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

Abstract

A variety of support vector machine (SVM) approaches have been applied to the multiple instance (MI) setting. We theoretically compare these approaches using soundness and completeness properties to describe the feasible regions of MI SVMs. Using hardness results, we show that unless P = NP, there is no SVM formulation for MI classification that is sound, complete, and convex. We then analyze practical trade-offs between these properties by empirically comparing the performance of several representative algorithms. We find that (i) algorithms that are both sound and complete are more accurate than those that are only sound, but (ii) some unsound but complete set kernel based approaches produce classifiers with the best performance and (iii) convex formulations work better than non-convex ones for our domains. Further experiments show that the surprising accuracy of set kernels is partly due to their ability to exploit bag structure and size information that other approaches cannot use, and the benefit of convexity is partly due to the failure of random restarts as an effective technique in these domains.

1. Introduction

In the drug activity prediction domain, one goal is to classify molecules as being "active" or "inactive." That is, given a particular molecule (e.g. a candidate drug), can we predict whether the molecule bind to a target such as a receptor or other protein? While it is possible to represent molecules as simple feature vectors based on their structure (Cramer et al., 1988), flexibility of chemical bonds allows molecules to exist in multiple shapes, called *conformations*, in solution. Therefore, if a molecule is observed to be active, that implies that at least one conformation binds to the target. On the other hand, inactivity of a molecule means that no conformation is active.

The multiple-instance (MI) learning framework was motivated by the above problem (Dietterich et al., 1997), and explicitly encodes this relationship between an observed label and a set of instances that may be responsible for that label. In particular, a dataset is treated as a set of labeled *bags*, each of which contains several *instances*, which are typically 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 learning task is to produce a classifier that can accurately label new bags.

Numerous supervised learning approaches, such as decision trees (Blockeel et al., 2005), artificial neural networks (Ramon & Raedt, 2000; Zhou & Zhang, 2002), Gaussian models (Maron, 1998; Zhang & Goldman, 2001) and logistic regression (Xu & Frank, 2004; Ray & Craven, 2005), have been extended to the MI setting. In particular, kernel methods such as support vector machines (SVMs) have been modified to handle MI data, and are the focus of this study (Andrews et al., 2003; Bunescu & Mooney, 2007; Zhou & Xu, 2007; Mangasarian & Wild, 2008). Below, we describe notions of soundness and completeness for algorithms that produce large margin hyperplane classifiers for MI data, and characterize a variety of existing techniques in terms of these properties. Using hardness results, we show that no SVM optimization program for MI classification can be sound, complete, and convex, unless P = NP. To investigate the trade-offs between soundness, completeness, and convexity, we empirically evaluate and contrast the performance of several algorithms on MI datasets. We generally find that both soundness and completeness lead to better performance. However, some unsound techniques outperform those that are both sound and complete, and we argue that this is because they are able to encode additional information such as bag structure to solve the MI classification problem.

2. Complexity and Learnability of MI Hyperplanes

We begin with an analysis of the complexity of MI classification via hyperplanes. First, we will only consider classification in the instance space or the feature space of a kernel defined over instances. However, these results also have implications for approaches that use set

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

kernels (Gärtner et al., 2002) as we show below.

Definition 1. *For a MI problem (B, Y) where B is a list of bags, each bag B_i is a set of instances $x_{ij} \in \mathbb{R}^n$, and Y is a list of labels with $Y_i \in \{-1, 1\}$ and $|Y| = |B|$, a* classifying hyperplane *defined by (w, b) with $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ 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 previous paper has shown a specific margin-maximizing formulation of MI classification to be NP-complete (Kundakcioglu et al., 2010). A more general result is possible[1]:

Theorem 1. *Given a MI problem (B, Y), a set of bags with $|B_i| \leq k$, $k \geq 3$, the decision problem of determining whether there exists a hyperplane consistent with (B, Y) (*MI−CONSIS*) is* NP*-complete.*

The same result holds for a stronger definition of consistency that requires the extra condition $\forall x_{ij} \in B_i : \langle w, x_{ij} \rangle + b \geq 1 \vee \langle w, x_{ij} \rangle + b \leq -1$ (i.e. each instance has a label). Now, we show that general forms of margin-maximizing and error-minimizing MI classification are also NP-complete.

Definition 2. *A hyperplane (w, b) is* μ-margin consistent *with (B, Y) if it is consistent and $m(w, b) \geq \mu$, where m is any polynomially computable function satisfying $\forall w, b : m(w, b) \geq 0$ that measures the margin of the hyperplane (e.g. $m(w, b) = \|w\|_2^{-1}$).*

Definition 3. *A hyperplane (w, b) is* ϵ-error consistent *with (B, Y) if for each $B_i \in B$:*

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

with $\xi \geq 0$ and $e(\xi) \leq \epsilon$, where ξ is a vector of the ξ_{ij} used in the above conditions, and e is any polynomially computable function satisfying $\forall \xi \neq 0 : e(\xi) > 0$ and $e(0) = 0$ that measures classification error (e.g. $e(\xi) = \|\xi\|_1$).

It follows that the problem of deciding whether there exists a μ-margin and/or ϵ-error consistent hyperplane for (B, Y, μ, ϵ) is also NP-complete. Given the polynomial computability of m and e, we can verify a certificate in polynomial time, and MI−CONSIS can be trivially reduced to either or a combination of these problems by using $\mu, \epsilon = 0$. Therefore, finding a margin-maximizing, error-minimizing MI classifying hyperplane (in the sense described above) is NP-hard. A result from optimization theory states that the family $\mathcal{C}$ of convex optimization programs including linear and quadratic programs are polynomially solvable (Ben-Tal & Nemirovskiĭ, 2001), so we have:

Corollary 1. *If* P $\neq$ NP*, there is no convex program in $\mathcal{C}$ that always finds a margin-maximizing, error-minimizing MI classifying hyperplane.*

Finally, the complexity results above allow us to show that MI concepts over arbitrary distributions are not PAC-learnable with classifying hyperplanes. Previous work (Auer et al., 1997) reduced PAC-learning axis parallel rectangles (APR) for MI concepts over arbitrary distributions to PAC-learning DNF formulas. 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 & Kalai, 1998). Some recent results give a bound on the γ-Fat shattering dimension of MI learning, showing that MI concepts are learnable via instance-based SVMs when the underlying instance problem is learnable (Sabato et al., 2010; Sabato & Tishby, 2011). Because we can reduce MI−CONSIS to an algorithm that PAC-learns MI concepts with hyperplanes (see the appendix for details), we could produce a RP algorithm to solve MI−CONSIS. Therefore, as for APR, the following result holds:

Proposition 1. *If* RP $\neq$ NP*, then there is no algorithm $\mathcal{A}$ that (for arbitrary bag distributions) PAC-learns MI concepts using the hypothesis space of hyperplanes.*

3. Theoretical Analysis of MI SVM Approaches

Many MI SVM techniques attempt to modify the standard supervised SVM quadratic program:

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

Several approaches formulate non-convex optimization programs that are solved heuristically (Andrews et al., 2003; Bunescu & Mooney, 2007; Mangasarian & Wild, 2008), while other approaches use set kernels as described in (Gärtner et al., 2002) to formulate convex quadratic programs (Bunescu & Mooney, 2007; Gärtner et al., 2002). However, as shown above, no convex program can always solve the MI classification problem, unless P = NP. Therefore, we analyze

[1] We include a proof in the appendix and some unpublished work contains a similar result and proof (Diochnos et al., 2011)

approaches with respect to two properties, *soundness* and *completeness*.

Definition 4. *An optimization program that produces a margin-maximizing, $e(\xi)$-minimizing classifier is* sound *iff all feasible points with the added constraint $e(\xi) = 0$ are consistent MI solutions, and is* complete *iff all consistent solutions are feasible with $e(\xi) = 0$.*

Intuitively, we would like all consistent MI solutions to be feasible points in an optimization program with zero slack (completeness) and vice versa (soundness). We analyze several techniques from this perspective.

SIL. The single instance learning (SIL) approach assigns each instance the label of its bag, creating a supervised learning problem at the expense of mislabeling negative instances in positive bags. The SIL technique is sound, since each feasible solution is consistent with the MI assumption. However, there are clearly MI solutions that do not require all positive bag instances to be positively classified, so SIL is not complete. Since SIL uses a standard SVM formulation after instance labeling, it is convex.

mi-SVM. The mixed integer SVM (mi-SVM) formulation uses standard SVM constraints, but leaves the y_i's as unknowns over $\{-1, +1\}$ for instances in positive bags (Andrews et al., 2003). It also adds a constraint for each positive bag: $\sum_i \frac{y_i+1}{2} \geq 1$, where y_i is the unknown label of the i^{th} instance in this positive bag. This ensures that at least one instance label in each positive bag is positive. The mi-SVM approach uses a mixed integer formulation to capture the MI assumption, so it is not convex. Because mi-SVM forces a label assignment on each instance, the feasible solutions are consistent, but consistent solutions (in the sense of Definition 1) are not all feasible. However, mi-SVM is complete under the alternate definition of consistency that requires each instance to have a label.

msMIL. The minimum slack MI learning (msMIL) algorithm is a novel approach that uses the minimum slack variable for each positive bag in the objective function, along with the slack variable of every negative instance. This method is sound because if at least one positive instance is correctly classified with zero slack for a bag, then the solution is feasible with zero slack for that bag in the objective function. On the other hand, if slack is required for a positive bag, then every instance in the bag must have some nonzero slack and the hyperplane is not consistent. However, since the minimization term in the objective function is not convex, msMIL uses the concave-convex procedure (CCCP) to iteratively linearize and then solve a quadratic program. The dual of the msMIL program is the same as that for a traditional SVM, except the constraints on the dual variables for positive bag slacks become $0 \leq \alpha \leq C \cdot \nabla \min(\xi)$, where the gradient of the minimum function with respect to slack ξ_{ij} is 0 if ξ_{ij} is not a minimum within the i^{th} bag, or $1/n_i$ if it is one of the n_i minimizers of the i^{th} bag. The effect is that at each iteration, instances with non-minimal slack are ignored as potential support vectors. The msMIL approach is similar to multiple-instance SVM (MI-SVM), discussed in the appendix, except that it possibly selects more than one instance to (partially) represent a bag (Andrews et al., 2003).

NSK. Subsequent approaches are all *instance-based* in that the corresponding SVM formulations act on instances. Because kernel functions implicitly define a feature space mapping on arbitrary objects, another approach is to define kernels over entires sets of instances to classify entire bags. A particular *set-based* approach, the normalized set kernel (NSK), computes a kernel of two sets or bags as the sum of the pairwise instance kernel of elements of the bag (Gärtner et al., 2002). The kernel can be normalized either by the length of the vector in the resulting feature space (*feature space normalization*), or by the number of elements in the bags (*averaging normalization*).

Gärtner et al. prove a completeness result about the set kernel; namely, their Lemma 4.2 states that if an underlying instance concept is separable with a hyperplane in an instance kernel feature space, then there is some related set kernel (called an MI kernel) that separates bags (2002). The same paper also has a corresponding soundness result (Lemma 4.3), and concludes that a MI concept is separable by a MI kernel if and only if the underlying instance concept is separable by the instance kernel (Theorem 4.4). However, in light of Corollary 1, this would imply that $\mathsf{P} = \mathsf{NP}$, since the resulting technique would provide an algorithm to solve $\mathsf{MI{-}CONSIS}$ with a convex quadratic program. The soundness result does not hold, since by constructing an instance separator via the MI separator acting on singleton bags, $f_{\text{I}}(z) = f_{\text{MI}}(\{z\})$, the function f_{I} can no longer generally be written as a dot product in the instance kernel feature space.

Furthermore, a simple 1-D example shows that set kernels are generally not sound with respect to the instance kernel. Consider a synthetic dataset with negative bags $\{0\}$ and $\{1\}$, and a positive bag $\{0.25, 0.5, 0.75\}$. This is not linearly separable in the instance space. With a linear instance kernel, $\sum_{i,j} k(x_i, x_j) = \sum_{i,j} \langle x_i, x_j \rangle = \left\langle \sum_i x_i, \sum_j x_j \right\rangle$, so the feature space map ϕ simply sums instances within a bag. Therefore, ϕ leaves the negative bags unchanged, but maps the positive bag to the sum of the

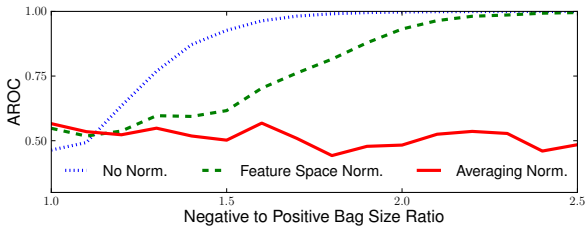

Figure 1. With certain types of normalization, set kernels can separate bags of difference sizes even when there is no underlying MI concept.

instances, 1.5, so the dataset is now separable.

Finally, another form of unsoundness arises for set kernels due to the effects of bag size. For example, consider a MI problem in which all bag instances are identical (say $x_{ij} = 1 \in \mathbb{R}$), but positive bags have size 10 while negative bags have size 5. Then for an unnormalized linear kernel, the feature space 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. Some forms of set kernel normalization are successful at eliminating the effects of bag size. For example, averaging normalization explicitly divides by the bag size, which eliminates the difference between positive and negative bags in the feature space for the example above. However, as Figure 1 illustrates using synthetic datasets, feature space normalization permits learning from bags sizes. In these datasets, each instance has 25 features, which are drawn i.i.d. from the standard normal distribution. 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.

sMIL. The sparse MI learning (sMIL) algorithm (Bunescu & Mooney, 2007) uses a hybrid of set and instance kernels, and introduces a balancing constraint on the average instance label within a bag, assuming that in the worst case all but one instance in the bag is negative:

$$\langle w, \phi(x_{ij}) \rangle + b \leq -1 + \xi_{ij} \qquad \forall x_{ij} \text{ with } Y_i < 0$$

$$\left\langle w, \frac{\phi(B_i)}{|B_i|} \right\rangle + b \geq \frac{2 - |B_i|}{|B_i|} - \xi_i \qquad \forall B_i \text{ with } Y_i > 0$$

Here, $\phi(B_i)$ denotes the feature space mapping induced by the set kernel for bags, and $\phi(x_{ij})$ the equivalent mapping for negative instances.

The sMIL approach is convex, but is neither sound nor complete. The counterexample to soundness is shown in Figure 2 (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 requiring any slack. Therefore, this solution is feasible and optimal without slack, 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 2 (right). While the solution is consistent, it is not feasible without slack because the averages of the instances in the large positive bag lies below the separating line and therefore do not satisfy the balancing constraint.

stMIL. The sparse transductive MI learning (stMIL) formulation includes the sMIL constraints, along with the additional constraint $|\langle w, x_{ij} \rangle + b| \geq +1 - \xi_{ij}$ for every instance x_{ij}, which force instances within bags to be far away from the margin (Bunescu & Mooney, 2007). The addition of these constraints makes the problem non-convex. But like mi-SVM, these constraints imposes a label on every instance, so stMIL is sound. To see why, suppose the constraints are satisfied, but for some positive bag all instances are classified with a negative label. By the transductive constraints, all instances in that bag are assigned a label that is at most -1. Therefore, for the average label, the following inequality holds: $\left\langle w, \frac{\phi(B_i)}{|B_i|} \right\rangle + b \leq -1$. However, from the balancing constraint we also have $\left\langle w, \frac{\phi(B_i)}{|B_i|} \right\rangle + b \geq \frac{2-|B_i|}{|B_i|} > -1$, which is a contradiction, so at least one instance in each positive bag must be positively labeled. The scenario in Figure 2 (right) is also a counterexample to the completeness of stMIL because the instances in the large bag satisfy the transductive constraint.

In addition to the algorithms described above, we apply a similar analysis to MI learning by semi-supervised SVM (MissSVM) (Zhou & Xu, 2007), the MI classification algorithm (MICA) (Mangasarian & Wild, 2008), and sparse balanced MI learning (sbMIL) (Bunescu & Mooney, 2007) (see appendix for details). Figure 3 summarizes the theoretical analysis in this section.

4. Empirical Evaluation

Given the properties possessed by (or lacking in) the various classification algorithms analyzed above, it is

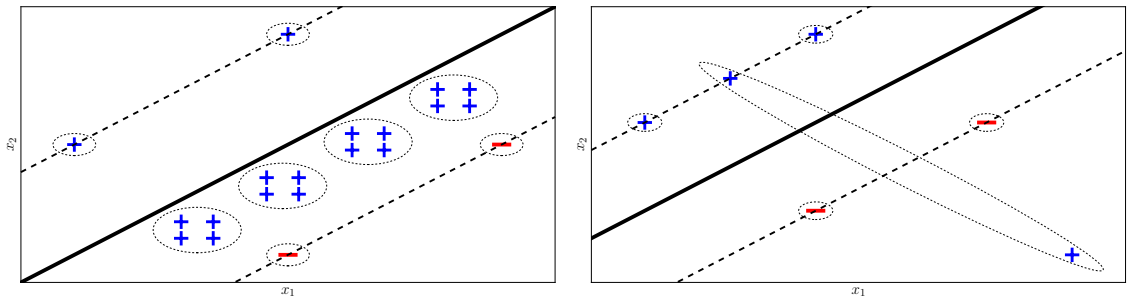

Figure 2. Synthetic datasets illustrating when soundness and/or completeness fail for some techniques. **Left** shows when a sMIL solution without slack allows a misclassification of an arbitrary number of bags whose average lies close to the wrong size of the classifier. **Right** shows a valid MI separator that requires slack for sMIL and stMIL.

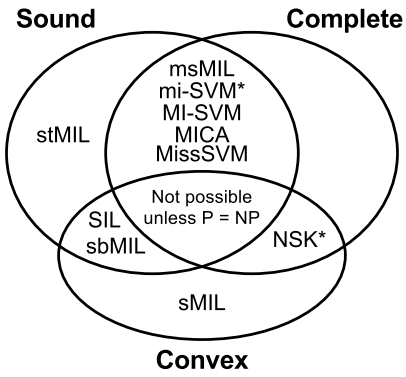

Figure 3. Soundness, completeness and convexity of various MI classification algorithms (*see description for caveats). By Corollary 1, no algorithm possesses all three properties unless P = NP.

natural to wonder whether one property is more important than another for classification accuracy. In this section we perform a detailed empirical comparison of several algorithms on a variety of real world problems. We select these algorithms here to provide good coverage of the properties in Figure 3. [2]

Datasets. Five standard MI datasets are used for evaluation. The MUSK1 and MUSK2 datasets come from the drug activity prediction domain, which motivated the creation of the MI classification framework (Dietterich et al., 1997; Frank & Asuncion, 2010). The three animal datasets (Elephant, Tiger, and Fox) are from the content-based image retrieval (CBIR) domain (Andrews et al., 2003). The properties of these datasets are summarized in Table 2.

Methodology. We implement each technique in Python (van Rossum, 1995) using NumPy (Ascher et al., 2001) and SciPy (Jones et al., 2001) for general matrix computations, and the CVXOPT library (Dahl & Vandenberghe, 2009) for solving quadratic programs. For each dataset, we use ten-fold cross validation with the same folds across all techniques and area under the receiver operating characteristic curve (AROC) as the performance metric. We evaluate linear, quadratic, and radial basis function (RBF) kernels. We fix the scale parameter γ of the RBF kernel to 10^{-5} using the heuristic that $\gamma \approx \frac{1}{2f^2}$ is a good choice for data with f features (Gärtner et al., 2002). Grid search is used to choose the regularization parameter C over several orders of magnitude, $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, using a five-fold inner validation to evaluate each value. For techniques that rely on iteratively solving quadratic programs, iteration continues at most 50 times or until the change in objective function value falls below 10^{-6}.

Table 2. Dataset Statistics

Dataset	Bags	Average Bag Size		Features
		Positive	Negative	
MUSK1	92	4.4	6.0	166
MUSK2	102	26	89	166
Elephant	200	7.6	6.3	230
Tiger	200	5.4	6.8	230
Fox	200	6.5	6.7	230

The results are shown in Table 1. The primary insights from these experiments are: (i) techniques that are both sound and complete tend to outperform those that are only sound, while those that are only complete outperform both sound and complete methods (ii) set kernel based approaches tend to outperform instance kernel approaches, and (iii) convex methods tend to outperform nonconvex methods. Further experiments indicate that the good results of set kernels may partly be due to their ability to exploit information about bag

[2] We have implemented and tested other algorithms as well, not shown due to lack of space. The full set of results is available in the appendix. Our conclusions hold in general over those results as well.

Table 1. Experimental results for select techniques, using AROC to measure classification performance (with best values indicated in boldface). Additional columns show results with 15 random restarts for msMIL and mi-SVM. Both averaging (AV) and feature space (FS) normalization were used with the set kernel.

Dataset	Kernel	msMIL	msMIL (w/RR)	mi-SVM	mi-SVM (w/RR)	stMIL	SIL	sMIL	NSK-AV	NSK-FS
MUSK1	linear	0.815	0.843	0.852	0.873	0.795	0.869	0.888	**0.915**	0.910
	quadratic	0.877	0.899	0.900	0.755	0.815	0.888	0.887	0.920	**0.952**
	rbf	0.689	0.569	0.791	0.790	0.544	0.558	0.685	0.853	**0.854**
MUSK2	linear	0.836	0.790	0.791	0.808	0.650	0.806	0.785	**0.915**	0.888
	quadratic	0.864	0.849	0.857	0.759	0.831	0.849	0.875	**0.942**	0.932
	rbf	0.626	0.551	0.787	0.610	0.642	0.513	0.661	**0.840**	0.839
Tiger	linear	0.885	0.865	0.853	0.725	0.811	0.837	0.880	0.895	**0.906**
	quadratic	0.869	0.886	0.807	0.704	0.823	0.825	0.902	0.887	**0.903**
	rbf	0.838	0.838	0.732	0.694	0.577	0.653	0.847	0.872	**0.873**
Elephant	linear	0.877	0.890	0.870	0.851	0.842	0.900	0.910	**0.923**	0.920
	quadratic	0.897	0.911	0.864	0.774	0.852	0.888	0.890	0.881	**0.918**
	rbf	0.846	0.845	0.832	0.805	0.548	0.658	0.860	0.869	**0.875**
Fox	linear	0.614	0.584	0.551	0.513	0.524	0.602	0.580	0.601	**0.657**
	quadratic	0.579	0.602	0.549	0.566	0.584	0.587	0.603	0.614	**0.636**
	rbf	0.569	0.548	0.516	0.521	0.512	0.514	0.565	**0.637**	0.633

structure and bag size, which is generally ignored by instance based methods. Further, part of the benefit of convexity in these problems comes from the fact that random restarts are less effective in these problems for nonconvex approaches. Below we discuss these observations in detail.

Effect of Soundness and Completeness. Comparing methods which are only sound to those which are both sound and complete, we see results that align with our intuition. Namely, when techniques such as msMIL and mi-SVM are complete and allow all consistent MI solutions, they more frequently outperform methods that restrict the feasible region for classifiers. However, set kernel techniques that are only complete often outperform sound and complete techniques. Below, we offer an explanation of this unexpected result by describing how set kernels might exploit information not accessible to instance-based approaches.

Within sound and complete methods, msMIL tends to outperform mi-SVM, and msMIL benefits more often from random restarts. The CCCP optimization of msMIL might make it less prone to over-fitting than the iterative label updates of mi-SVM. Furthermore, forcing each instance to have a label as mi-SVM does might over-constrain the problem, making it difficult to find good solutions in practice.

Effect of Bag Structure. Although the NSK is not sound, it often outperforms methods that are both sound and complete. To explain this counterintuitive result, we hypothesize that the set kernel is capable of using information about bags to which instance-based approaches do not have access.

For example, consider a set kernel using a linear instance kernel and averaging normalization. In this case, a bag is essentially mapped to the average of its instances. Now suppose that the distribution of instances in each positive bag B_i has an expected value that is linearly related to the prime instance, $x_i^\star$, i.e. $\mathbb{E}\left[P(x_{ij} \mid x_i^\star)\right] = ax_i^\star + \mathbf{c}$, with $a \neq 0$. For negative bags, we can assume that at least one instance acts as a "prime instance" with the same relationship to the mean of the bag distribution.

Under this assumption, for large bag sizes, averages of bag instances in the set kernel feature space, denoted $\overline{B_i}$, will be approximately $\overline{B_i} \approx ax_i^\star + \mathbf{c}$. Then given $f(x_{ij}) = \langle w, x_{ij}\rangle + b$, a hyperplane separating instances, if we pick a linear function in the set kernel feature space given by $f(B_i) = \left\langle \frac{w}{a}, \overline{B_i}\right\rangle + \left(b - \frac{\langle w,\mathbf{c}\rangle}{a}\right)$, then its value will be approximately: $f(B_i) \approx \left\langle \frac{w}{a}, ax_i^\star + \mathbf{c}\right\rangle + \left(b - \frac{\langle w,\mathbf{c}\rangle}{a}\right) \approx \langle w, x_i^\star\rangle + b$. Therefore, when there is a relationship between prime instances and instance distributions within bags, the NSK can take advantage of this information. However, other techniques (even those that are sound and complete) that select single instances from bags have no mechanism to learn from bag distributions.

To verify this hypothesis, we analyzed a collection of SIVAL image classification datasets that have been annotated with instance labels so that we could identify prime instances (Settles et al., 2008). We used the linear set kernel. For bags with multiple positive instances, we chose the most positively labeled positive instance as the prime instance after a classifier had been found. Once a prime instance was selected from the positive bags, we computed the R^2 coefficient of

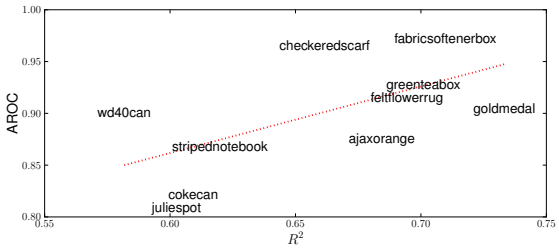

Figure 4. These results show an association between the degree of linear relationship between prime instances and bag averages, and the classification performance of a set kernel with averaging normalization across several image categorization datasets.

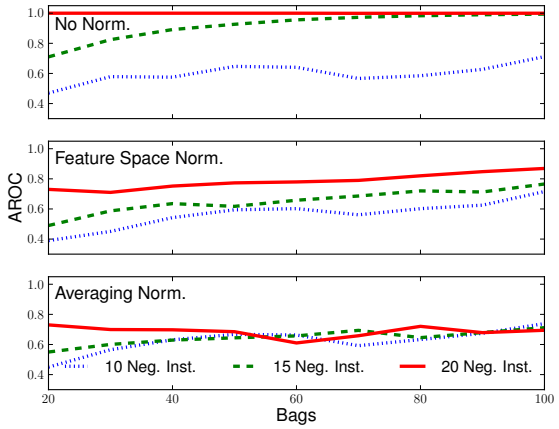

Figure 5. These learning curves for synthetic datasets demonstrate that if a MI concept is present, a discrepancy in bag size can improve classification performance of set kernels with certain types of normalization. Each dataset contains positive bags with 10 instances, but the number of negative instances per bag varies across datasets.

determination from a least squares multiple linear regression between bag averages and prime instances. In Figure 4, we plot the AROC against the R^2 for several datasets in which the linear set kernel found a good classifier (high AROC). We observe that indeed, for these datasets, there is a general association ($r = 0.61$) between a strong linear relationship between bag average and prime instance, and set kernel classifier accuracy. This indicates that at least in some cases good performance may be due to the NSK being able to take advantage of bag structure.

Effect of Bag Size. Another source of information that set kernels may be able to use is a *difference in the size* of positive and negative bags, depending on the type of normalization used. Figure 5 demonstrates how this can occur using synthetic datasets. The plot shows learning curves for set kernels with different normalization as the number of bags in the training set varies. Each positive bag has 9 instances with 5 features drawn from the standard normal distribution, and a prime instance also drawn from a normal distribution with mean 1. The number of negative instances per bag (again with features drawn from the standard normal distribution) varies across datasets. We see that the bag size discrepancy makes no difference under averaging normalization, but with feature space normalized or unnormalized kernels, more accurate classifiers are found from smaller data sets.

Our results for real datasets show that bag size differences are not always beneficial for feature space normalization. For example, MUSK2 has the largest discrepancy between bag sizes, yet averaging normalization leads to better performance. In this case, because bag sizes are so large, averaging normalization may be better able to learn from bag structure as discussed above. However, for other datasets where feature space normalization works better than averaging normalization this effect might be present.

Effect of Random Restarts. Comparing convex and non-convex approaches, one can see that convex approaches (even those that are neither sound nor complete) often outperform their non-convex counterparts. One possible explanation is that non-convex approaches rely on heuristic optimization techniques that are only likely to converge to local optima. To explore this possibility, we ran the non-convex approaches again using 15 random restarts (using the instance labeling heuristic as one of the restarts), with results listed in the "RR" columns of Table 1.

We observe that the effects of random restarts are mixed; performance both improves and degrades across datasets for each technique. We believe this behavior is likely due to over-fitting, because many techniques only select one instance from each bag to find a classifier, so there are fewer data points than the dimensionality of the dataset (see Table 2). Learning theory results show that the maximum margin formulations of MI-SVM and similar approaches are theoretically justified and will generalize when datasets are sufficiently large (Sabato & Tishby, 2011). Therefore, the behavior we see might be restricted to small datasets such as the ones we use.

Within the set of convex approaches, set kernels approaches such as NSK outperform the instance kernel approaches like SIL. We suspect that the properties of set kernels discussed above contribute to the relatively good performance of convex techniques.

Other Observations. In terms of time and space, instance kernel approaches are much more expensive than set kernel approaches, since kernel sizes are $O(n^2)$

in terms of the number of instances rather than bags. Runtime also increases significantly for instance methods due to the increased number of variables in the optimization program, as well as the amount of space needed for storing support vectors.

5. Conclusion

In this work, we formally specify soundness and completeness properties desired in algorithms for MI classification via hyperplanes. Using hardness results, we show that unless $\mathsf{P} = \mathsf{NP}$, there is no MI SVM formulation that is sound, complete, and convex. We analyze a variety of existing techniques to determine which properties they possess, and we evaluate their performance empirically to characterize the trade-offs between properties. In general, we see that soundness and completeness lead to better performance over techniques that are only sound. However, we also show that set kernels, which are not sound, often have the best performance. We hypothesize that set kernels can use additional information not available to instance-based approaches, such as bag structure and size. In future work, we plan to further investigate the properties of set kernels as well as their potential applications to other MI learning problems.

References

Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. In *NIPS*, volume 15, pp. 561–568. MIT Press, 2003.

Ascher, D., Dubois, P., Hinsen, K., Hugunin, J., and Oliphant, T. *Numerical Python*. Lawrence Livermore National Laboratory, Livermore, CA, 2001.

Auer, P., Long, P., and Srinivasan, A. Approximating hyper-rectangles: Learning and pseudo-random sets. In *J. of Comp. & Sys. Sci.*, pp. 314–323. ACM, 1997.

Ben-Tal, A. and Nemirovskiĭ, A.S. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. MPS-SIAM Series on Optimization. SIAM, 2001.

Blockeel, H., Page, D., and Srinivasan, A. Multi-instance tree learning. In *ICML '05*, pp. 57–64, 2005.

Blum, A. and Kalai, A. A note on learning from multiple-instance examples. *Mach. Learn.*, 30(1):23–29, 1998.

Bunescu, R and Mooney, R. Multiple instance learning from sparse positive bags. In *ICML '07*, pp. 105–112, Corvallis, OR, USA, 2007.

Cramer, R. D., Patterson, D. E., and Bunce, J. D. Comparative molecular field analysis (CoMFA). Effect on binding of steroids to carrier proteins. *J. of the American Chemical Society*, 110(18):5959–5967, 1988.

Dahl, J. and Vandenberghe, L. CVXOPT: A python package for convex optimization, 2009.

Dietterich, T. G., Lathrop, R. H., and Lozano-Perez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

Diochnos, D., Sloan, R., and Turán, Gy. On multiple-instance learning of halfspaces. http://homepages.math.uic.edu/~diochnos/research/publications/dst-MIL-Halfspaces.pdf, 2011.

Frank, A. and Asuncion, A. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

Gärtner, T., Flach, P., Kowalczyk, A., and Smola, A. Multi-instance kernels. In *ICML '02*, pp. 179–186. Morgan Kaufmann, 2002.

Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, 2001.

Kundakcioglu, O., Seref, O., and Pardalos, P. Multiple instance learning via margin maximization. *Applied Numerical Mathematics*, 60(4):358–369, 2010.

Mangasarian, O. and Wild, E. Multiple instance classification via successive linear programming. *J. of Optimization Theory and Applications*, 137:555–568, 2008.

Maron, O. *Learning from Ambiguity*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1998.

Ramon, J. and Raedt, L. De. Multi instance neural networks. In *Proc. ICML '00 workshop on Attribute-Value and Relational Learning.*, 2000.

Ray, S. and Craven, M. Supervised versus multiple instance learning: an empirical comparison. In *ICML '05*, pp. 697–704, New York, NY, USA, 2005. ACM.

Sabato, S. and Tishby, N. Multi-instance learning with any hypothesis class. *CoRR*, abs/1107.2021, 2011.

Sabato, S., Srebreo, N., and Tishby, N. Reducing label complexity by learning from bags. *AI*, pp. 685–692, 2010.

Settles, B., Craven, M., and Ray, S. Multiple-instance active learning. In *NIPS*, pp. 1289–1296. MIT Press, 2008.

van Rossum, G. *Python tutorial*. Centrum voor Wiskunde en Informatica (CWI), May 1995.

Xu, X. and Frank, E. Logistic regression and boosting for labeled bags of instances. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pp. 272–281, Sydney, 2004. Springer-Verlag.

Zhang, Qi and Goldman, Sally. EM-DD: An improved multiple-instance learning technique. In *NIPS*, pp. 1073–1080. MIT Press, 2001.

Zhou, Z. and Xu, J. On the relation between multi-instance learning and semi-supervised learning. In *ICML '07*, pp. 1167–1174, Corvallis, OR, USA, 2007.

Zhou, Z.-H. and Zhang, M.-L. Neural networks for multi-instance learning. In *Proceedings of the International Conference on Intelligent Information Technology*, 2002.