

A Theoretical and Empirical Analysis of Support Vector Machine Methods for Multiple-Instance Classification

Gary Doran · Soumya Ray

Received: date / Accepted: date

Abstract The standard support vector machine (SVM) formulation, widely used for supervised learning, possesses several intuitive and desirable properties. In particular, it is convex and assigns zero loss to solutions if, and only if, they correspond to consistent classifying hyperplanes with some nonzero margin. The traditional SVM formulation has been heuristically extended to multiple-instance (MI) classification in various ways. In this work, we analyze several such algorithms and observe that all MI techniques lack at least one of the desirable properties above. Further, we show that this tradeoff is fundamental, stems from the topological properties of consistent classifying hyperplanes for MI data, and is related to the computational complexity of learning MI hyperplanes. We then study the empirical consequences of this three-way tradeoff in MI classification using a large group of algorithms and datasets. We find that the experimental observations generally support our theoretical results, and properties such as the labeling task (instance versus bag labeling) influence the effects of different tradeoffs.

Keywords multiple-instance learning · support vector machines · kernel methods

1 Introduction

A goal of drug activity prediction is to classify molecules as “active” or “inactive” depending on whether they bind to a target protein. Molecules may

G. Doran
Case Western Reserve University, 10900 Euclid Ave, Cleveland, OH 44106
E-mail: gary.doran@case.edu

S. Ray
Case Western Reserve University, 10900 Euclid Ave, Cleveland, OH 44106
E-mail: sray@case.edu

exist in multiple shapes, called *conformations*, in solution; however, the binding activity of individual conformations is often not observable. Therefore, if a molecule is active (i.e. binds to a target), that implies that *at least one* of its conformations is active. On the other hand, inactivity of a molecule means that *no* conformation binds. The multiple-instance (MI) learning framework was motivated by the above problem (Dietterich et al. 1997), and encodes this relationship between an observed label and a set of instances responsible for that label. In particular, a dataset is treated as a set of labeled *bags*, each of which contains several *instances*, which are feature vectors. If a bag is labeled positive, then at least one instance in the bag is positive. However, if a bag is negative, then every instance in the bag is negative. The learner has to produce a classifier that can accurately label new bags.

Numerous supervised learning algorithms have been extended to the MI classification (MIC) setting in prior work (Blockeel et al. 2005; Maron 1998; Ramon and De Raedt 2000; Xu and Frank 2004; Zhang and Goldman 2001; Zhou and Zhang 2002). In particular, kernel methods such as support vector machines (SVMs) have been modified to handle MI data, and are the focus of our analysis in this work (Andrews et al. 2003; Bunescu and Mooney 2007; Mangasarian and Wild 2008; Zhou and Xu 2007). The approaches we study adapt supervised SVMs to the MI setting by adding constraints to the optimization program or by constructing set kernels from instance kernels. Many of these approaches incorporate heuristic ideas. We seek to understand the theoretical basis for these approaches and study the empirical effects of the theoretical tradeoffs they make. In this work, we do not consider methods that construct new kernels directly on bags or create entirely different feature spaces to find classifiers (Chen et al. 2006; Tao et al. 2004; Zhou et al. 2009).

Conceptually, we consider the relationship between two spaces: the space of classifying *hyperplanes* and the space of feasible *solutions* allowed by an MI optimization program. The theoretical framework we investigate is based on three intuitive questions about these spaces and the relationship between them. First, consider the set of consistent hyperplanes for an MI classification problem (the classifying hyperplanes that fit the training bags without errors). Which MI SVMs encode a feasible solution with zero loss corresponding to each consistent hyperplane? We call approaches “complete” if this property holds across all MI classification problems. Second, consider the set of feasible solutions with zero loss in an MI SVM approach. Do all of these solutions correspond to consistent hyperplanes? If this is the case over all MI classification problems, we call the approach “sound.” Third, is the MI SVM approach and the corresponding set of feasible solutions convex? For standard supervised learning, the SVM program (Equation 1) is convex, sound, and complete by definition, since every zero-loss solution corresponds to a hyperplane that produces a correct labeling of the bags, and every hyperplane that produces a correct labeling has a corresponding zero-loss solution, for all supervised learning problems.

What is the importance of these properties? Soundness and completeness of the SVM ensure that loss is appropriately measured on solutions correspond-

ing to consistent hyperplanes, so that structural risk minimization (SRM) approaches will generalize to new data. We show that, surprisingly, for MI classification, *no approach can have all three properties*. In practice, we show that each MI SVM algorithm makes different tradeoffs: some sacrifice soundness, some completeness, and others convexity. As part of this work, we prove new results on the ability of normalized set kernels (Gärtner et al. 2002) (Equation 2) to separate MI data. Then, using topological arguments, we show that the tradeoff between soundness, completeness, and convexity is *fundamental* in MI learning, and we relate our result to the computational complexity of MI classification via hyperplanes. Finally, we carry out a detailed, large-scale empirical comparison using numerous datasets and algorithms to understand the practical implications of this tradeoff. Our experimental results are generally consistent with our theoretical analysis, and the results shed additional light on how the value of each tradeoff is affected by factors such as the labeling task (i.e. labeling instances versus labeling bags).

2 Consistency, Soundness, and Completeness in MI Classification

An MI problem (B, Y) is given by B , a list of bags, where each bag B_i is a set of instances $x_{ij} \in \mathcal{X}$, and Y is a list of corresponding labels with $Y_i \in \{-1, 1\}$. In our analysis, we describe the properties of optimization programs that attempt to learn a *classifying hyperplane* for MI data. Let \mathcal{H} be a space of classifying hyperplanes defined over the instance space \mathcal{X} . If the instance space $\mathcal{X} = \mathbb{R}^n$, then a hyperplane (w, b) with $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ labels instances via the function $f(x_{ij}) = \text{sign}(\langle w, x_{ij} \rangle + b)$. In the standard supervised setting, a hyperplane is *consistent* with a dataset if it correctly labels each instance. For an MI dataset, a consistent hyperplane correctly labels all bags; that is, it labels at least one instance in each positive bag as positive, and all instances in negative bags as negative:

Definition 1 (Consistency) A classifying hyperplane (w, b) is *consistent* with (B, Y) if for each $B_i \in B$:

$$\begin{cases} \exists x_{ij} \in B_i : \langle w, x_{ij} \rangle + b \geq 1 & \text{if } Y_i > 0 \\ \forall x_{ij} \in B_i : \langle w, x_{ij} \rangle + b \leq -1 & \text{if } Y_i < 0 \end{cases}$$

A consistent hyperplane separates bags, but not necessarily instances (e.g. an instance in a positive bag can lie within the margin as long as some other instance is classified as positive). We use $\mathcal{C} \subseteq \mathcal{H}$ to denote the set of all consistent hyperplanes. In some cases, we consider a stronger notion of consistency in which each instance must lie outside the margin of the classifier:

Definition 2 (Strong Consistency) A classifying hyperplane is *strongly consistent* with (B, Y) if it is consistent and for each instance x_{ij} , either $\langle w, x_{ij} \rangle + b \geq 1$ or $\langle w, x_{ij} \rangle + b \leq -1$.

Note that the choice of ± 1 labels for separating instances is arbitrary and equivalent to any other constant value, since we can always rescale (w, b) to achieve the same separation with labels of other values.

As in the supervised case, for some MI datasets there may not be a consistent hyperplane, or regularization might be favored over strict consistency. Therefore, an SRM approach is usually taken in which empirical risk (error on the training set) is minimized while the complexity of the selected model is controlled via regularization. Learning hyperplanes for MI data in this way entails solving an optimization problem of the form: $\min_{s \in \mathcal{F}} \lambda \Omega(s) + \ell(s)$, with a solution space \mathcal{S} , a feasible region $\mathcal{F} \subseteq \mathcal{S}$, a regularizer Ω , a nonnegative loss function ℓ , and a tradeoff parameter λ . We call this an MI optimization program if there exists a continuous function $\mu : \mathcal{S} \rightarrow \mathcal{H}$ that maps solutions in \mathcal{S} to classifying hyperplanes in \mathcal{H} .

Most MI SVMs used in practice are encompassed by this general formulation. Typically, the variables (w, b) are included directly in the optimization program, in which case μ simply projects a solution $s \in \mathcal{S}$ onto the dimensions corresponding to (w, b) . When a kernel function $k(\cdot, \cdot)$ is used, the space of hyperplane classifiers \mathcal{H} is a reproducing kernel Hilbert space (RKHS), and $s \in \mathcal{S}$ contains dual variables α_i such that a hyperplane $f \in \mathcal{H}$ is represented via $f = \sum_i \alpha_i \phi(x_i)$, where $\phi(\cdot)$ is the feature map corresponding to $k(\cdot, \cdot)$.

Recent results in learning theory show that the Vapnik–Chervonenkis (VC) dimension of an MI bag classifying hypothesis space is bounded in terms of the VC dimension of the underlying instance hypothesis space (Sabato and Tishby 2012). Therefore, given an appropriate kernel function, we can learn to classify bags using classifying hyperplanes via SRM strategies. However, the theoretical guarantees of risk minimization strategies require that ℓ provides an accurate assessment of empirical risk. Guarantees can generally not be made for algorithms whose loss functions do not reflect true empirical risk. In particular, if loss is appropriately measured, there should be a correspondence between the set of consistent hyperplanes $\mathcal{C} \subseteq \mathcal{H}$, and the set of zero-loss, feasible solutions $\mathcal{Z} = \{s \in \mathcal{F} : \ell(s) = 0\} \subseteq \mathcal{S}$. This correspondence between consistency and loss is fundamental (and usually trivially satisfied) in the supervised setting. However, as we show in Section 3, loss functions are used in several existing MI SVM formulations for which this correspondence *does not* hold.

We now define two desired properties of a loss function ℓ used in an MI optimization program. For any MI dataset, if a hyperplane (w, b) is consistent, then there should exist a corresponding solution s ($\mu(s) = (w, b)$) with zero loss, since the hyperplane properly “separates” bags in the sense described above. Conversely, every zero-loss solution s should correspond (again, via μ) to a consistent hyperplane (w, b) ; otherwise, the hyperplane misclassifies bags without penalty. We call these two properties “completeness” and “soundness.”

Definition 3 (Soundness) An MI optimization program is *sound* if for any dataset, all feasible, zero-loss solutions correspond to consistent hyperplanes; that is, if $\mu(\mathcal{Z}) \subseteq \mathcal{C}$.

Definition 4 (Completeness) An MI optimization program is *complete* if for any dataset, there exists a feasible, zero-loss solution corresponding to every consistent hyperplane; that is, if $\mathcal{C} \subseteq \mu(\mathcal{Z})$.

Therefore, sound and complete MI optimization programs have the property that $\mu(\mathcal{Z}) = \mathcal{C}$, or that the set of feasible, zero-loss solutions corresponds to the set of consistent hyperplanes. Note that these properties are quantified over all datasets, so although they focus on the datasets for which consistent hyperplanes exist, they apply to all MI optimization programs of the form described above. That is, these properties naturally apply to algorithms that handle nonseparable data with regularization. Though these properties in some sense “ignore” the behavior of such algorithms on nonseparable datasets, any algorithm that correctly measures empirical risk must *at least* do so correctly on separable datasets.

There is one additional desirable property of MI optimization programs, as in supervised learning: convexity. We define this formally below:

Definition 5 (Convexity) An optimization program is *convex* if for any dataset, and any $\lambda \geq 0$, \mathcal{F} is a convex set, and $\lambda\Omega(s) + \ell(s)$ is a convex function. With $\lambda = 0$, this implies that $\ell(s)$ is also a convex function.

The standard supervised SVM has the following formulation:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i, \quad \text{s.t.} \quad y_i (\langle w, \phi(x_i) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad (1)$$

where $\phi(\cdot)$ is the feature map corresponding to some kernel, $k(\cdot, \cdot)$. Under the supervised learning notion of consistency, this formulation is sound, complete, and convex. It is convex because the feasible region is a convex set, and the objective function is convex for any choice of C . It is sound, because if any solution corresponds to a hyperplane that does not fit the data, some ξ_i must be nonzero to satisfy the constraints, so the loss term $\sum_i \xi_i$ will be nonzero. Finally, it is complete, since for any solution corresponding to a consistent hyperplane, there exists a solution with each $\xi_i = 0$, which means that $\sum_i \xi_i = 0$. Having these three properties has contributed to making the SVM a popular and successful supervised learning approach. In our work, we study which MI SVM formulations have these properties.

The soundness and completeness concepts described above consider hyperplanes in the instance space (or in the feature space corresponding to an instance kernel). Other approaches use *set kernels* to map entire sets of instances into a feature space. For example, the MI kernel approach (Gärtner et al. 2002) computes a kernel k_{MI} via pairwise instance kernel k_{I} values between bag instances x in B_i and x' in B_j : $k_{\text{MI}}(B_i, B_j) = \sum \sum k_{\text{I}}^p(x, x')$, where

p is a positive integer parameter. The kernel can be normalized, for example by the length of the vector in the resulting feature space (*feature space normalization*), or by the number of instances (*averaging normalization*):

$$k_{\text{NSK}}(B_i, B_j) = \frac{\sum \sum k_{\text{I}}^p(x, x')}{(f_{\text{norm}}(B_i) \cdot f_{\text{norm}}(B_j))}, \quad (2)$$

where $f_{\text{norm}}(B_i)$ is a normalizer.

Since bags become individual points in a feature space, the set kernel corresponds to a space \mathcal{H}_{B} of hyperplanes in the space of *bags*, not instances. There is a separate classifying hyperplane space \mathcal{H}_{I} corresponding to the instance kernel k_{I} . As in Definition 1, there is a set \mathcal{C}_{I} of consistent solutions in the instance hyperplane space, and we say that k_{I} *separates* instances when $\mathcal{C}_{\text{I}} \neq \emptyset$ (there is some consistent hyperplane that separates instances). Similarly, the set of consistent hyperplanes \mathcal{C}_{B} in the bag hyperplane space \mathcal{H}_{B} are the hyperplanes that, in a supervised learning sense, separate bags by assigning the appropriate label to each bag in the dataset. A set kernel k_{MI} separates bags when $\mathcal{C}_{\text{B}} \neq \emptyset$.

Since the set kernel approach uses the standard supervised SVM quadratic program with a modified kernel, its loss function is not problematic ($\mu(\mathcal{Z}) = \mathcal{C}_{\text{B}}$); rather, the questions of soundness and completeness must now consider the relationship between consistent hyperplanes in the bag (\mathcal{C}_{B}) and instance (\mathcal{C}_{I}) hyperplane spaces, as done in prior work (Gärtner et al. 2002).

Definition 6 (Soundness for Set Kernels) A set kernel k_{NSK} is *sound* w.r.t. instance kernel k_{I} iff for any MI dataset, k_{NSK} separates bags only if k_{I} separates instances ($(\mathcal{C}_{\text{B}} \neq \emptyset) \implies (\mathcal{C}_{\text{I}} \neq \emptyset)$).

Definition 7 (Completeness for Set Kernels) A set kernel k_{NSK} is *complete* w.r.t. instance kernel k_{I} iff for any MI dataset, k_{NSK} separates bags if k_{I} separates instances ($(\mathcal{C}_{\text{I}} \neq \emptyset) \implies (\mathcal{C}_{\text{B}} \neq \emptyset)$).

For set kernels, soundness and completeness intuitively mean that it is possible to construct a set kernel from an instance kernel such that a zero-loss, consistent hyperplane exists in the set kernel feature space if and only if one exists in the original instance kernel feature space. Note that these notions do not require a bijection between \mathcal{C}_{B} and \mathcal{C}_{I} , because in general the feature maps corresponding to k_{I} and k_{NSK} can have an arbitrarily complex relationship depending on the specific set kernel. We show in Section 3.2 that even this weak feature space correspondence is *not* maintained by the normalized set kernel (NSK): while it is always possible to construct a complete NSK, such kernels might not be sound in the sense of Definition 6.

3 Theoretical Analysis of MI SVMs

We now analyze MI SVM approaches with respect to the three properties described above: *soundness*, *completeness*, and *convexity*. We consider three

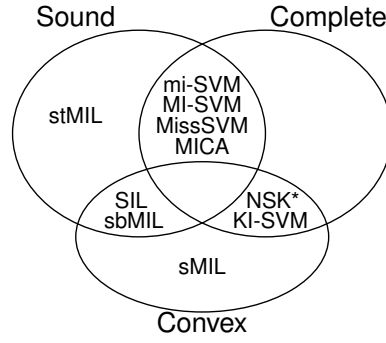


Fig. 1 Soundness, completeness and convexity of various algorithms (*under the definitions for set kernels).

types of approaches: (i) instance-based techniques that use instance kernels to learn hyperplanes that classify instances (e.g. (Andrews et al. 2003)) and then derive bag classifications from these, (ii) set-based techniques that use set kernels to learn hyperplanes that directly classify bags (e.g. (Gärtner et al. 2002)) and (iii) hybrid approaches that contain elements of both instance and set kernels (e.g. (Bunescu and Mooney 2007)). See Appendix A for the detailed formulation of the approaches discussed in this section.

Since the mapping μ between solutions and hyperplanes is trivial for most of the algorithms described below, we will often refer to “solutions” and “hyperplanes” interchangeably. For example, “a consistent solution s ” means that the corresponding hyperplane $\mu(s)$ is consistent. The distinction between solutions and hyperplanes is made clear when it is important.

Figure 1 summarizes the theoretical analysis in this section. An interesting point is that no algorithm is sound, complete, and convex. In Section 4, we show that there can be no MI SVM formulation with all three properties.

3.1 Instance-Based Kernels

SIL (Sound, Convex). A naïve approach to MI learning, single-instance learning (SIL), assigns each instance the label of its bag, creating a supervised learning problem but mislabeling negative instances in positive bags. SIL is sound, since each zero-loss solution is consistent with the MI assumption. However, there are clearly consistent MI solutions that do not require *all* instances in positive bags to be positively classified. SIL is not complete because it does not allow these solutions without loss. Since SIL uses a standard SVM formulation (Equation 1) after instance labeling, it is convex.

MI-SVM (Sound, Complete). The MI-SVM (Andrews et al. 2003) approach effectively chooses a single *witness* or *prime instance* from both positive and negative bags in the dataset. Using an optimization program with a “max”

constraint:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_i \xi_i, \\ \text{s.t.} \quad & Y_i \left[\max_{x_{ij} \in B_i} (\langle w, \phi(x_{ij}) \rangle + b) \right] \geq 1 - \xi_i, \xi_i \geq 0. \end{aligned}$$

has the effect of choosing the instance with the most positive or least negative label to represent each bag. MI-SVM is not convex because it selects instances with the maximum function in the constraints. Its formulation is equivalent to choosing the minimum of the slack variables required for each positive bag instance to put into the objective function. To see why, consider positive bags with instances x_{ij} and a constraint for each instance of the form $\langle w, x_{ij} \rangle + b \geq 1 - \xi_{ij}$, or $\xi_{ij} \geq 1 - (\langle w, x_{ij} \rangle + b)$. To minimize ξ_{ij} , for a given (w, b) , we will have $\xi_{ij} = (1 - (\langle w, x_{ij} \rangle + b))_+$, where $(x)_+ = \max\{0, x\}$, since $\xi_{ij} \geq 0$. Then, if a bag's slack ξ_i is defined as the minimum of instance slacks:

$$\begin{aligned} \xi_i = \min(\xi_{ij}) &= \min(1 - (\langle w, x_{ij} \rangle + b))_+ = (\min(1 - (\langle w, x_{ij} \rangle + b)))_+ \\ &= (1 - \max(\langle w, x_{ij} \rangle + b))_+. \end{aligned}$$

This is equivalent to minimizing ξ_i under the constraints $\xi_i \geq 0$ and $\xi_i \geq 1 - \max(\langle w, x_{ij} \rangle + b)$, so we have recovered the positive bag constraints for MI-SVM. By a similar argument, MI-SVM essentially chooses the maximum of the negative instance slacks to put into the objective function. Therefore, MI-SVM is sound and complete because it can be formulated with the following loss function:

$$\ell(\xi) = \sum_i \begin{cases} \min_j \xi_{ij} & \text{if } Y_i > 0 \\ \max_j \xi_{ij} & \text{if } Y_i < 0 \end{cases}.$$

In this form, it is easy to see that the loss is zero if and only if all negative bag instances are correctly classified, and at least one positive bag instance is classified as positive.

MICA (Sound, Complete). Instead of choosing a particular instance to act as a witnesses for each bag as is done by MI-SVM, the MI classification algorithm (MICA) finds an arbitrary convex combination of points in a positive bag to act as a witness (Mangasarian and Wild 2008). If a convex combination of points lies on one side of a hyperplane, then at least one of the points must lie on that side of the hyperplane, so MICA is sound. Conversely, for any consistent hyperplane labeling certain instances as positive, there is a convex combination of instances in each positive bag (e.g. one that just chooses some positively labeled instance), such that the solution will have zero loss. Therefore, MICA is also complete, but not convex due to the bilinear constraint introduced by the convex combination.

mi-SVM (Sound, Complete under strong consistency). The mixed integer SVM (mi-SVM) formulation (Andrews et al. 2003) uses standard SVM constraints while leaving the y_i variables unknown over $\{-1, +1\}$ for instances

in positive bags. Optimizing over binary labels makes the program nonconvex. An additional constraint $\sum_j \frac{y_{ij}+1}{2} \geq 1$ for positive bags ensures that at least one instance label in each positive bag is positive and guarantees soundness.

Some MI SVM approaches, including mi-SVM, make stronger assumptions about what it means for a hyperplane to be “consistent” with an MI dataset (see Definition 2). In particular, strong consistency also assumes that each instance has some $\{-1, +1\}$ label. Therefore, the set of strongly consistent hyperplanes \mathcal{C}' is a subset of consistent hyperplanes \mathcal{C} . Soundness and completeness can also be defined w.r.t. \mathcal{C}' rather than \mathcal{C} . This makes “strong soundness” a stronger condition than soundness, and “strong completeness” a *weaker* condition than completeness. Thus, a “complete” algorithm using the strong consistency assumption might not be complete in the sense of Definition 4. However, under the generative assumption that each instance has a label, weakening the condition for completeness in this way does not affect the behavior of the algorithm w.r.t. SRM (the target classifying hyperplane assigns a $\{-1, +1\}$ label to each instance). Accordingly, we consider such algorithms to be complete with this caveat. Because a proper choice of each y_i allows a zero-loss solution for any (strongly) consistent hyperplane, mi-SVM is complete.

MissSVM (Sound, Complete under strong consistency). Much like mi-SVM, MI learning by semi-supervised SVM (MissSVM) approaches an MI dataset as a semi-supervised learning problem in which positive bag instances have unknown labels (Zhou and Xu 2007). The MissSVM optimization program is equivalent to MI-SVM, with two additional sets of constraints that enforce both a positive and a negative label on positive bag instances (with slack). However, because only one of these constraints is expected to be satisfied for each instance, the minimum of the two slack variables corresponding to the instance constraint is included in the objective function. Because the feasible region for MissSVM is a subset of that for MI-SVM, it is sound. MissSVM requires each positive bag instance to have a label so it is complete under strong consistency. The minimum term in the objective function and the maximum terms in the constraints render MissSVM nonconvex.

I-KI-SVM (Complete, Convex). The instance-level key-instance SVM (I-KI-SVM) algorithm uses a multiple kernel learning (MKL) approach to formulate a convex program for learning *key* (or *prime*) instances from MI data (Li et al. 2009). A constraint is enforced so that each negative instance is labeled correctly. For the positive bags, a kernel is included in the MKL formulation for each possible way of choosing a positive instance out of each positive bag. If there are n positive bags each with m instances, the optimization program searches over convex combinations of m^n kernels, each representing one possible consistent instance labeling. Therefore, even though this formulation is convex, the number of variables is exponential in the problem size (in practice, the cutting plane algorithm (Kelley 1960) is used to avoid enumerating these variables). This approach is complete, since for any hyperplane corresponding to a consistent labeling, selecting (via the convex combination) the

kernel corresponding to that labeling makes the solution corresponding to that hyperplane feasible with zero loss.

On the other hand, I-KI-SVM is not sound because forming convex combinations of kernels allows hyperplanes which are not consistent to be feasible solutions. As an example, consider a one-dimensional MI dataset with negative bag $\{0\}$ and positive bags $\{0, -1\}$ and $\{0, 1\}$. Clearly, this dataset is not linearly separable, so a sound MI optimization program should have no feasible, zero-loss solution. However, learning an appropriate combination of T kernels is like learning T hyperplanes $\{w_1, \dots, w_T\}$ and summing their predictions. In the primal formulation, the constraint for a positive bag B_i with m_i instances is:

$$\sum_{t=1}^T \left(\sum_{j=1}^{m_i} d_{ij}^t \langle w, \phi(x_{ij}) \rangle \right) \geq 1,$$

where $d_{ij}^t \in \{0, 1\}$ selects the appropriate instance to include in the sum for the t^{th} combination. The constraint for each negative bag instance is:

$$\sum_{t=1}^T \langle w, \phi(x_{ij}) \rangle \leq -1.$$

So that the notation matches the original work, we exclude the constant threshold b , and instead use the equivalent feature space map $\phi(x_{ij}) = (x_{ij}, 1)$ (Li et al. 2009). For our example, there are $T = 2^2 = 4$ possible selection vectors. Concatenating the instances in each positive bag as rows in the matrix $x_{ij} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$, the corresponding selection variables are:

$$d_{ij}^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad d_{ij}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad d_{ij}^3 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad d_{ij}^4 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

Suppose we choose $w_1 = (2, -\frac{1}{2})$, $w_2 = (-2, -\frac{1}{2})$, and $w_3 = w_4 = (0, 0)$, so the last two terms are excluded from the sum over T . Then for the negative instance, 0, we have:

$$\sum_{t=1}^T \langle w, \phi(x_{ij}) \rangle = \langle (2, -\frac{1}{2}), (0, 1) \rangle + \langle (-2, -\frac{1}{2}), (0, 1) \rangle = -1 \leq -1.$$

For the first positive bag, we have:

$$\sum_{t=1}^T \left(\sum_{j=1}^{m_i} d_{ij}^t \langle w, \phi(x_{ij}) \rangle \right) = \langle (2, -\frac{1}{2}), (1, 1) \rangle + \langle (-2, -\frac{1}{2}), (0, 1) \rangle = 1 \geq 1,$$

and for the second:

$$\sum_{t=1}^T \left(\sum_{j=1}^{m_i} d_{ij}^t \langle w, \phi(x_{ij}) \rangle \right) = \langle (2, -\frac{1}{2}), (0, 1) \rangle + \langle (-2, -\frac{1}{2}), (-1, 1) \rangle = 1 \geq 1.$$

Thus, this solution satisfies constraints with zero loss, but corresponds to no consistent hyperplane, so I-KI-SVM is not sound.

3.2 Set-Based Kernels

NSK (Complete, Convex). The NSK, described in Section 2, uses an instance kernel k_I to construct a set kernel k_{NSK} that maps entire bags into a feature space (Gärtner et al. 2002). As in prior work (Gärtner et al. 2002), we say that a concept is *separable* by a kernel if there exists some consistent hyperplane classifier (with some nonzero margin) in the feature space of the kernel. For a bag classifier, “consistency” is in the supervised learning sense, where each bag is assigned its correct label, regardless of the instances. We denote the set of consistent bag hyperplanes in the feature map of the NSK by \mathcal{C}_B , and the set of consistent instance hyperplanes (in the sense of Definition 1) in the feature map of k_I by \mathcal{C}_I . As defined in Section 2, a set kernel is sound and complete when it separates bags if and only if the corresponding instance kernel separates instances, ($\mathcal{C}_I \neq \emptyset \iff \mathcal{C}_B \neq \emptyset$).

Lemma 4.2 in Gärtner et al. (2002) shows that if an underlying instance concept is separable by k_I , then there is some power $p > 0$ for which the *unnormalized* set kernel k_{MI} separates bags. This work uses a different, equivalent notion of consistency in which instead of assigning ± 1 labels to instances, a hyperplane c_ϕ in the feature space of a kernel with feature map ϕ assigns a label $1 \leq \langle \phi(x_{ij}), c_\phi \rangle$ to all positive instances, and $0 \leq \langle \phi(x_{ij}), c_\phi \rangle \leq 1 - \epsilon$ to all negative instances. Here, $\epsilon > 0$ is some arbitrary margin. To better align our results with prior work, we adopt these conventions for the remainder of this section, without loss of generality.

A proof of *both* soundness and completeness of the unnormalized NSK seems to appear as Theorem 4.4 in prior work (Gärtner et al. 2002). However, we show next that though it is always possible to construct a complete NSK, such kernels might not be sound in the sense of Definition 6.¹ We start by extending Lemma 4.2 of that work, which shows that the *unnormalized* set kernel k_{MI} is complete, to the normalized set kernel. That is, we show that k_{NSK} , constructed from the instance kernel k_I , can separate bags with margin ϵ in the RKHS of k_{NSK} with feature map Φ on bags.

Proposition 1 *An MI concept² is separable with k_{NSK} (Equation 2), using sufficiently large p , if the underlying instance concept is separable with margin ϵ by k_I , the bag size is bounded by m , and there are constants F and G such that $0 < F \leq f_{\text{norm}}(B_i) \leq G$ for all bags B_i .*

Proof. Choose an integer power $p > 0$ satisfying $p > -\frac{\log(mG/F)}{\log(1-\epsilon)}$.

¹ In a personal communication, Gärtner suggests that Theorem 4.4 (Gärtner et al. 2002) might be interpreted to mean that there is *some* k_{MI} iff there is *some* k_I .

² The set of all MI datasets derived from an instance concept is referred to as the *MI concept* in (Gärtner et al. 2002).

Let c_ϕ be the vector such that $\langle \phi(x_{ij}), c_\phi \rangle$ separates the instance concept in the instance kernel feature space. Then consider the function on bags:

$$f(B_i) = \frac{G}{f_{\text{norm}}(B_i)} \sum_{x_{ij} \in B_i} \langle \phi(x_{ij}), c_\phi \rangle^p.$$

If B_i is a positive bag, then by the MI assumption, at least one instance $x_{ij} \in B_i$ satisfies $\langle \phi(x_{ij}), c_\phi \rangle \geq 1$, so:

$$f(B_i) \geq \frac{G(1^p)}{f_{\text{norm}}(B_i)} \geq \frac{G}{G} = 1.$$

On the other hand, if B_i is a negative bag, then all instances $x_{ij} \in B_i$ satisfy $\langle \phi(x_{ij}), c_\phi \rangle \leq 1 - \epsilon$, so:

$$f(B_i) \leq \frac{Gm(1 - \epsilon)^p}{f_{\text{norm}}(B_i)} \leq \frac{Gm}{F} (1 - \epsilon)^p < \frac{Gm}{F} (1 - \epsilon)^{-\frac{\log(mG/F)}{\log(1 - \epsilon)}} = 1.$$

Therefore, this function separates bags.

To see that the function $f(B_i)$ can be written as a dot product in the RKHS corresponding to k_{NSK} , first note that if $k(x, y) = \langle \phi(x), \phi(y) \rangle$, and we raise it to power p , this is also a positive definite kernel, which is equivalent to some $\langle \psi(\phi(x)), \psi(\phi(y)) \rangle$. Therefore, the NSK feature map is given by

$$\Phi(B_i) = \frac{\sum_{x_{ij} \in B_i} \psi(\phi(x_{ij}))}{f_{\text{norm}}(B_i)}$$

. Therefore, we can rewrite f as:

$$\begin{aligned} f(B_i) &= \frac{G}{f_{\text{norm}}(B_i)} \sum_{x_{ij} \in B_i} \langle \phi(x_{ij}), c_\phi \rangle^p \\ &= \frac{G}{f_{\text{norm}}(B_i)} \sum_{x_{ij} \in B_i} \langle \psi(\phi(x_{ij})), \psi(c_\phi) \rangle \\ &= \left\langle \frac{\sum_{x_{ij} \in B_i} \psi(\phi(x_{ij}))}{f_{\text{norm}}(B_i)}, G\psi(c_\phi) \right\rangle = \langle \Phi(B_i), C_\Phi \rangle. \end{aligned}$$

So f is a hyperplane C_Φ in the normalized set kernel feature space. \square

As a corollary, given an upper bound m on bag size, the averaging normalization function $f_{\text{norm}}(B_i) = |B_i|$ is bounded by $1 \leq |B_i| \leq m$, so using $F = 1$ and $G = m$ in the proposition above, $p > -\frac{2 \log m}{\log(1 - \epsilon)}$ works to separate bags. This is just twice the p required in the unnormalized case. This result shows that even using various forms of normalization (which are useful in practice (Gärtner et al. 2002)), the NSK is complete.

On the other hand, the NSK is not sound. That is, the NSK can separate bags even when the corresponding instance kernel cannot separate the instances. Consider the instance space $\mathcal{X} = \{(1, 0), (-1, 0), (0, 1), (0, -1)\}$,

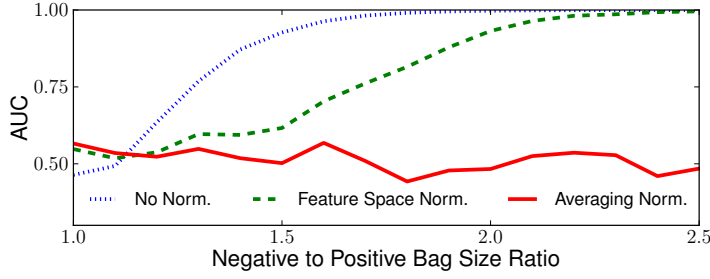


Fig. 2 Certain types of NSKs separate bags of different sizes with no underlying MI concept. The y -axis shows area under ROC curve.

with respective labels $\{+1, +1, -1, -1\}$ (corresponding to XOR). With a linear instance kernel $k_I(x, x') = \langle x, x' \rangle$, the instance concept is clearly not separable. However, with $p = 2$, the set kernel $k_{MI}(X, X') = \sum_{x, x'} \langle x, x' \rangle^2$ can separate any MI dataset derived from these instances. To see why, consider the explicit feature map ϕ of the quadratic kernel $(u, v) \mapsto (u^2, \sqrt{2}uv, v^2)$. Then, the set kernel feature map is the sum of instance kernel feature maps: $\Phi(X) = \sum_{x \in X} \phi(x)$. The linear function $f(X) = \langle \Phi(X), (1, 0, 0) \rangle$ in the feature space of $k_{MI}(\cdot, \cdot)$ then separates any MI dataset, since the first component of the map $\Phi(X)$ is nonzero if and only if $X \in 2^{\mathcal{X}}$ contains either $(1, 0)$ or $(-1, 0)$. Gärtner (2002) states that if f_{MI} separates an MI dataset, then $f_I(x) = f_{MI}(\{x\})$ can separate instances. However, this is not sound under our definition because $f_{MI} \notin \mathcal{H}_I$; i.e., there is no instance hyperplane in the original instance hyperplane space corresponding to the bag hyperplane applied to singleton sets.

Finally, another form of unsoundness arises for set kernels due to the effects of bag size. For example, consider an MI problem in which all bag instances are identical (say $x_{ij} = 1$), but positive bags have size 10 while negative bags have size 5. Then for an unnormalized linear kernel, the feature mapping of a positive bag will be 10, while the negative bag feature space value will be 5. Clearly, there is no underlying MI concept; yet, the set kernel is able to separate positive and negative bags in the feature space via the effects of bag size. We illustrate this further in Figure 2 using synthetic datasets. In these datasets, each instance has 25 features, which are drawn independently from the standard normal distribution $\mathcal{N}(0, 1)$. There are 50 positive bags, each with 10 instances, and 50 negative bags of sizes that vary across datasets. Even though there is no underlying instance concept to learn, the set kernels with either no normalization or feature space normalization can learn to distinguish between positive and negative bags as the discrepancy in sizes grows.

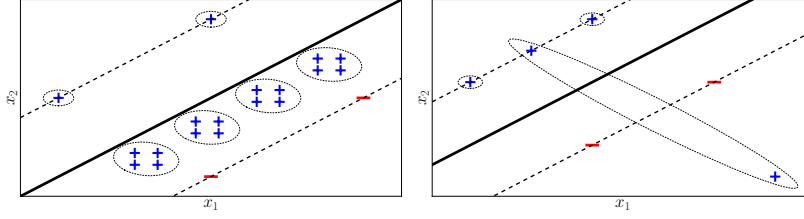


Fig. 3 Synthetic datasets illustrating when soundness and/or completeness fail for sMIL and stMIL. **Left** shows when a sMIL solution without loss allows a misclassification of an arbitrary number of bags whose averages lie close to the wrong side of the (inconsistent) classifier. **Right** shows a consistent MI separator with nonzero loss for sMIL and stMIL.

3.3 Hybrid Approaches

sMIL (Convex). The sparse MI learning (sMIL) algorithm uses both set and instance kernels (Bunescu and Mooney 2007). Assuming that in the worst case all but one instance in each positive bag is negative, the average instance label within positive bags is controlled by a “balancing constraint” $\langle w, \Phi(B_i) \rangle + b \geq \frac{2-|B_i|}{|B_i|} - \xi_i$. Here, $\Phi(B_i) = \frac{1}{|B_i|} \sum_j \phi(x_{ij})$ denotes the feature map induced by the averaging normalized set kernel for bags. The standard supervised constraint is used for negative instances, with an instance kernel feature map ϕ . This is the same as putting each negative instance into its own bag. The sMIL approach is convex, but is neither sound nor complete. A counterexample to soundness is shown in Figure 3 (left). In the figure, all instances in positive bags are marked with plus signs, and the negative instances are marked with minus signs. Because the misclassified bags contain four instances, they are allowed to be within $\frac{2-4}{4} = -\frac{1}{2}$ of the margin without any loss. Therefore, this solution is feasible and optimal without loss, but not consistent. In fact, an arbitrary number of positive bags can be placed within the margin as shown, leading to an arbitrarily poor classification of bags. A counterexample to the completeness of sMIL is shown in Figure 3 (right). While the solution is consistent, it is not feasible without loss because the average of the instances in the large positive bag lies below the separating line and therefore does not satisfy the balancing constraint.

stMIL (Sound). The sparse transductive MI learning (stMIL) formulation includes the sMIL constraints, as well as $|\langle w, \phi(x_{ij}) \rangle + b| \geq 1 - \xi_{ij}$ for every instance x_{ij} in a positive bag, which force instances within bags to be outside the margin (Bunescu and Mooney 2007). The addition of these constraints makes the problem nonconvex. But like mi-SVM, these constraints impose a label on every instance, so stMIL is sound by avoiding cases such as Figure 3 (left). The scenario in Figure 3 (right) is also a counterexample to the completeness of stMIL because the instances in the large bag satisfy the transductive constraint but violate the balancing constraint.

sbMIL (Sound, Convex). A third approach from Bunescu and Mooney (2007), sparse balanced MI learning (sbMIL), searches for a *balancing* param-

eter η representing the fraction of positive examples in positive bags. First, an initial solution is found via sMIL. Then, the top η instances with the highest classifications from each positive bag are assigned a positive label, and the remaining are assigned a negative label. Finally, a standard supervised SVM is used to produce a final classifier from the instances. The sbMIL formulation is sound because it imposes a labeling on instances that is consistent with the MI assumption. However, it is not complete (by design) because it fixes a particular set of preassigned labels from the sMIL classifier. Since sMIL is convex, the selected set of instances for each η is unique and so sbMIL does not permit all possible consistent solutions. Because it successively uses two convex approaches, we consider it to be convex.

B-KI-SVM (Complete, Convex). A variant of I-KI-SVM, described above, is bag-level key-instance SVM (B-KI-SVM) (Li et al. 2009). B-KI-SVM is a hybrid because it uses the average of instances in each negative bag to represent the bag, rather than including a constraint for every negative instance. The example for unsoundness of I-KI-SVM above also applies to B-KI-SVM, since the negative bag contains a single instance. However, B-KI-SVM is still complete, since the solution corresponding to a consistent hyperplane labels all negative instances negative, and so the average of these instances is also labeled negative and has zero loss under this solution. Furthermore, the B-KI-SVM is convex, but still uses an optimization program with an exponential number of variables.

4 Can All Three Properties Be Satisfied?

None of the algorithms discussed above are sound, complete, *and* convex. In this section we prove that this observation is no coincidence:

Theorem 1 *No MI optimization program is sound, complete, and convex.*

Proof Suppose some MI optimization program is sound, complete, and convex. Then for any dataset, ℓ is a convex function, so $\{s \in \mathcal{S} : \ell(s) = 0\}$ is a convex set. Since the feasible region \mathcal{F} is also convex, the set $\mathcal{Z} = \mathcal{F} \cap \{s \in \mathcal{S} : \ell(s) = 0\}$ is convex as well. Since \mathcal{Z} is convex, it is a *path-connected* set. A set $V \in \mathcal{S}$ is path-connected when for any two points $v_1, v_2 \in V$, there exists a continuous parametric function $p : [0, 1] \rightarrow \mathcal{S}$ such that $p(0) = v_1$, $p(1) = v_2$, and $p([0, 1]) \subseteq V$. For a convex set, the lines connecting any two points in the set are such paths.

Furthermore, since the MI optimization program is sound and complete, the mapping between the solution and hyperplane spaces is such that $\mu(\mathcal{Z}) = \mathcal{C}$; that is, the set of consistent hyperplanes is the image of the \mathcal{Z} under μ . Since μ is continuous and \mathcal{Z} path-connected, this implies that \mathcal{C} is path-connected. Intuitively, the image of a path-connect set under a continuous function is also path-connected because the composition of continuous functions is also continuous. Thus, any path in \mathcal{Z} composed with μ produces a continuous path in \mathcal{C} .

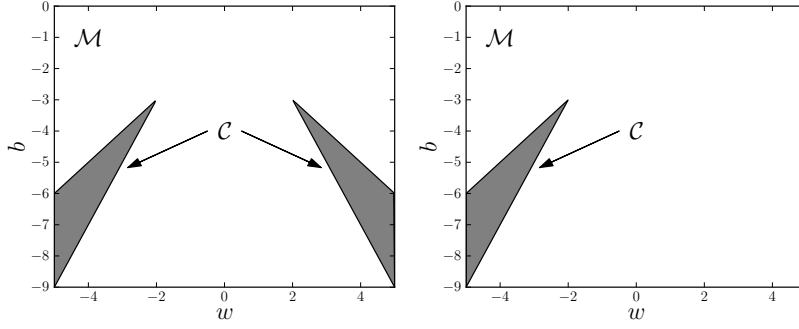


Fig. 4 (Left) The shaded region shows the set \mathcal{C} of consistent hyperplanes in the space of hyperplanes (w, b) for the example in the proof of Theorem 1. **(Right)** When a supervised labeling is applied to each instance, the set of consistent hyperplanes becomes a convex set.

However, consider the one-dimensional dataset with a positive bag $\{-2, 2\}$ and a negative bag $\{-1, 1\}$. A consistent linear “hyperplane” $((w, b) \in \mathbb{R}^2)$ must label either 2 or -2 “positive” and the other instances negative. The “support vectors” for these two scenarios are either -2 and -1 , or 1 and 2 . Therefore, the set of consistent hyperplanes is the union of the regions where (1) $w + b \leq -1$ and (2) $w + b \geq 1$, or where $(-1)w + b \leq -1$ and $(-2)w + b \geq 1$. This set is shown in Figure 4 (left), and is clearly not path-connected (no path connects the two disjoint regions). Thus, we have a contradiction with the implication that \mathcal{C} is path-connected for every MI dataset, so there cannot be a sound, complete, and convex MI optimization program. \square

Intuitively, the inability to satisfy all three properties is related to the disjoint nature of the set of consistent hyperplanes, which is in turn related to the combinatorial nature of the set of consistent instance labelings. In the standard supervised setting, this difficulty does not arise, since the set of consistent hyperplanes forms a convex set. For example, if we fix a labeling in the example of Theorem 1 so that -2 is positive and the other instances are negative, the set of consistent hyperplanes collapses to the convex set shown in Figure 4 (right).

Theorem 1 is in line with previous complexity results for MI classification via hyperplanes (Kundakcioglu et al. 2010; Diochnos et al. 2012). For clarity, we include the theorem below, expressed in terms of our formalism:

Theorem 2 *Given an MI problem (B, Y) , a set of bags with $|B_i| \leq k$, $k \geq 3$, the decision problem MI-CONSIS of determining whether there exists a hyperplane consistent with (B, Y) (i.e. is $\mathcal{C} = \emptyset$?) is NP-complete. It is also NP-complete to determine if $\mathcal{C}' = \emptyset$, where \mathcal{C}' is the set of strongly consistent hyperplanes.*

The proof of Theorem 2 (Diochnos et al. 2012) reduces a 3-SAT instance to an instance of MI-CONSIS such that there is a strongly consistent hyperplane if the 3-SAT formula is satisfiable and no consistent (in the usual sense)

hyperplane if the formula is not satisfiable. Thus, the proof works for either notion of consistency, though the distinction is not made in the original work.

If there were a sound, complete, and convex MI optimization program, the question $\mathcal{C} = \emptyset$ is equivalent to asking whether $\mathcal{Z} = \emptyset$, or whether there is a feasible, zero-loss solution to the MI optimization program. Similarly, if a set kernel approach is sound and complete, then $\mathcal{C}_B = \emptyset \iff \mathcal{C}_I = \emptyset$. Thus, if we could construct a sound and complete set kernel in polynomial time, we could use it in conjunction with a standard convex SVM formulation to search for a consistent bag classifier to decide whether the instances were separable. In either case, we could solve an NP-complete problem via convex programming, which is generally regarded to be efficiently solvable, albeit in a non-Turing model of computation (Ben-Tal and Nemirovskiĭ 2001).

Finally, the complexity results above allow us to show that MI hypotheses over arbitrary distributions are not efficiently probably approximately correct (PAC) learnable with classifying hyperplanes. Previous work (Auer et al. 1997) reduced PAC learning axis parallel rectangles (APRs) for MI concepts over arbitrary distributions to PAC learning disjunctive normal form (DNF) formulae. Other work has shown that concepts PAC learnable from one-sided noise are also PAC learnable from MI examples, assuming that all bag instances are drawn independently from some instance distribution (Blum and Kalai 1998). Some recent results give a bound on the sample complexity of bag classifiers in terms of bag sizes, and show that it is possible to learn a *bag* classifier from labeled bags (Sabato et al. 2010; Sabato and Tishby 2012). Because we can reduce MI-CONSIS to an algorithm that PAC learns MI concepts with hyperplanes, we could produce an RP algorithm to solve MI-CONSIS. An RP algorithm for a decision problem runs in polynomial time in the input instance size, always returns **false** when the correct answer is **false**, and returns **true** at least half of the time across randomized runs when the correct answer is **true**.

Proposition 2 *If $\text{RP} \neq \text{NP}$, then there is no algorithm \mathcal{A} that (for arbitrary bag distributions) efficiently PAC learns MI hyperplanes.*

Proof. We can reduce MI-CONSIS to an algorithm \mathcal{A} that efficiently PAC learns MI concepts with hyperplanes. For an instance of MI-CONSIS, (B, Y) , we can construct $(B, Y, D, \epsilon, \delta)$, an instance of the PAC learning problem to provide to \mathcal{A} , where labeled examples are drawn from (B, Y) via the uniform distribution D . By setting $\epsilon = \frac{1}{|B|+1}$ and $\delta = \frac{1}{4}$ (any arbitrary $\delta < \frac{1}{2}$ works), we ensure that \mathcal{A} will produce a classifier consistent with every bag, with probability $1 - \delta > \frac{1}{2}$. If \mathcal{A} fails to produce a classifier, we return **false**. If \mathcal{A} produces a classifier, we check it for consistency with each bag in B , and return **true** if it is, or **false** otherwise. Note that $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$ are polynomial in the size of the input, and the reduction takes polynomial time to check each bag for consistency.

Now, suppose that MI-CONSIS returns **false**, then \mathcal{A} will either fail to learn a classifier, or produce an inconsistent classifier and return **false** during the consistency check. On the other hand, if MI-CONSIS returns **true**, then

with probability $1 - \delta > \frac{1}{2}$, \mathcal{A} will learn a classifier with error less than ϵ (i.e. one consistent with all bags), and return **true**. Because \mathcal{A} only requires a polynomial number of examples, the reduction produces an algorithm in RP to solve MI-CONSIS. Since MI-CONSIS is NP-complete, it is impossible for \mathcal{A} to exist unless $\text{RP} = \text{NP}$. \square

In light of these complexity results, practical algorithms must sacrifice either soundness, completeness, or convexity. Therefore, we empirically evaluate how tradeoffs between these three properties affect classification performance.

5 Empirical Evaluation

Given the properties possessed by (or lacking in) the various classification algorithms analyzed above, it is natural to wonder whether one property is more important than another for classification accuracy in practice. In this section we perform a large-scale, detailed empirical comparison of several algorithms on a variety of real-world problems to provide insight into this question.

Datasets. We use twenty-two MI benchmark datasets for evaluation. The two **musk** datasets come from the drug activity prediction domain, which originally motivated the creation of the MIC framework (Dietterich et al. 1997; Frank and Asuncion 2010), and the two **text** datasets are from the text categorization domain (Andrews et al. 2003). There are three animal image datasets (**elephant**, **tiger**, and **fox**) and three scene image datasets (**mountain**, **field**, **flower**) from the content-based image retrieval (CBIR) domain (Andrews et al. 2003; Zhang et al. 2002). Also from the CBIR domain is a twenty-five class spatially independent, variable area, and lighting (SIVAL) dataset, which has been annotated with labels for both bags and instances (Settles et al. 2008). From these, we construct twelve one-vs.-one datasets by randomly pairing images classes. We use all twenty-two datasets (ignoring instance labels) when evaluating algorithms on the bag-labeling task and the twelve SIVAL datasets when evaluating them on the instance-labeling task.

Methodology. We implement most techniques in Python using NumPy (Ascher et al. 2001) for general matrix computations, and the CVXOPT library (Dahl and Vandenberghe 2009) for solving quadratic programs (QPs).³ We use the authors’ MATLAB code for the key-instance SVM (KI-SVM) approaches.⁴ For each dataset, we use ten-fold cross-validation with the same folds across all techniques and accuracy as the performance metric. We use the radial basis function (RBF) kernel for all approaches, and it serves as the instance kernel in set kernel approaches. We implement the normalized set kernel with both averaging and feature space normalization, described in Section 2. We use random parameter search (Bergstra and Bengio 2012) with five-fold cross-validation to select the RBF kernel parameter γ from $[10^{-6}, 10^1]$ and the regularization parameter C from $[10^{-2}, 10^5]$. For the set kernel, we fix $p = 1$,

³ The code is available online at http://engr.case.edu/doran_gary/code.html

⁴ http://lamda.nju.edu.cn/code_KISVM.ashx

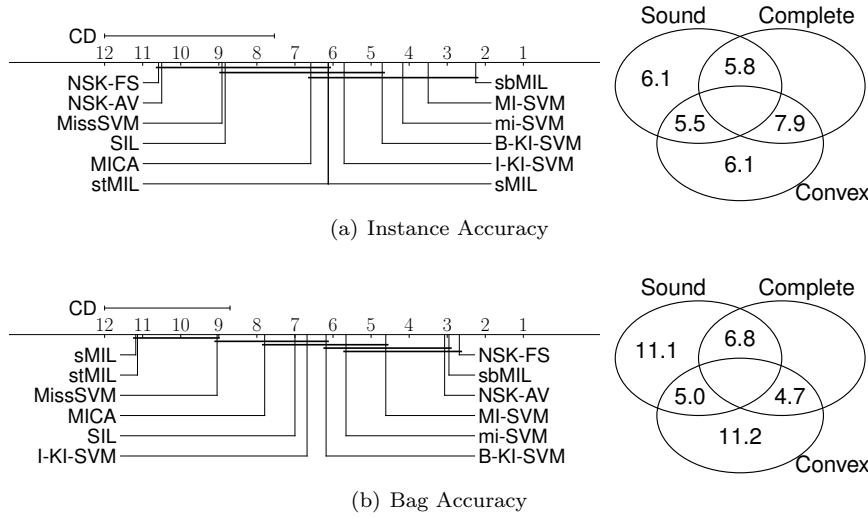


Fig. 5 Ranks of the various MI SVM approaches on the instance and bag labeling tasks using accuracy for evaluation. The critical difference diagrams (**Left**) show the average rank of each technique across datasets, with techniques being statistically different at an $\alpha = 0.1$ significance level if the ranks differ by more than the critical difference (CD) indicated above the axis. Thick horizontal lines indicate statistically indistinguishable groups (i.e. a technique is statistically different from any technique to which it is *not* connected with a horizontal line). The Venn diagrams (**Right**) show the average ranks of techniques within each Sound/Complete/Convex categorization.

but with an RBF kernel, p can be absorbed into the constant γ . We search for the η parameter of sbMIL within the range $[0, 1]$. For an algorithm requiring $m \in \{1, 2, 3\}$ parameters, we evaluate 5^m random parameter combinations for the search. For techniques that rely on iteratively solving QPs, iteration continues at most 50 times or until the change in objective function value falls below 10^{-6} . MICA was originally formulated using L_1 regularization, but in our experiments we use the L_2 norm to provide a more direct comparison to other approaches. We only use bag labels when performing parameter cross-validation, even for the instance-labeling task (we only use instance labels to perform the final evaluation of instance predictions pooled across the ten outer folds).

Hypothesis Tests. To statistically compare the classifiers, we use the approach described in Demšar (2006). First, we rank (with respect to accuracy) the k algorithms for each dataset, with 1 being the best and k the worst, and then we average the ranks across datasets. Next, we use the Friedman test to reject the null hypothesis that the algorithms perform equally at an $\alpha = 0.001$ significance level. Finally, we plot the average ranks using a *critical difference* diagram, which uses the Nemenyi test to identify statistically equivalent groups of classifiers at an $\alpha = 0.1$ significance level. Figure 5 shows the resulting ranks for the instance and bag labeling tasks using the accuracy evaluation metric for ranking.

Effect of Soundness and Completeness. Conceptually, soundness is more important than completeness for SRM approaches. Soundness ensures that, on the set of consistent hyperplanes, the loss on the corresponding solutions is an upper bound on the true risk. On the other hand, techniques lacking soundness might return solutions that appear to perform well (with respect to empirical risk), but do not generalize to new data. This hypothesis about the relative importance of soundness and completeness is generally consistent with the results in Figure 5, where we see that techniques that are either sound and complete or sound and convex are generally more accurate than other approaches. The few exceptions to this trend are explained below.

Effect of the Labeling Task. The results for bag accuracy in Figure 5(b) appear to contradict the observation that unsound approaches will produce solutions that do not generalize well to new data. In particular, the unsound NSK approaches are among the top performers on the bag-labeling task. However, the success of the NSK makes sense because it uses a sound and complete supervised SVM to perform SRM using hyperplanes defined over *bags*. The unsoundness of the NSK is caused by the tenuous relationship between the bag and instance hypothesis spaces. This can be also seen from the fact that the NSK performs poorly on the instance labeling task, as shown in Figure 5(a). The case of the NSK shows a previously observed empirical phenomenon, that there might be little correlation between the performance of algorithms on the two labeling tasks (Tragante do O et al. 2011). Unfortunately, few MI datasets come with instance labels, making comparisons between the two tasks difficult in practice. We feel that it is important to construct more such datasets to better understand the relationship between algorithms’ behavior on these two tasks.

Effect of Instance Distributions within Bags. Although we might expect the NSK to perform well on the bag labeling task, we still need to explain why it often performs *better* than other sound and complete MI approaches. We hypothesize that the set kernel is capable of using information about bags to which instance-based approaches do not have access. The NSK with averaging normalization can be thought of as mapping (empirical) distributions of instances within bags into an RKHS for classification via the *kernel mean map* (Smola et al. 2007). When a kernel such as the RBF kernel is used, the mean map is an injective mapping of distributions into a feature space, which allows learning linear concepts directly from instance distributions within bags (Muandet et al. 2012). In many domains, it is natural to think that the distribution of even negative instances within positive bags might provide additional useful contextual information to a bag classifier. For example, when trying to learn a classifier for images containing spoons, it is more likely that other objects in such images will be forks, plates, or other tableware rather than grass, trees, or water. This context may be beneficial, for example, if only part of a spoon were visible in an image. Instance classifiers that select only single instances from bags to learn a concept are not able to take advantage of such contextual information. Although the MI *formulation* says nothing about contextual relationships between instances, many MI

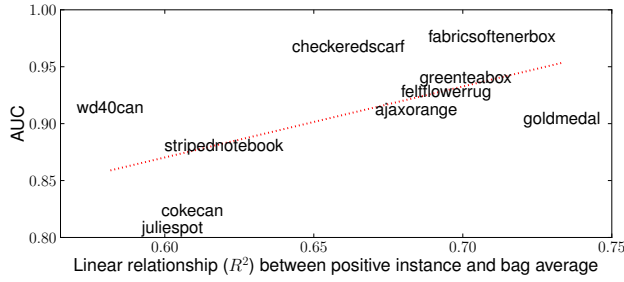


Fig. 6 The degree of linear relationship between prime instances and bag averages and the classification performance of a linear set kernel with averaging normalization are correlated.

Table 1 Average ranks (lower ranks are better) of techniques across 7 datasets with and without 15 random restarts. Boldface indicates that the ranks are significantly different. Using random restarts only occasionally affords a significant advantage to nonconvex approaches.

| Technique | mi-SVM | MI-SVM | MICA | MissSVM |
|--------------|--------|-------------|------|---------|
| without R.R. | 1.43 | 1.57 | 1.43 | 1.71 |
| with R.R. | 1.57 | 1.43 | 1.57 | 1.29 |

domains may possess them in practice, leading to improved performance for techniques that can take advantage of such relationships.

To demonstrate this effect, we design an experiment to test the hypothesis that when there is a relationship between bag distributions and positive instances within positive bags, the NSK will be more accurate. To test this hypothesis, we use the one-vs.-all instance-labeled SIVAL datasets (Settles et al. 2008) and the linear instance kernel. In this case, the set kernel maps each bag to the average of the bag’s instances. We then determine if there is a (linear) relationship between the distribution representer (bag average) and positive instances as follows. We pick the most positively labeled positive instance after a classifier is found, and compute the R^2 coefficient of determination from a least squares multiple linear regression between bag averages and these instances. If this R^2 is high, a (linear) relationship exists. Because there is a large class imbalance in these one-vs.-all datasets, we use area under ROC curve (AUC) rather than accuracy as a performance metric. In Figure 6, we plot the AUC against the R^2 for several datasets in which the linear set kernel found a good classifier (high AUC). We observe that indeed, for these datasets, there is a general association ($r = 0.59$) between a strong linear relationship between bag averages and positive instances, and classifier accuracy, indicating how the linear NSK can take advantage of bag distributions in practice.

Effect of Convexity. In the Venn diagram of Figure 5(b), the approaches sacrificing completeness for convexity appear to outperform nonconvex, sound, complete approaches. A possible explanation for this behavior is that nonconvex approaches rely on heuristic optimization techniques that are only likely to converge to local optima. For example, some optimization programs require an initial labeling of instances. A widely-used heuristic is to assign each instance

the label of its bag. However, use of this heuristic is justified intuitively, not theoretically. Therefore, the disparity between convex and nonconvex approaches might be caused by a deficiency in optimization heuristics rather than in the formulation itself. To rule out this possibility, we also run the nonconvex approaches using 15 random restarts (using the instance-labeling heuristic as one restart). Due to practical time constraints in running many random restarts on large datasets, we use the `musk1`, `elephant`, `tiger`, `fox`, `mountain`, `field`, and `flower` datasets to explore the effects of random restarts. We tabulate performance in terms of a 1 or 2 rank (comparing random restarts to no random restarts) of each technique averaged across datasets and tested for significance at an $\alpha = 0.1$ level with the Wilcoxon signed-rank test.

From the results (Table 1), we observe that random restarts only occasionally offer a significant advantage to nonconvex, sound, complete approaches. These results align with previous work, showing that the initial heuristic assignment of labels can produce relatively good classifiers in terms of classification accuracy, especially when there are multiple positive instances per positive bag (Gehler and Chapelle 2007). We conclude that the choice of optimization heuristic for nonconvex approaches does not present a serious disadvantage relative to sound, convex approaches.

Time and Space Requirements. Instance kernel approaches are much more computationally expensive than set kernel approaches, since kernel sizes are $O(n^2)$ in terms of the number of instances rather than bags. For the datasets used, instance kernels range from having 1.8×10^5 to 3.5×10^7 entries, whereas bag kernels contain from 6.9×10^3 to 1.3×10^5 entries. Runtime also increases significantly for instance methods due to the increased number of variables in the optimization program. For set-based methods, median training and testing time for any particular set of parameters takes between 0.4 and 50 seconds across datasets. For hybrid methods, this increases to between 1 and 635 seconds, and for instance-based methods it is between 1 and 1440 seconds. These figures increase by a large factor when parameter search and cross validation are used. In particular, the actual overall training time of the hybrid sbMIL algorithm is longer due to the extra search for the balancing parameter η required. Therefore, training instance-based classifiers on very large datasets becomes impractical.

Finally, we note that our analysis of soundness and completeness is motivated by the need for loss functions to accurately assess the empirical risk of MI hyperplane concepts. Empirical risk is expressed in terms of classification accuracy. On the other hand, AUC is another popular evaluation metric, which can be viewed as an estimate of the probability that a classifier will correctly “rank” examples (assign higher real-valued labels to positive examples than to negative examples). A key direction for future work is to extend our approach to the AUC metric, for which corresponding notions of soundness and completeness might be defined.

Another direction for future work is to generalize the definitions of soundness and completeness to evaluate the behavior of approaches on nonseparable datasets. However, as generalizations, these properties would need to coincide

with ours on the separable datasets. Therefore, if an algorithm lacks soundness or completeness under our definitions, it would also lack these properties under the alternate definitions. On the other hand, an algorithm might be sound or complete under our definition but not under the alternate definitions. Thus, each algorithm can only lose these properties (not gain them) under generalized definitions.

6 Conclusion

In this work, we formally specify soundness and completeness properties desired in algorithms for MI classification via hyperplanes. We use these properties to analyze a variety of existing techniques, and show that no convex approach can have both properties. We evaluate the performance of these approaches empirically to characterize the practical tradeoffs between properties. Though the experimental results generally align with our theoretical analysis, we find that the effects of soundness and completeness depend on the labeling task. We hypothesize that set kernels can use additional information about instance distributions within bags not available to instance-based approaches. Sound and complete approaches lack convexity, but we show using random restarts that more rigorous optimization of the objective function does not significantly improve the performance of these techniques. In future work, we plan to explore the relationship between the instance- and bag-labeling tasks to see if techniques such as the set kernel, which are good at bag classification but not at instance classification, can be modified to accurately label instances. We also plan to define corresponding notions of soundness and completeness for other metrics such as AUC, and derive similar tradeoffs for that case.

Acknowledgements

We thank the anonymous reviewers for their comments. G. Doran was supported by GAANN grant P200A090265 from the US Department of Education. S. Ray was partially supported by CWRU award OSA110264.

References

- Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568.
- Ascher, D., Dubois, P., Hinsén, K., Hugunin, J., and Oliphant, T. (2001). *Numerical Python*. Lawrence Livermore National Laboratory, Livermore, CA.
- Auer, P., Long, P., and Srinivasan, A. (1997). Approximating hyper-rectangles: Learning and pseudo-random sets. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computation*, pages 314–323. ACM.
- Ben-Tal, A. and Nemirovskii, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. MPS-SIAM Series on Optimization. SIAM.

- Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305.
- Blockeel, H., Page, D., and Srinivasan, A. (2005). Multi-instance tree learning. In *Proceedings of the International Conference on Machine Learning*, pages 57–64.
- Blum, A. and Kalai, A. (1998). A note on learning from multiple-instance examples. *Machine Learning Journal*, 30:23–29.
- Bunescu, R. and Mooney, R. (2007). Multiple instance learning from sparse positive bags. In *Proceedings of the International Conference on Machine Learning*, pages 105–112.
- Chen, Y., Bi, J., and Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947.
- Dahl, J. and Vandenberghe, L. (2009). CVXOPT: A python package for convex optimization.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.
- Dietterich, T. G., Lathrop, R. H., and Lozano-Perez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71.
- Diochnos, D., Sloan, R., and Turán, G. (2012). On multiple-instance learning of halfspaces. *Information Processing Letters*, 112(23):933–936.
- Frank, A. and Asuncion, A. (2010). UCI machine learning repository.
- Gärtner, T., Flach, P., Kowalczyk, A., and Smola, A. (2002). Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning*, pages 179–186.
- Gehler, P. and Chapelle, O. (2007). Deterministic annealing for multiple-instance learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 123–130.
- Kelley, Jr., J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial & Applied Mathematics*, 8(4):703–712.
- Kundakcioglu, O., Seref, O., and Pardalos, P. (2010). Multiple instance learning via margin maximization. *Applied Numerical Mathematics*, 60(4):358–369.
- Li, Y.-F., Kwok, J. T., Tsang, I. W., and Zhou, Z.-H. (2009). A convex method for locating regions of interest with multi-instance learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 15–30. Springer.
- Mangasarian, O. and Wild, E. (2008). Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications*, 137:555–568.
- Maron, O. (1998). *Learning from Ambiguity*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.
- Muandet, K., Fukumizu, K., Dinuzzo, F., and Schölkopf, B. (2012). Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems*, pages 10–18.
- Ramon, J. and De Raedt, L. (2000). Multi instance neural networks. In *Proceedings of the ICML-2000 workshop on Attribute-Value and Relational Learning*.
- Sabato, S., Srebro, N., and Tishby, N. (2010). Reducing label complexity by learning from bags. In *International Conference on Artificial Intelligence and Statistics*, pages 685–692.
- Sabato, S. and Tishby, N. (2012). Multi-instance learning with any hypothesis class. *Journal of Machine Learning Research*, 13:2999–3039.
- Settles, B., Craven, M., and Ray, S. (2008). Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, pages 1289–1296.
- Smola, A., Gretton, A., Song, L., and Schölkopf, B. (2007). A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31.
- Tao, Q., Scott, S. D., and Vinodchandran, N. V. (2004). SVM-based generalized multiple-instance learning via approximate box counting. In *Proceedings of the International Conference on Machine Learning*, pages 779–806.
- Tragante do O, V., Fierens, D., and Blockeel, H. (2011). Instance-level accuracy versus bag-level accuracy in multi-instance learning. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence*.
- Xu, X. and Frank, E. (2004). Logistic regression and boosting for labeled bags of instances. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 272–281.

- Zhang, Q. and Goldman, S. (2001). EM-DD: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems*, pages 1073–1080.
- Zhang, Q., Yu, W., Goldman, S., and Fritts, J. (2002). Content-based image retrieval using multiple-instance learning. In *Proceedings of the International Conference on Machine Learning*, pages 682–689. Morgan Kaufmann.
- Zhou, Z. and Xu, J. (2007). On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the International Conference on Machine Learning*, pages 1167–1174.
- Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009). Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the International Conference on Machine Learning*, pages 1249–1256.
- Zhou, Z.-H. and Zhang, M.-L. (2002). Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology*.

A MI SVM Formulations

The formulations described in Section 3 are listed in detail below. In many cases, the notation and precise formulation of each approach has been modified slightly from the original publication for consistency. In the constraints, there is implicit universal quantification over the indices corresponding to each bag in the dataset and the instances within each bag. For formulations that mix instance- and bag-level slack variables ξ_{ij} and ξ_i , respectively, sums of the form $\sum_{i,j} \xi_{ij}$ are intended to be taken over both sets of slack variables. The notation $|B^+|$, $|X^+|$, and $|X^-|$ is used to denote the number of positive bags, instances in positive bags, and instances in negative bags, respectively.

SIL (Sound, Convex).

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i,j} \xi_{ij}, \quad \text{s.t.} \quad Y_i (\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0$$

MI-SVM (Sound, Complete). (Andrews et al. 2003)

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i, \quad \text{s.t.} \quad Y_i \left[\max_{x_{ij} \in B_i} (\langle w, \phi(x_{ij}) \rangle + b) \right] \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

MICA (Sound, Complete). (Mangasarian and Wild 2008)

$$\begin{aligned} \min_{w,b,\xi,d} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i,j} \xi_{ij}, \\ \text{s.t.} \quad & \begin{cases} \langle w, \sum_j d_{ij} \phi(x_{ij}) \rangle + b \geq 1 - \xi_i & \text{if } Y_i = 1 \\ -(\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_{ij} & \text{if } Y_i = -1 \end{cases}, \quad \sum_j d_{ij} = 1, \quad d_{ij} \geq 0, \quad \xi_{ij} \geq 0 \end{aligned}$$

mi-SVM (Sound, Complete under strong consistency). (Andrews et al. 2003)

$$\begin{aligned} \min_{w,b,\xi,y} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i,j} \xi_{ij}, \\ \text{s.t.} \quad & y_{ij} (\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0, \quad y_{ij} \in \{-1, 1\}, \quad \begin{cases} \sum_j \frac{y_{ij}+1}{2} \geq 1 & \text{if } Y_i = 1 \\ y_{ij} = -1 & \text{if } Y_i = -1 \end{cases} \end{aligned}$$

MissSVM (Sound, Complete under strong consistency). (Zhou and Xu 2007)

$$\begin{aligned} \min_{w,b,\xi,\epsilon} \quad & \frac{1}{2} \|w\|^2 + \frac{C}{|B^+|} \sum_i \xi_i^+ + \frac{C}{|X^-|} \sum_{i,j} \xi_{ij}^- + \frac{C}{|X^+|} \sum_{i,j} \min(\epsilon_{ij}^+, \epsilon_{ij}^-), \\ \text{s.t.} \quad & \begin{cases} \max_j (\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_i^+ \\ \langle w, \phi(x_{ij}) \rangle + b \geq 1 - \epsilon_{ij}^+ \\ -(\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \epsilon_{ij}^- \end{cases} \quad \text{if } Y_i = 1, \quad \xi_{ij} \geq 0, \quad \epsilon_{ij} \geq 0 \\ & \begin{cases} -(\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_{ij}^- \end{cases} \quad \text{if } Y_i = -1 \end{aligned}$$

I-KI-SVM (Complete, Convex). (Li et al. 2009)

$$\begin{aligned} \min_{w,\rho,\xi,d} \quad & \frac{1}{2} \|w\|^2 - \rho + C_1 \sum_i (\xi_i^+)^2 + C_2 \sum_{i,j} (\xi_{ij}^-)^2, \\ \text{s.t.} \quad & \begin{cases} \langle w, \sum_j d_{ij} \phi(x_{ij}) \rangle \geq \rho - \xi_i^+ & \text{if } Y_i = 1 \\ -\langle w, \phi(x_{ij}) \rangle \geq \rho - \xi_{ij}^- & \text{if } Y_i = -1 \end{cases}, \quad \sum_j d_{ij} = 1, \quad d_{ij} \in \{0, 1\}, \quad \xi_{ij} \geq 0 \end{aligned}$$

NSK (Complete, Convex). (Gärtner et al. 2002)

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i, \quad \text{s.t.} \quad Y_i \left(\left\langle w, \frac{1}{f_{\text{norm}}(B_i)} \sum_j \phi(x_{ij}) \right\rangle + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

sMIL (Convex). (Bunescu and Mooney 2007)

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + \frac{C}{|B^+|} \sum_i \xi_i^+ + \frac{C}{|X^-|} \sum_{i,j} \xi_{ij}^-,$$

$$\text{s.t.} \quad \begin{cases} \left\langle w, \frac{1}{|B_i|} \sum_j \phi(x_{ij}) \right\rangle + b \geq \frac{2-|B_i|}{|B_i|} - \xi_i^+ & \text{if } Y_i = 1 \\ -(\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_{ij}^- & \text{if } Y_i = -1 \end{cases}, \quad \xi_{ij} \geq 0$$

stMIL (Sound). (Bunescu and Mooney 2007)

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + \frac{C}{|B^+|} \sum_i \xi_i^+ + \frac{C}{|X^-|} \sum_{i,j} \xi_{ij}^- + \frac{C}{|X^+|} \sum_{i,j} \xi_{ij}^*,$$

$$\text{s.t.} \quad \begin{cases} \left\langle w, \frac{1}{|B_i|} \sum_j \phi(x_{ij}) \right\rangle + b \geq \frac{2-|B_i|}{|B_i|} - \xi_i^+ \\ |\langle w, \phi(x_{ij}) \rangle + b| \geq 1 - \xi_{ij}^* \end{cases} \quad \text{if } Y_i = 1, \quad \xi_{ij} \geq 0$$

$$-(\langle w, \phi(x_{ij}) \rangle + b) \geq 1 - \xi_{ij}^- \quad \text{if } Y_i = -1$$

sbMIL (Sound, Convex). (Bunescu and Mooney 2007) sbMIL first solves the sMIL formulation, then uses the resulting function to rank instances from least to most positive. The top (most positive) η fraction of instances from positive bags are labeled positive and the rest negative. Then a standard SVM is trained on the resulting supervised dataset.

B-KI-SVM (Complete, Convex). (Li et al. 2009)

$$\min_{w,\rho,\xi,d} \frac{1}{2} \|w\|^2 - \rho + C_1 \sum_i (\xi_i^+)^2 + C_2 \sum_i (\xi_i^-)^2,$$

$$\text{s.t.} \quad \begin{cases} \left\langle w, \sum_j d_{ij} \phi(x_{ij}) \right\rangle \geq \rho - \xi_i^+ & \text{if } Y_i = 1 \\ -\left\langle w, \frac{1}{|B_i|} \sum_j \phi(x_{ij}) \right\rangle \geq \rho - \xi_i^- & \text{if } Y_i = -1 \end{cases}, \quad \sum_j d_{ij} = 1, \quad d_{ij} \in \{0, 1\}, \quad \xi_i \geq 0$$

B Numerical Results

Here, we include numerical results for the critical difference diagrams computed in the paper. Table 2 shows results under instance accuracy and Table 3 for bag accuracy.

Table 2 Instance Accuracy Results

| Dataset | MICA | MissSVM | mi-SVM | MI-SVM | sMIL | stMIL | sbMIL | SL | NSK-AV | NSK-FS | I-KI-SVM | B-KI-SVM |
|---------|-------|---------|--------------|--------------|-------|-------|--------------|-------|--------|--------|--------------|----------|
| sival01 | 0.498 | 0.898 | 0.964 | 0.964 | 0.948 | 0.948 | 0.971 | 0.808 | 0.456 | 0.360 | 0.945 | 0.952 |
| sival02 | 0.926 | 0.890 | 0.933 | 0.947 | 0.935 | 0.935 | 0.939 | 0.509 | 0.540 | 0.352 | 0.935 | 0.890 |
| sival03 | 0.885 | 0.855 | 0.865 | 0.872 | 0.877 | 0.877 | 0.887 | 0.811 | 0.489 | 0.411 | 0.872 | 0.886 |
| sival04 | 0.411 | 0.780 | 0.954 | 0.894 | 0.846 | 0.846 | 0.908 | 0.789 | 0.778 | 0.670 | 0.867 | 0.868 |
| sival05 | 0.711 | 0.731 | 0.896 | 0.812 | 0.732 | 0.732 | 0.805 | 0.839 | 0.795 | 0.806 | 0.765 | 0.787 |
| sival06 | 0.869 | 0.854 | 0.876 | 0.899 | 0.863 | 0.863 | 0.922 | 0.640 | 0.596 | 0.697 | 0.858 | 0.859 |
| sival07 | 0.917 | 0.853 | 0.919 | 0.936 | 0.911 | 0.911 | 0.942 | 0.685 | 0.361 | 0.530 | 0.919 | 0.919 |
| sival08 | 0.905 | 0.864 | 0.955 | 0.904 | 0.866 | 0.866 | 0.917 | 0.899 | 0.547 | 0.456 | 0.872 | 0.875 |
| sival09 | 0.884 | 0.831 | 0.828 | 0.885 | 0.877 | 0.877 | 0.888 | 0.734 | 0.598 | 0.463 | 0.881 | 0.883 |
| sival10 | 0.768 | 0.930 | 0.953 | 0.908 | 0.945 | 0.945 | 0.976 | 0.855 | 0.482 | 0.411 | 0.929 | 0.947 |
| sival11 | 0.933 | 0.842 | 0.573 | 0.915 | 0.936 | 0.936 | 0.923 | 0.738 | 0.517 | 0.452 | 0.938 | 0.931 |
| sival12 | 0.864 | 0.675 | 0.880 | 0.869 | 0.810 | 0.810 | 0.865 | 0.433 | 0.756 | 0.831 | 0.862 | 0.867 |

Table 3 Bag Accuracy Results

| Dataset | <i>MCA</i> | <i>MissSVM</i> | <i>ml-SVM</i> | <i>ML-SVM</i> | <i>sMIL</i> | <i>stMIL</i> | <i>sbMIL</i> | <i>SL</i> | <i>NSK-AV</i> | <i>NSK-FS</i> | <i>I-KI-SVM</i> | <i>B-KI-SVM</i> |
|----------|--------------|----------------|---------------|---------------|-------------|--------------|--------------|--------------|---------------|---------------|-----------------|-----------------|
| musk1 | 0.511 | 0.565 | 0.848 | 0.772 | 0.750 | 0.728 | 0.859 | 0.848 | 0.913 | 0.837 | 0.870 | 0.815 |
| musk2 | 0.696 | 0.725 | 0.647 | 0.833 | 0.608 | 0.618 | 0.843 | 0.804 | 0.814 | 0.873 | 0.755 | 0.745 |
| elephant | 0.680 | 0.700 | 0.750 | 0.815 | 0.520 | 0.590 | 0.810 | 0.740 | 0.845 | 0.855 | 0.830 | 0.690 |
| fox | 0.550 | 0.600 | 0.590 | 0.585 | 0.520 | 0.515 | 0.600 | 0.645 | 0.615 | 0.580 | 0.540 | 0.555 |
| tiger | 0.535 | 0.680 | 0.770 | 0.785 | 0.590 | 0.615 | 0.800 | 0.780 | 0.825 | 0.845 | 0.725 | 0.760 |
| text1 | 0.627 | 0.557 | 0.762 | 0.650 | 0.500 | 0.500 | 0.932 | 0.728 | 0.885 | 0.873 | 0.670 | 0.642 |
| text2 | 0.500 | 0.512 | 0.595 | 0.552 | 0.500 | 0.500 | 0.618 | 0.637 | 0.682 | 0.703 | 0.520 | 0.565 |
| mountain | 0.615 | 0.660 | 0.860 | 0.855 | 0.500 | 0.500 | 0.815 | 0.830 | 0.845 | 0.860 | 0.790 | 0.795 |
| field | 0.650 | 0.635 | 0.760 | 0.805 | 0.500 | 0.500 | 0.810 | 0.725 | 0.815 | 0.830 | 0.700 | 0.745 |
| flower | 0.780 | 0.695 | 0.800 | 0.815 | 0.500 | 0.500 | 0.805 | 0.775 | 0.740 | 0.800 | 0.810 | 0.765 |
| sival01 | 0.617 | 0.575 | 0.850 | 0.875 | 0.500 | 0.500 | 0.883 | 0.558 | 0.842 | 0.883 | 0.600 | 0.758 |
| sival02 | 0.558 | 0.600 | 0.758 | 0.867 | 0.500 | 0.500 | 0.808 | 0.542 | 0.908 | 0.908 | 0.708 | 0.575 |
| sival03 | 0.833 | 0.558 | 0.733 | 0.600 | 0.500 | 0.500 | 0.750 | 0.650 | 0.908 | 0.925 | 0.650 | 0.800 |
| sival04 | 0.583 | 0.575 | 0.892 | 0.975 | 0.500 | 0.500 | 0.967 | 0.575 | 0.933 | 0.767 | 0.783 | 0.742 |
| sival05 | 0.658 | 0.692 | 0.908 | 0.917 | 0.500 | 0.500 | 0.975 | 0.650 | 0.958 | 0.983 | 0.842 | 0.942 |
| sival06 | 0.725 | 0.492 | 0.775 | 0.933 | 0.500 | 0.500 | 0.908 | 0.542 | 0.800 | 0.883 | 0.450 | 0.675 |
| sival07 | 0.733 | 0.567 | 0.833 | 0.900 | 0.500 | 0.500 | 0.925 | 0.533 | 0.867 | 0.892 | 0.817 | 0.875 |
| sival08 | 0.840 | 0.487 | 0.798 | 0.807 | 0.504 | 0.504 | 0.899 | 0.655 | 0.966 | 0.941 | 0.714 | 0.731 |
| sival09 | 0.700 | 0.550 | 0.525 | 0.658 | 0.500 | 0.500 | 0.817 | 0.583 | 0.883 | 0.950 | 0.583 | 0.750 |
| sival10 | 0.500 | 0.600 | 0.817 | 0.667 | 0.500 | 0.500 | 0.925 | 0.750 | 0.825 | 0.817 | 0.642 | 0.783 |
| sival11 | 0.808 | 0.517 | 0.500 | 0.533 | 0.500 | 0.500 | 0.700 | 0.583 | 0.933 | 0.900 | 0.683 | 0.683 |
| sival12 | 0.967 | 0.575 | 0.917 | 0.950 | 0.500 | 0.500 | 0.958 | 0.517 | 0.908 | 0.917 | 0.950 | 0.967 |