A Linear Primal-Dual Multi-Instance SVM for Big Data Classifications

Lodewijk Brand
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
lbrand@mines.edu

Lauren Zoe Baker
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
laurenzoebaker@mines.edu

Carla Ellefsen
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
cellefsen@mines.edu

Jackson Sargent
Computer Science and Engineering Division
University of Michigan
Ann Arbor, Michigan, USA
jacsarge@umich.edu

Hua Wang
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
huawangcs@gmail.com

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00012

***Abstract*—Multi-instance learning (MIL) is an area of machine learning that handles data that is organized into sets of instances known as bags. Traditionally, MIL is used in the supervised-learning setting and is able to classify bags which can contain any number of instances. This property allows MIL to be naturally applied to solve the problems in a wide variety of real-world applications from computer vision to healthcare. 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 (pdMISVM) derivation and implementation 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 (ADMM). The approach presented in this work is able to scale to large-scale data since it avoids iteratively solving quadratic programming problems which are generally used to optimize MIL algorithms based on SVMs. In addition, we modify our derivation to include an additional optimization designed to avoid solving a least-squares problem during our algorithm; this optimization increases the utility of our approach to handle a large number of features as well as bags. Finally, we apply our approach to synthetic and real-world multi-instance datasets to illustrate the scalability, promising predictive performance, and interpretability of our proposed method. We end our discussion with an extension of our approach to handle non-linear decision boundaries. Code and data for our methods are available online at: https://github.com/minds-mines/pdMISVM.jl.**

***Index Terms*—multi-instance learning, support vector machine, alternating direction method of multipliers, scalability**

I. 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 the instance-level. This is also known as *weakly supervised* learning in the literature. Algorithms that adhere to this type 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, patients can be represented by a collection of medical records, to name a few. However, 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 the data directly. Significant research efforts have been made to design algorithms that can handle this type of data.

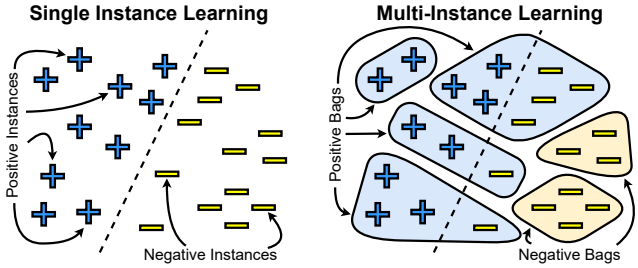

Fig. 1. An illustrative comparison of the single instance and multi-instance learning paradigms. Algorithms that operate on multi-instance data must contend with the fact that instances are rarely individually-labeled. Instead, labels are generally provided at the *bag*-level. Thus, the goal of a multi-instance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

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 ranging from drug activity prediction [9], content-based image retrieval [10], medical image analysis [11], document classification [12], among many other application areas [13]. Recently, deep learning-based MIL methods [14]–[16] have also been proposed to handle multi-instance data. While these methods have demonstrated their effectiveness in solving a variety of real-world problems by multi-instance learning, their limitations have also been discussed in literature [17], [18]. For example, a recent survey

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00012

paper [18] 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 [17] and requires that new algorithms 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 that rely on instance-level information, are necessary for further adoption.

In this work, we focus specifically on scaling SVM-based MIL algorithms, as they have shown consistent performance and can be further extended to non-linear decision boundaries via the kernel trick. Popular SVM-based MIL approaches such as miSVM/MISVM [1], NSK [19], 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 [20]. 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.

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.

II. Methods

In this section we begin with a sketch for the steps required for the standard multi-instance SVM (MISVM) derivation initially presented by Andrews *et al*. [1]. Then, following the multi-block ADMM framework [21]–[23], we construct the augmented Lagrangian which will be used to derive the solution to the proposed pdMISVM method; this is followed by a step-by-step derivation to optimize the proposed objective. Finally, we extend our approach to handle a large number of features through an application of the optimal line search method [24].

A. 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 $\mathbf{m}^i$ and $\mathbf{m}_j$, respectively. Similarly, m_j^i 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.

B. Extending the MISVM to K-classes

In the binary multi-instance classification problem, the MIL algorithm is presented with a collection of bags and labels represented by the set $\{\mathbf{X}_i, y_i\} : i \in 1, \ldots, N$ where $y_i \in \{-1, 1\}$, $\mathbf{X}_i \in \mathbb{R}^{d \times n_i}$ 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 Andrew's *et al*. [1] 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

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

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

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

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) are incorporated via a Lagrangian function

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

and is solved with respect to the dual variables (α_i) using off-the-shelf quadratic programming solvers or heuristic algorithms like sequential minimal optimization [25] which takes advantage of a limited number of support vectors. Although the MISVM formulation from Andrews *et al*. 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

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

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 [26] 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 [27] MISVM extension to Eq. (2)

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

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 [26] 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 [23] to optimize Eq. (5).

C. A Primal-Dual Multi-Instance SVM

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

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

which is difficult to solve given the coupling across $\mathbf{w}_k$, b_m, and the $\max(\cdot)$ operations. Following the multi-block ADMM approach we introduce the following constraints, inspired by [24], [28], and rewrite Eq. (6) as

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

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

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

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 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.

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

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

where i' indicates the column blocks in $\mathbf{X}$ (and the corresponding columns of $\mathbf{U}$ and $\boldsymbol{\Xi}$) 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

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

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, setting the result equal to zero, gives the update

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

E update. Dropping terms from Eq. (8), that do not contain $\mathbf{E}$ and decoupling element-wise gives $K \times N$ problems

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

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 in three cases

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

Q & R update. Keeping only terms with $\mathbf{Q}$ in Eq. (8) and decoupling element-wise gives $K \times N$ problems

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

Taking the derivative of Eq. (14) with respect to q_i^m, setting the result equal to zero, and solving for q_i^m gives the 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

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

T & U update. Keeping terms in Eq. (8) containing $\mathbf{T}$ and decoupling across K and N gives the following

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

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

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

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 $t_{i,j}^m$, setting the result equal to zero, and solving for $t_{i,j}^m$, gives the 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 $\mathbf{U}$, giving

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

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 1.

D. Scaling to a Large Number of Features

Although the updates derived in Section II-C provide a suitable algorithm as the number of bags increase, the least-squares solver used to update $\mathbf{W}$ in Eq. (10) has computational complexity $O\left((n_1 + \cdots + n_N) \cdot d^2\right)$ and 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. To handle this issue, we propose an alternative optimal line search method [24] to update $\mathbf{W}$ instead.

W Update, 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 minimization

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

in terms of s_m instead of $\mathbf{w}_m$. Differentiating Eq. (22) with respect to s_m, setting the result equal to zero, and solving for s_m gives

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

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 $\hat{\mathbf{u}}_{i'}^{m'} = \mathbf{u}_{i'}^{m'} - \mathbf{w}_m^T \mathbf{X}_{i'} - \mathbf{1} b_m + \xi_{i'}^{m'}/\mu$. Recognizing that the denominator of Eq. (23) is equivalent to

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

allows for Eq. (23) to be calculated efficiently in $O\left((n_1 + \cdots + n_N) \cdot d\right)$ time. Combining, Eq. (21) and Eq. (23) can then be used to 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 1 to extend our method to handle a large number of features.

III. Experiments

In this section we explore the performance of our exact and inexact pdMISVM implementations. We first test our method against an array of standard MIL benchmarks to explore how our implementations compare against other MIL methods.

Model	MUSK-2			Elephant			Fox			Tiger			MNIST-Bags		
	ACC	BACC	TT	ACC	BACC	TT	ACC	BACC	TT	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	0.575±0.050	0.579±0.064	1.77	0.695±0.070	0.697±0.053	0.47	0.420±0.156	0.495±0.078	0.88
miSVM	0.657±0.136	0.696±0.103	278.07	0.790±0.043	0.784±0.036	13.72	*0.595±0.087*	*0.602±0.089*	28.03	0.760±0.082	0.762±0.083	12.25	0.401±0.104	0.502±0.074	1.50
MISVM	0.794±0.081	0.768±0.107	252.16	0.840±0.052	0.844±0.045	21.11	0.570±0.037	0.569±0.044	31.92	0.790±0.091	0.792±0.092	17.09	0.499±0.099	0.489±0.073	8.99
NSK	0.814±0.058	0.808±0.063	1.38	*0.854±0.081*	*0.855±0.072*	1.55	0.535±0.099	0.539±0.102	0.99	0.795±0.113	0.787±0.118	0.87	*0.640±0.166*	*0.641±0.167*	1.21
sMIL	0.725±0.127	0.732±0.128	21.09	0.500±0.057	0.500±0.000	1.44	0.529±0.115	0.529±0.086	1.25	0.670±0.065	0.660±0.068	0.79	0.609±0.069	0.510±0.073	**0.21**
sbMIL	0.657±0.141	0.592±0.130	27.82	0.665±0.036	0.663±0.055	2.13	0.599±0.093	0.587±0.090	3.42	0.626±0.064	0.627±0.046	1.64	0.539±0.066	0.501±0.066	0.63
miNet	*0.853±0.104*	*0.847±0.121*	17.79	0.844±0.081	0.849±0.068	23.62	0.561±0.056	0.563±0.060	23.85	*0.805±0.067*	**0.805±0.058**	24.33	0.608±0.137	0.556±0.109	15.05
MINet	**0.882±0.064**	**0.864±0.070**	19.61	**0.860±0.053**	**0.864±0.050**	23.88	0.585±0.120	0.585±0.089	24.20	**0.820±0.083**	*0.805±0.087*	24.48	0.591±0.139	0.511±0.075	15.35
Ours	0.794±0.152	0.802±0.160	**0.72**	0.825±0.053	0.822±0.062	*0.23*	0.590±0.103	0.586±0.097	*0.30*	0.795±0.084	0.798±0.081	*0.21*	0.616±0.180	0.582±0.135	3.30
Ours (inexact)	0.804±0.080	0.811±0.085	*0.83*	0.830±0.047	0.837±0.042	**0.14**	**0.640±0.047**	**0.647±0.045**	**0.20**	0.780±0.063	0.780±0.064	**0.16**	**0.672±0.105**	**0.645±0.100**	*0.45*

TABLE I

CLASSIFICATION PERFORMANCE AND TRAIN-TIME (SECONDS) OF OUR METHOD AND EIGHT OTHER MIL LEARNING METHODS ON FIVE BENCHMARK DATASETS. THE REPORTED ACCURACY AND STANDARD DEVIATIONS ARE CALCULATED ACROSS TEN SIX-FOLD CROSS VALIDATION EXPERIMENTS. BEST RESULTS ARE MARKED **IN BOLD**, SECOND BEST *in italics*.

Algorithm 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 $(b_1, \ldots, b_K) \in \mathbf{b}$.

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. Finally, we present an extension of our primal-dual derivation to handle arbitrary kernel functions. We experimentally verify this extended derivation on synthetic data and identify a limitation of our approach to be addressed in future work.

A. Settings & Data

We compare our method against eight 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 [19], 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 [20], based on deep neural-network (DNN) architectures.

These methods are compared against the proposed pdMISVM (Ours) method, and the inexact variation, described in Algorithm 1. The grid search and performance calculations for each method-dataset pair are conducted using the MLJ library [29] 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*. [30] while the DNN methods are implemented using the code[3] provided as a companion to the paper by Wang *et al*. [20]. 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, and Tiger [1] datasets are standard small scale MIL evaluation datasets and are widely cited in the MIL literature. The MUSK-2 dataset is designed to classify chemical compounds as either "musk" or "non-

[1] https://github.com/minds-mines/pdMISVM.jl
[2] https://github.com/garydoranjr/misvm
[3] https://github.com/yanyongluan/MINNs

Model	SIVAL-3			SIVAL-5			SIVAL-10			SIVAL-25			SIVAL-25-deep		
	ACC	BACC	TT	ACC	BACC	TT	ACC	BACC	TT	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	–	–	T/O	–	–	T/O	–	–	T/O
miSVM	0.856±0.062	0.858±0.053	469.83	–	–	T/O	–	–	T/O	–	–	T/O	–	–	T/O
MISVM	*0.909±0.058*	*0.908±0.066*	457.65	–	–	T/O	–	–	T/O	–	–	T/O	–	–	T/O
NSK	0.633±0.070	0.634±0.085	1.44	0.547±0.088	0.551±0.089	6.27	0.462±0.042	0.469±0.053	50.2	–	–	T/O	–	–	T/O
sMIL	0.544±0.119	0.541±0.121	7.52	0.443±0.112	0.441±0.131	68.51	–	–	T/O	–	–	T/O	–	–	T/O
sbMIL	0.767±0.123	0.775±0.128	44.08	–	–	T/O	–	–	T/O	–	–	T/O	–	–	T/O
miNet	0.644±0.066	0.672±0.038	72.49	0.203±0.104	0.240±0.101	72.49	0.100±0.020	0.102±0.024	203.93	–	–	T/O	–	–	T/O
MINet	0.589±0.066	0.558±0.043	78.91	0.253±0.043	0.261±0.063	78.91	0.135±0.037	0.135±0.019	226.43	–	–	T/O	–	–	T/O
Ours	**0.911±0.058**	**0.917±0.059**	**0.48**	**0.840±0.051**	**0.846±0.047**	**1.02**	**0.732±0.057**	**0.742±0.057**	**7.88**	**0.487±0.041**	**0.489±0.039**	**24.01**	–	–	T/O
Ours (inexact)	0.767±0.067	0.763±0.069	*0.74*	*0.730±0.065*	*0.733±0.057*	*1.67*	*0.577±0.061*	*0.581±0.055*	*13.57*	*0.388±0.045*	*0.383±0.044*	*36.17*	**0.888±0.04**	**0.887±0.036**	**977.66**

TABLE II

CLASSIFICATION AND TRAIN-TIME (SECONDS) PERFORMANCE OF OUR METHOD AND EIGHT OTHER MIL LEARNING METHODS ON VARIANTS OF THE SIVAL DATASET ACROSS A DIFFERENT NUMBER OF CLASSES AND PREPROCESSING PIPELINES. THE REPORTED ACCURACY AND STANDARD DEVIATIONS ARE CALCULATED ACROSS TEN SIX-FOLD CROSS VALIDATION EXPERIMENTS. BEST RESULTS ARE MARKED **IN BOLD**, SECOND BEST *in italics*.

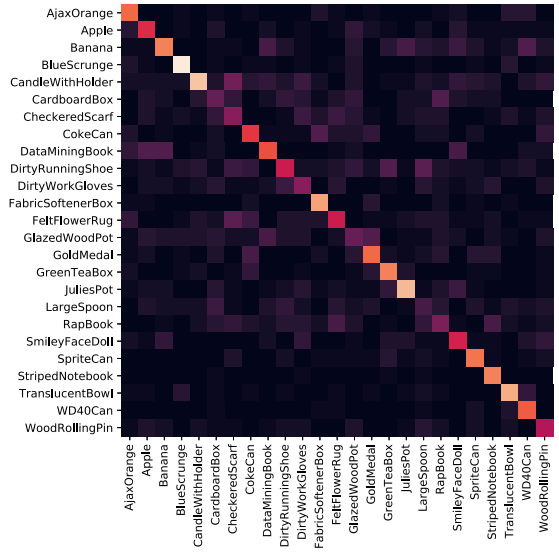

Fig. 2. Confusion matrix of the pdMISVM tested on the original SIVAL-25 dataset with 30 features per-instance. Results are derived from a six-fold cross-validation experiment across all 1,500 bags.

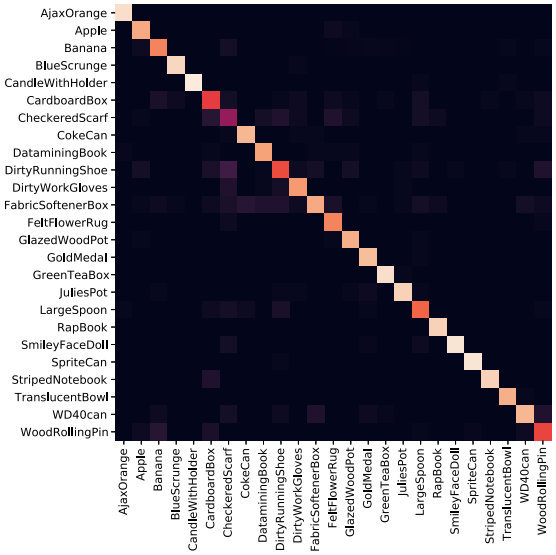

Fig. 3. Confusion matrix of the inexact pdMISVM approach tested on the SIVAL-25-deep dataset created from the patch-wise application of a convolutional neural network as a pre-processing step. Results are derived from a six-fold cross-validation experiment across all 1,463 bags.

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 [31] and each contain 100 positive and 100 negative bags with 143 non-zero features per instance.

The MNIST-bags [15] 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 1,500 bags. In this work we use the processed dataset provided in the initial work of Rahmani *et al.* [32] and create a novel 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, and SIVAL-25 datasets each containing three, five, ten, and twenty-five classes from the SIVAL dataset, respectively.

In addition, we construct the "SIVAL-25-deep" dataset, which is inspired by the "hybrid" approach detailed by Zheng *et al.* [33] which investigates the ongoing shift from SIFT-based descriptors [34] to convolutional neural networks for generating image descriptors. To create this multi-instance dataset we extract patches from the raw

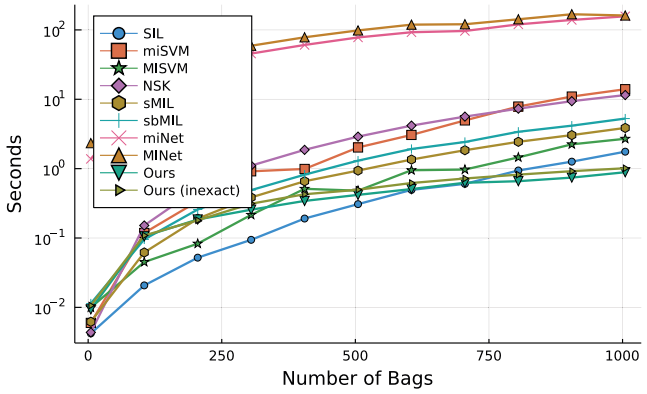

Fig. 4. Time to train our method and other MIL methods on synthetic multi-instance data where the number of bags increase. Both the exact and inexact methods end up training faster than the competing methods once the number of synthetic bags is greater than eight-hundred.

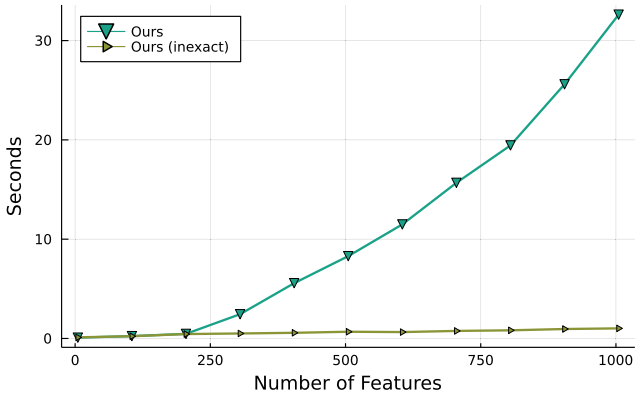

Fig. 5. Elapsed time to train the exact and inexact methods on synthetic multi-instance data as the number of features is varied. As expected, the exact approach performs poorly as the number of features increases but the inexact method continues to scale linearly.

SIVAL images using the EdgeBox [35] proposal generator (`eta=0.2`, `minScore=0.04`, `maxBoxes=200`) provided in the OpenCV library.[4] These extracted patches are fed into a pre-trained AlexNet [36] 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, could 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.

B. Classification Performance

In Tables I and II we provide the classification performance of our approach compared against the other MIL algorithms. 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 six-fold cross validation experiments. We can see from Table I 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 implementation of our approach outperforms all other methods on both the Fox and MNIST-Bags datasets. In Table II the exact method 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 method 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 Figures 2 and 3. It's clear from these results that both the exact and inexact methods are capable of providing competitive performance results on a variety of multi-instance datasets.

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

C. Bag/Feature Scalability

The *key contribution* of this work is that the derived algorithms described in Section II-C and Section II-D scale to large datasets. This can clearly be seen in the SIVAL-25 column of Table II 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 Figure 4 we report training time results on a synthetic multi-instance dataset where we increase the number of bags as described in Section III-A. Our approach scales well with respect to the number of bags which illustrates the importance of our primal-dual derivation. 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 between the exact and inexact methods using synthetic data where the number of features are increased to see if our inexact variation provides improved runtime performance. We can see in Figure 5 that the proposed inexact method significantly reduces the training time of our approach as the number of features increase.

D. 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

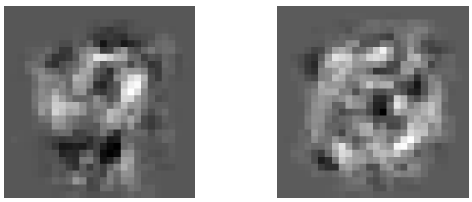

Fig. 6. Learned class-specific hyperplanes of the pdMISVM method on the MNIST-bags dataset plotted in a 28×28 grid. **Left:** Learned coefficients for predicting whether a bag contains the MNIST handwritten digit '9'. **Right:** The learned coefficients for predicting whether a bag *does not* contain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

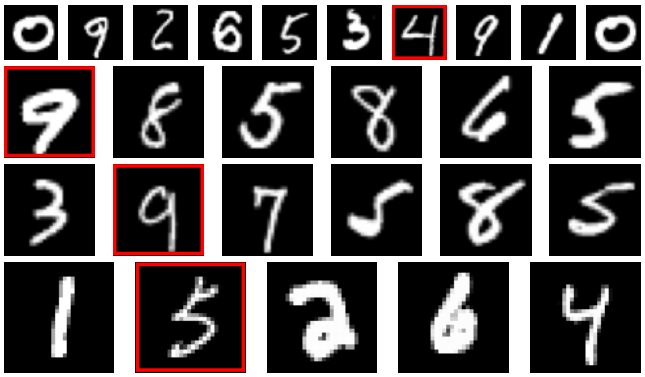

Fig. 7. Instance identification results on the first five testing bags of our method on the MNIST-bags dataset with the detectors in Figure 6. Our approach correctly classifies the first, second and third bags. Although the first bag is classified correctly the "9"s are not properly identified.

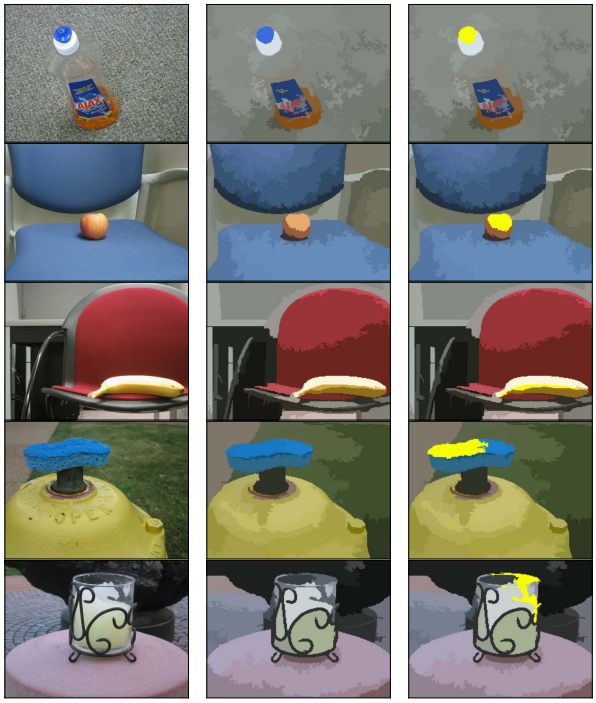

Fig. 8. Instance identification on the SIVAL-25 dataset across different classes. In each set of three pictures the leftmost is the original image, the middle shows the bag of patches extracted by the original authors [10], and the final image highlights the patch identified by our approach for classifying the image.

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 Figure 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 Figure 7. On the left-hand side of Figure 6, we can see that our method can roughly detect the loop at the top of the '9' although it is clear from this interpretation that the 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 Figure 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 Figure 8, we provide image patches identified by our approach on images chosen from the SIVAL-25 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 Figure 9 we present the neural-network embedded patches extracted via the EdgeBox detection algorithm and the identified patches. We can clearly see in Figure 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 Figures 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

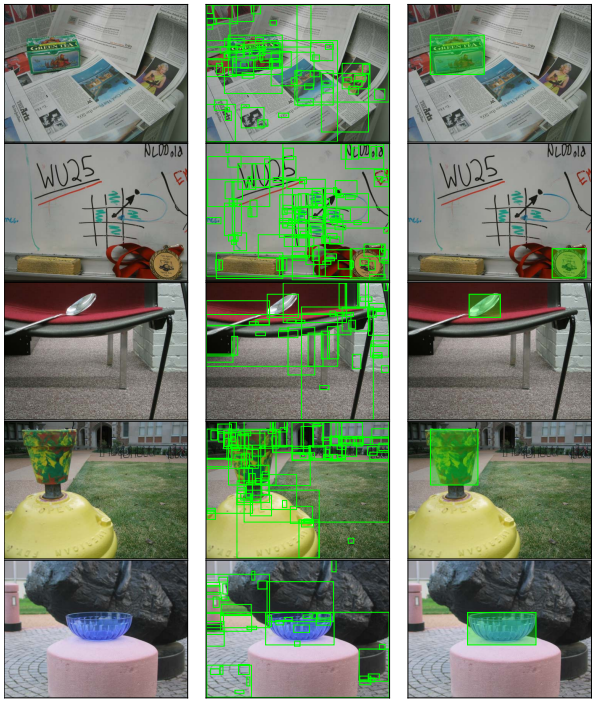

Fig. 9. Instance identification on the SIVAL-25-deep dataset across different classes. In each set of three pictures the leftmost is the original image, the middle shows the bag of patches extracted by the EdgeBox detector [35], and the final image highlights the patch identified by our approach for classifying the image.

object proposal methods [37], [38], neural architectures [39], and applications [12], [13], [40].

E. A Kernel Extension

Although the proposed pdMISVM formulation and implementations are shown to work well on a variety of MIL datasets, they are limited to classifying data that is linearly separable. In order for our method to be able to handle nonlinear decision boundaries, it requires the addition of a kernel. To this end, we briefly sketch out a kernel extension to the $\mathbf{W}$ variable update proposed in Section II-C. We verify this addition in an experiment with synthetic data that can only be correctly classified by a nonlinear method.

W update (with kernel) Starting with Eq. (9) and replacing all columns of $\mathbf{X}_i$ with feature vectors calculated by an arbitrary kernel function $\phi(\mathbf{X}_i) = \mathbf{\Phi}_i$, taking its derivative with respect to $\mathbf{w}_m$, and solving for $\mathbf{w}_m$ gives

$$\mathbf{w}_m^T = \Big(\left[(\mathbf{t}^m - \mathbf{1}b_m + \boldsymbol{\theta}^m/\mu)\, \mathbf{\Phi}^T \right] + \textstyle\sum_{m'=1}^{K} \left[(\mathbf{u}_{\prime}^{m'} - \mathbf{1}b_m + \boldsymbol{\xi}_{\prime}^{m'}/\mu) \mathbf{\Phi}_{\prime}^T \right] \Big) * \left(\mathbf{I}/\mu + \mathbf{\Phi}\mathbf{\Phi}^T + K\mathbf{\Phi}_{\prime}\mathbf{\Phi}_{\prime}^T \right)^{-1} \tag{26}$$

where the $\mathbf{\Phi} = [\mathbf{\Phi}_1 \ \ldots \mathbf{\Phi}_N]$ and $\mathbf{\Phi}_{\prime} = [\mathbf{\Phi}_{1'} \ \ldots \mathbf{\Phi}_{N'}]$ contains the N' column blocks from $\mathbf{\Phi}$ that belong to the m-th class. Equation (26) can be written in matrix form

$$\mathbf{w}_m^T = \mathbf{s}^m \mathbf{D} \hat{\mathbf{\Phi}}^T * \left(\mathbf{I}/\mu + \hat{\mathbf{\Phi}} \mathbf{D} \hat{\mathbf{\Phi}}^T \right)^{-1} , \tag{27}$$

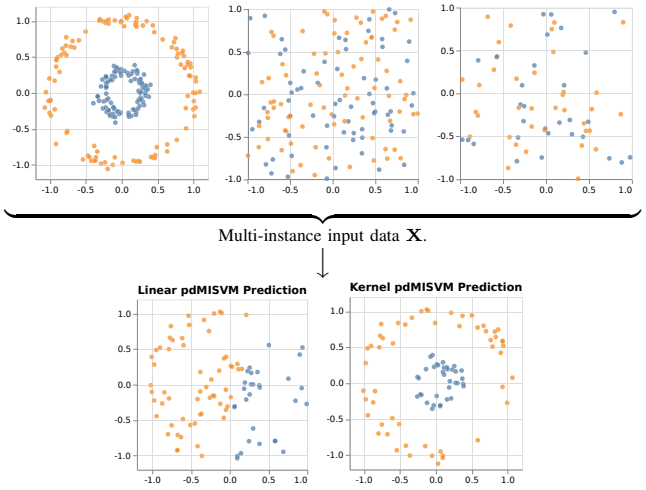

Fig. 10. Linear and kernel (radial basis function) pdMISVM predictions on synthetic multi-instance data. Each bag in the training dataset $\mathbf{X}$ has up to three instances, where only the first instance determines the correct classification. The kernel extension of our approach is able to correctly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

where $\mathbf{s}^m = [\mathbf{t}^m - \mathbf{1}b_m + \boldsymbol{\theta}^m/\mu \quad 1/K \sum_{m'=1}^{K} (\mathbf{u}_{\prime}^{m'} - \mathbf{1}b_m + \boldsymbol{\xi}_{\prime}^{m'}/\mu)]$, $\mathbf{D} = [\mathbf{I}\ \mathbf{0};\ \mathbf{0}\ K\mathbf{I}]$ 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. (27). In order to solve this issue we use the following trick [41]

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

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

$$\mathbf{w}_m^T = \mathbf{s}_m(\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}} + \mathbf{D}^{-1}/\mu)^{-1}\hat{\mathbf{\Phi}}^T \ . \tag{28}$$

The updated $\mathbf{w}_m$ in Eq. (28) can then be used to update $\mathbf{w}_m^T\phi(\mathbf{X}_i)$ and $\|\mathbf{w}_m\|_2^2 = \mathbf{tr}\left(\mathbf{w}_m^T\mathbf{w}_m\right)$ directly, as the kernel function occurs as an inner product in both cases and can be computed. We implement the kernel version of the pdMISVM method and compare it against the linear version in Figure 10; we can see this extension successfully extends our approach to correctly classify data that is not linearly separable. Unfortunately, the update in Eq. (28) is computationally expensive as it requires a least squares calculation that scales quadratically with respect to the total number of bags in the training dataset. We plan to address this limitation in our future work.

IV. 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 analyses that investigate the interpretability of our method on benchmark multi-instance datasets and develop an extension to the SIVAL dataset as part of this study. Finally, using the same primal-dual framework, we derive and implement a kernel extension of our approach that is able to learn non-linear decision boundaries on synthetic multi-instance data and identify areas for future work.

ACKNOWLEDGMENT

Corresponding author: Hua Wang (huawangcs@gmail.com).

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. Lodewijk Brand was supported in part by the Department of Defense (DoD) Science, Mathematics And Research for Transformation (SMART) Scholarship-for-Service program.

REFERENCES

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning." in *NIPS*, vol. 2. Citeseer, 2002, pp. 561–568.

[2] R. C. Bunescu and R. J. Mooney, "Multiple instance learning for sparse positive bags," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 105–112.

[3] H. Wang, F. Nie, and H. Huang, "Learning instance specific distance for multi-instance classification," in *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[4] H. Wang, H. Huang, F. Kamangar, F. Nie, and C. Ding, "Maximum margin multi-instance learning," *Advances in neural information processing systems*, vol. 24, pp. 1–9, 2011.

[5] H. Wang, F. Nie, and H. Huang, "Robust and discriminative distance for multi-instance learning," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2919–2924.

[6] H. Wang, C. Deng, H. Zhang, X. Gao, and H. Huang, "Drosophila gene expression pattern annotations via multi-instance biological relevance learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[7] K. Liu, H. Wang, F. Nie, and H. Zhang, "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*, 2018, pp. 7727–7735.

[8] L. Brand, L. Z. Baker, and H. Wang, "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*, 2021, pp. 1–6.

[9] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[10] Z.-H. Zhou, X.-B. Xue, and Y. Jiang, "Locating regions of interest in cbir with multi-instance learning techniques," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2005, pp. 92–101.

[11] V. Cheplygina, M. de Bruijne, and J. P. Pluim, "Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Medical image analysis*, vol. 54, pp. 280–296, 2019.

[12] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 2007, pp. 1289–1296.

[13] Y. Xu, J.-Y. Zhu, I. Eric, C. Chang, M. Lai, and Z. Tu, "Weakly supervised histopathology cancer image segmentation and classification," *Medical image analysis*, vol. 18, no. 3, pp. 591–604, 2014.

[14] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3460–3469.

[15] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International conference on machine learning*. PMLR, 2018, pp. 2127–2136.

[16] Y. Yan, X. Wang, X. Guo, J. Fang, W. Liu, and J. Huang, "Deep multi-instance learning with dynamic pooling," in *Asian Conference on Machine Learning*. PMLR, 2018, pp. 662–677.

[17] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 697–704.

[18] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognition*, vol. 77, pp. 329–353, 2018.

[19] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *ICML*, vol. 2, no. 3, 2002, p. 7.

[20] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognition*, vol. 74, pp. 15–24, 2018.

[21] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[22] L. Liu and Z. Han, "Multi-block admm for big data optimization in smart grid," in *2015 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2015, pp. 556–561.

[23] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, vol. 162, no. 1-2, pp. 165–199, 2017.

[24] F. Nie, Y. Huang, X. Wang, and H. Huang, "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-Volume 32*, 2014, pp. II–505.

[25] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[26] Ü. Dogan, T. Glasmachers, and C. Igel, "A unified view on multi-class support vector classification." *J. Mach. Learn. Res.*, vol. 17, no. 45, pp. 1–32, 2016.

[27] J. Weston, C. Watkins *et al.*, "Support vector machines for multi-class pattern recognition." in *Esann*, vol. 99, 1999, pp. 219–224.

[28] J. Wang and L. Zhao, "Nonconvex generalization of admm for nonlinear equality constrained problems," *arXiv preprint arXiv:1705.03412*, 2017.

[29] A. D. Blaom, F. Kiraly, T. Lienart, Y. Simillides, D. Arenas, and S. J. Vollmer, "MLJ: A julia package for composable machine learning," *Journal of Open Source Software*, vol. 5, no. 55, p. 2704, 2020.

[30] G. Doran and S. Ray, "A theoretical and empirical analysis of support vector machine methods for multiple-instance classification," *Machine learning*, vol. 97, no. 1-2, pp. 79–102, 2014.

[31] K. Chakrabarti and S. Mehrotra, "The hybrid tree: An index structure for high dimensional feature spaces," in *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*. IEEE, 1999, pp. 440–447.

[32] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts, "Localized content based image retrieval," in *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, 2005, pp. 227–236.

[33] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2017.

[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[35] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*. Springer, 2014, pp. 391–405.

[36] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv preprint arXiv:1404.5997*, 2014.

[37] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[38] S. Kong and C. C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9018–9028.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] X. Cai, H. Wang, H. Huang, and C. Ding, "Joint stage recognition and anatomical annotation of drosophila gene expression patterns," *Bioinformatics*, vol. 28, no. 12, pp. i16–i24, 2012.

[41] M. Welling, "Kernel ridge regression," *Max Welling's Classnotes in Machine Learning*, pp. 1–3, 2013.