42
sbMIL	0.593 ± 0.077	0.628 ± 0.081	32.81
miNet	0.658 ± 0.053	0.671 ± 0.045	59.09
MINet	0.687 ± 0.101	0.691 ± 0.071	48.08
AMIL	0.729 ± 0.071	0.735 ± 0.064	428.07
LAMIL	**0.802 ± 0.106**	0.79 ± 0.083	794.52
Ours (linear, exact)	0.784 ± 0.082	0.796 ± 0.084	*1.85*
Ours (linear, inexact)	0.790 ± 0.091	0.802 ± 0.083	2.14
Ours (RBF, exact)	0.755 ± 0.083	0.784 ± 0.090	3.62
Ours (poly, inexact)	0.795 ± 0.064	**0.803 ± 0.075**	6.14
	VBM		
Model	ACC	BACC	TT
SIL	0.392 ± 0.043	0.427 ± 0.058	16.81
miSVM	0.584 ± 0.050	0.617 ± 0.042	81.91
MISVM	0.690 ± 0.050	0.717 ± 0.049	68.38
NSK	0.547 ± 0.088	0.551 ± 0.089	6.27
sMIL	0.483 ± 0.091	0.510 ± 0.019	8.08
sbMIL	0.608 ± 0.070	0.616 ± 0.024	29.14
miNet	0.703 ± 0.081	0.694 ± 0.073	58.17
MINet	0.714 ± 0.062	0.738 ± 0.059	40.50
AMIL	0.765 ± 0.117	0.783 ± 0.091	378.05
LAMIL	0.809 ± 0.074	0.815 ± 0.090	290.37
Ours (linear, exact)	0.808 ± 0.077	0.813 ± 0.084	**1.46**
Ours (linear, inexact)	**0.814 ± 0.058**	0.809 ± 0.041	*1.91*
Ours (RBF, exact)	0.747 ± 0.049	0.763 ± 0.072	3.84
Ours (poly, inexact)	0.803 ± 0.065	**0.818 ± 0.076**	5.85

The reported accuracy and standard deviations are calculated across ten sixfold cross-validation experiments. Best results are marked in bold, second best in italics

hands, four variations of our models show the comparable performance with a few seconds of training time.

3.6.2 Identification of brain regions

Likewise we analyze the hyperplane in Fig. 6 from MNIST digits dataset, we analyze the hyperplane to identify the brain regions exhibiting AD risk factors. Since the p-th feature of

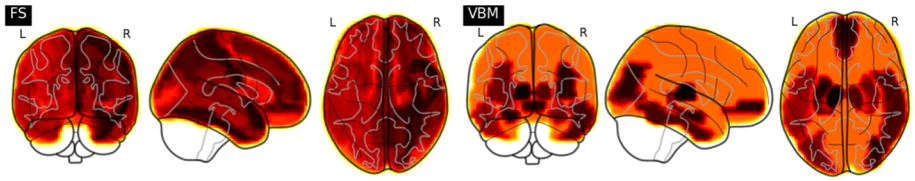


Fig. 11 Visualization of contributions of the brain regions to the AD diagnosis classification. The brain regions of the larger contribution are plotted with the darker colors. The top four AD relevant regions are identified in **FS:** left thalamus, left lateral ventrical, right caudate, and brodmann area 44, in **VBM:** left thalamus, left hippocampus, right medial occipital, and left amygdala

each instance is multiplied with the p-th weight of $\mathbf{w}_m$ to contribute to the response for the m-th class, we calculate the contribution of p-th feature as the summation of p-th weight of hyperplanes $\sum_{m=1}^{K} \|w_m^p\|$. Each feature of instance (neuroimage) represents each ROI in the brain, therefore we visualize the disease relevance of each brain region in Fig. 11.

The brain regions identified by our method (linear, exact) have been appeared in the previous medical literatures. For example, the pathological changes in the Brodmann area 44 (Broca's area) involves in the comprehension and production of verbs and communication [54]. Based on the previous study [55], the larger ventricular volume is the risk factor of dementia related disease in the future. The change in venticular volume is also associated with the cognitive decline and dementia [55, 56]. The atrophy of the caudate nucleus is also related to the cognitive decline [57]. The previous study [58] found that the anterior thalamus plays an important role in generating attention, and it is in charge of declarative memory functioning. The hippocampus is particularly susceptible to damage from AD and involves long-term memory and spatial navigation in AD patients [59]. The amygdala region, which is related to the emotional response, can also be easily damaged by AD [60]. The identified brain regions in this case study are well represented in the previous AD studies, and support the correctness of our methods by providing the further interpretability.

4 Conclusion

In this work, we propose a *primal–dual multi-instance SVM* method that is able to scale to large multi-instance datasets. Our SVM-based approach is able to handle data that grows in terms of bags as well as features since it avoids solving a quadratic programming problem that limits the adoption of traditional SVM-based MIL techniques. Throughout the manuscript, we provide detailed derivations, implementations, and experimental results which illustrate the utility of our approach on both synthetic and real-world datasets. In addition, we provide the kernel extensions of our approach which scale to the number of instances or features. Our experimental results on synthetic multi-instance data validate our kernel extension successfully learn the nonlinear decision boundaries. Finally, we investigate the interpretability of our method on benchmark multi-instance datasets and develop an extension to the ADNI dataset as part of this study. From the ADNI dataset, our method identifies the disease relevant brain regions which are in nice accordance with the existing medical studies.

Acknowledgements This work was supported in part by the National Science Foundation (NSF) under the grants of IIS 1652943, IIS 1849359, CNS 1932482 and CCF 2029543. Part of Lodewijk Brand's work was supported in part by the Department of Defense (DoD) Science, Mathematics And Research for Transformation (SMART) Scholarship-for-Service program.

References

1. Andrews S, Tsochantaridis I, Hofmann T (2002) Support vector machines for multiple-instance learning. In: NIPS, vol 2. Citeseer, pp 561–568
2. Bunescu RC, Mooney RJ (2007) Multiple instance learning for sparse positive bags. In: Proceedings of the 24th international conference on machine learning, pp 105–112
3. Wang H, Nie F, Huang H (2011) Learning instance specific distance for multi-instance classification. In: Twenty-fifth AAAI conference on artificial intelligence
4. Wang H, Huang H, Kamangar F, Nie F, Ding C (2011) Maximum margin multi-instance learning. Adv Neural Inf Process Syst 24:1–9
5. Wang H, Nie F, Huang H (2012) Robust and discriminative distance for multi-instance learning. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 2919–2924
6. Wang H, Deng C, Zhang H, Gao X, Huang H (2016) Drosophila gene expression pattern annotations via multi-instance biological relevance learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 30, p 1
7. Liu K, Wang H, Nie F, Zhang H (2018) Learning multi-instance enriched image representations via non-greedy ratio maximization of the ℓ_1-norm distances. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7727–7735
8. Brand L, Baker LZ, Wang H (2021) A multi-instance support vector machine with incomplete data for clinical outcome prediction of covid-19. In: Proceedings of the 12th ACM conference on bioinformatics, computational biology, and health informatics, pp 1–6
9. Dietterich TG, Lathrop RH, Lozano-Pérez T (1997) Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 89(1–2):31–71
10. Zhou ZH, Xue XB, Jiang Y (2005) Locating regions of interest in CBIR with multi-instance learning techniques. In: Australasian joint conference on artificial intelligence. Springer, pp 92–101
11. Wang H, Nie F, Huang H, Yang Y (2011) Learning frame relevance for video classification. In: Proceedings of the 19th ACM international conference on multimedia, pp 1345–1348
12. Cheplygina V, de Bruijne M, Pluim JP (2019) Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Med Image Anal 54:280–296
13. Settles B, Craven M, Ray S (2007) Multiple-instance active learning. In: Proceedings of the 20th international conference on neural information processing systems, pp 1289–1296
14. Xu Y, Zhu J-Y, Eric I, Chang C, Lai M, Tu Z (2014) Weakly supervised histopathology cancer image segmentation and classification. Med Image Anal 18(3):591–604
15. Wu J, Yu Y, Huang C, Yu K (2015) Deep multiple instance learning for image classification and auto-annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3460–3469
16. Ilse M, Tomczak J, Welling M (2018) Attention-based deep multiple instance learning. In: International conference on machine learning. PMLR, pp 2127–2136
17. Yan Y, Wang X, Guo X, Fang J, Liu W, Huang J (2018) Deep multi-instance learning with dynamic pooling. In: Asian conference on machine learning. PMLR, pp 662–677
18. Ray S, Craven M (2005) Supervised versus multiple instance learning: an empirical comparison. In: Proceedings of the 22nd international conference on machine learning, pp 697–704
19. Carbonneau M-A, Cheplygina V, Granger E, Gagnon G (2018) Multiple instance learning: a survey of problem characteristics and applications. Pattern Recogn 77:329–353
20. Wei X-S, Wu J, Zhou Z-H (2016) Scalable algorithms for multi-instance learning. IEEE Trans Neural Netw Learn Syst 28(4):975–987
21. Vatsavai RR (2012) Scalable multi-instance learning approach for mapping the slums of the world. In: SC companion: high performance computing, networking storage and analysis. IEEE, pp 833–837
22. Wang J, Zucker JD (2000) Solving multiple-instance problem: a lazy learning approach
23. Melendez J, van Ginneken B, Maduskar P, Philipsen RH, Reither K, Breuninger M, Adetifa IM, Maane R, Ayles H, Sánchez CI (2014) A novel multiple-instance learning-based approach to computer-aided detection of tuberculosis on chest x-rays. IEEE Trans Med Imaging 34(1):179–192
24. Rymarczyk D, Kaczyńska A, Kraus J, Pardyl A, Zieliński B (2021)Protomil: Multiple instance learning with prototypical parts for fine-grained interpretability. arXiv preprint arXiv:2108.10612
25. Gärtner TP, Flach A, Kowalczyk A, Smola AJ (2002) Multi-instance kernels. In: ICML, vol 2, no. 3, p 7
26. Wang X, Yan Y, Tang P, Bai X, Liu W (2018) Revisiting multiple instance neural networks. Pattern Recogn 74:15–24
27. Brand L, Baker LZ, Ellefsen C, Sargent J, Wang H (2021) A linear primal–dual multi-instance SVM for big data classifications. In: 2021 IEEE international conference on data mining (ICDM), pp 21–30

28. Boyd S, Parikh N, Chu E (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc
29. Lium L, Han Z (2015) Multi-block admm for big data optimization in smart grid. In: 2015 International conference on computing, networking and communications (ICNC). IEEE, pp 556–561
30. Hong M, Luo Z-Q (2017) On the linear convergence of the alternating direction method of multipliers. Math Program 162(1–2):165–199
31. Nie F, Huang Y, Wang X, Huang H (2014) New primal SVM solver with linear computational cost for big data classifications. In: Proceedings of the 31st international conference on international conference on machine learning, vol 32, pp II–505
32. Platt J (1998) Sequential minimal optimization: a fast algorithm for training support vector machines
33. Dogan Ü, Glasmachers T, Igel C (2016) A unified view on multi-class support vector classification. J Mach Learn Res 17(45):1–32
34. Weston J, Watkins C et al (1999) Support vector machines for multi-class pattern recognition. Esann 99:219–224
35. Wang J, Zhao L (2017) Nonconvex generalization of admm for nonlinear equality constrained problems. arXiv preprint arXiv:1705.03412
36. Welling M (2013) Kernel ridge regression. Max Welling's classnotes in machine learning, pp 1–3
37. Shi X, Xing F, Xie Y, Zhang Z, Cui L, Yang L (2020) Loss-based attention for deep multiple instance learning. Proc AAAI Conf Artif Intell 34(04):5742–5749
38. Blaom AD, Kiraly F, Lienart T, Simillides Y, Arenas D, Vollmer SJ (2020) MLJ: a Julia package for composable machine learning. J Open Source Softw 5(55):2704
39. Doran G, Ray S (2014) A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. Mach Learn 97(1–2):79–102
40. Borkowski AA, Bui MM, Thomas LB, Wilson CP, DeLand LA, Mastorides SM (2019) Lung and colon cancer histopathological image dataset (lc25000). arXiv preprint arXiv:1912.12142
41. Chakrabarti K, Mehrotra S (1999) The hybrid tree: an index structure for high dimensional feature spaces. In: Proceedings 15th international conference on data engineering (Cat. No. 99CB36337). IEEE, pp 440–447
42. Rahmani R, Goldman SA, Zhang H, Krettek J, Fritts JE (2005) Localized content based image retrieval. In: Proceedings of the 7th ACM SIGMM international workshop on multimedia information retrieval, pp 227–236
43. Zheng L, Yang Y, Tian Q (2017) Sift meets CNN: a decade survey of instance retrieval. IEEE Trans Pattern Anal Mach Intell 40(5):1224–1244
44. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
45. Zitnick, CL, Dollár P (2014) Edge boxes: locating object proposals from edges. In: European conference on computer vision. Springer, pp 391–405
46. Krizhevsky A (2014) One weird trick for parallelizing convolutional neural networks. arXiv preprint arXiv:1404.5997
47. Chang KW, Hsieh CJ, Lin CJ (2008) Coordinate descent method for large-scale l2-loss linear support vector machines. J Mach Learn Res 9:7
48. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171
49. Kong S, Fowlkes CC (2018) Recurrent pixel embedding for instance grouping. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9018–9028
50. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and patter recognition, pp 770–778
51. Cai X, Wang H, Huang H, Ding C (2012) Joint stage recognition and anatomical annotation of drosophila gene expression patterns. Bioinformatics 28(12):i16–i24
52. Petersen RC, Aisen P, Beckett LA, Donohue M, Gamst A, Harvey DJ, Jack C, Jagust W, Shaw L, Toga A et al (2010) Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization. Neurology 74(3):201–209
53. Risacher SL, Shen L, West JD, Kim S, McDonald BC, Beckett LA, Harvey DJ, Jack CR Jr, Weiner MW, Saykin AJ et al (2010) Longitudinal MRI atrophy biomarkers: relationship to conversion in the ADNI cohort. Neurobiol Aging 31(8):1401–1418
54. Bak TH, O'Donovan DG, Xuereb JH, Boniface S, Hodges JR (2001) Selective impairment of verb processing associated with pathological changes in brodmann areas 44 and 45 in the motor neurone disease-dementia-aphasia syndrome. Brain 124(1):103–120

55. Carmichael OT, Kuller LH, Lopez OL, Thompson PM, Dutton RA, Lu A, Lee SE, Lee JY, Aizenstein HJ, Meltzer CC et al (2007) Ventricular volume and dementia progression in the cardiovascular health study. Neurobiol Aging 28(3):389–397
56. Jack C, Shiung M, Gunter J, O'brien P, Weigand S, Knopman DS, Boeve BF, Ivnik R, Smith G, Cha R et al (2004) Comparison of different MRI brain atrophy rate measures with clinical disease progression in ad. Neurology 62(4):591–600
57. Apostolova LG, Beyer M, Green AE, Hwang KS, Morra JH, Chou Y-Y, Avedissian C, Aarsland D, Janvin CC, Larsen JP et al (2010) Hippocampal, caudate, and ventricular changes in Parkinson's disease with and without dementia. Mov Disord 25(6):687–695
58. Van der Werf YD, Witter MP, Uylings HB, Jolles J (2000) Neuropsychology of infarctions in the thalamus: a review. Neuropsychologia 38(5):613–627
59. Mu Y, Gage FH (2011) Adult hippocampal neurogenesis and its role in Alzheimer's disease. Mol Neurodegener 6(1):85
60. Poulin SP, Dautoff R, Morris JC, Barrett LF, Dickerson BC, Initiative ADN et al (2011) Amygdala atrophy is prominent in early Alzheimer's disease and relates to symptom severity. Psychiatry Res Neuroimaging 194(1):7–13

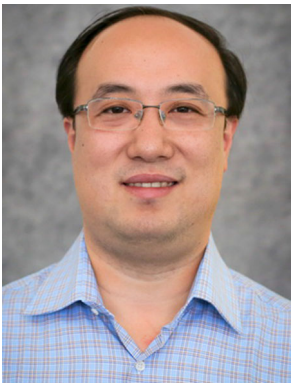
Lodewijk Brand received the Ph.D. degree in Computer Science from the Colorado School of Mines in 2021 in Golden, CO. Before that, he received the Bachelor's degree from Colorado College in Colorado Springs, CO. Currently, he is a research Scientist at the Naval Information Warfare Center, Atlantic in Charleston, SC. His research interests include the development of scalable and distributed machine learning algorithms applied to problems in bioinformatics, signal processing, and computer vision.

Hoon Seo received Master's degree in Computer Science from Colorado School of Mines in 2020. Before that, he received the Bachelor's degree from KyungHee University, Repulic of Korea in 2018. Since 2020 he has been a Ph.D. student at the Department of Computer Science of Colorado School of Mines. His research interests include machine learning and optimization, as well as their applications in bioinformatics, computer vision, and mining.

Lauren Zoe Baker is a senior student pursuing a double degree in Computer Science and Applied Math/Statistics at Colorado School of Mines. She has been participating in undergraduate research for four years since she was a freshman undergraduate student. Her research has been with the Machine Learning, Informatics, and Data Science lab (MINDS@Mines), headed by Dr. Hua Wang. Within the lab, her work has primarily focused on applying machine learning techniques to understand and analyze disease and human health. Zoe was named a Barry Goldwater scholar in 2020 for her research accomplishments. In the future, Zoe hopes to pursue a Ph.D. in Computer Science.

Carla Ellefsen is a first-year student at the Colorado School of Mines. She is currently studying for a Bachelor's degree in computer science with a focus in data science. Her research experience involves applying multi-instance learning to natural-scene imaging data and neural networks to traffic flow prediction. She hopes to continue her education and research in computer science and related fields in the future.

Jackson Sargent will receive a BSE in Computer Science from the University of Michigan College of Engineering in May of 2022. In his time at the University of Michigan, he has engaged in research focusing on machine learning applications in fields such as computational social science, natural language processing, computer vision, and bias and fairness in machine learning.



Hua Wang received a Ph.D. degree in Computer Science from the University of Texas at Arlington in 2012. Before that, he received a Bachelor's degree from Tsinghua University, China in 1999 and a Master's degree from Nanyang Technological University, Singapore in 2003. He is currently a Professor at the Department of Computer Science of Colorado School of Mines. Before this, he was an Assistant Professor from 2012 to 2018 and an Associate Professor from 2018 to 2023 in the same department. His research interests include machine learning and data mining, as well as their applications in medical image computing, health informatics, bioinformatics, and computer vision.