Noname manuscript No.
(will be inserted by the editor)

Multiple Instance Classification via Quadratic Programming

Emel Şeyma Küçükaşcı · Mustafa Gökçe Baydoğan · Z. Caner Taşkın

Received: date / Accepted: date

Abstract Multiple instance learning (MIL) is a variation of supervised learning, where data consists of labeled bags and each bag contains a set of instances. Unlike traditional supervised learning, labels are not known for the instances in MIL. Existing approaches in the literature make use of certain assumptions regarding the instance labels and propose mixed integer quadratic programs, which introduce computational difficulties. In this study, we present a novel quadratic programming (QP)-based approach to classify bags. Solution of our QP formulation links the instance-level contributions to the bag label estimates, and provides a linear bag classifier along with a decision threshold. Our approach imposes no additional constraints on relating instance labels to bag labels and can be adapted to learning applications with different MIL assumptions. Unlike existing specialized heuristic approaches to solve previous MIL formulations, our QP models can be directly solved to optimality using any commercial QP solver. Our computational experiments show that proposed QP formulation is efficient in terms of solution time, overcoming a main drawback of previous optimization algorithms for MIL. We demonstrate the classification success of our approach compared to the state-of-the-art methods on a wide range of real world datasets.

Keywords Multiple instance learning · Classification · Quadratic programming

E. Ş. Küçükaşcı
Department of Industrial Engineering, Istanbul Commerce University, İstanbul, Turkey
E-mail: eskucukasci@ticaret.edu.tr

M. G. Baydoğan
Department of Industrial Engineering, Boğaziçi University, İstanbul, Turkey
E-mail: mustafa.baydogan@boun.edu.tr

Z. C. Taşkın
Department of Industrial Engineering, Boğaziçi University, İstanbul, Turkey
E-mail: caner.taskin@boun.edu.tr

1 Introduction

Most data mining approaches focus on solving classification problems using machine learning and pattern recognition techniques. Classification tasks require input samples with given outputs, known as the class labels. In multiple instance learning (MIL), instances are grouped into bags and a class label is known for each bag, whereas the instance labels are not fully provided. The data representation and learning setup of MIL are in alignment with many real world applications. Current research areas of MIL include image classification, drug activity prediction, text mining and many others [5]. In these applications, global descriptions of the objects are decomposed into multiple parts. When objects are represented by multiple parts, only some parts may be relevant for classification. In addition, it is expensive and time consuming to collect true labels of parts individually. MIL paradigm provides an opportunity to solve classification problem under these circumstances.

For instance, consider sample images from Corel image classification dataset [6] in Figure 1. Under MIL scenario, images correspond to bags and patches sampled from the images correspond to the instances. In this example, images are classified either as positive or negative based on the presence of a horse on its patches as shown in Figure 1. Only some patches of an image are informative for classification and it is sufficient to label the whole image instead of the individual instances.

Figure 1. An illustration of MIL setting for image classification. Images on the left with located horses inside the red rectangles are classified as positive whereas the other images form the negative class.

Unknown instance labels and uncertainty on the bag formations contribute to the difficulty of MIL problem. Success of the MIL algorithms depends on their capability of capturing the internal structures of bags. The most common way of relating bag labels to the individual instance labels is introduced as standard MIL assumption in the first MIL application [8] and is widely used in several methods. The standard MIL assumption states that label of a bag is positive if and only if it contains at least one positive instance, otherwise the bag is negatively labeled.

In Figure 2, a regular input data with 12 instances and 3 features is used to form a MIL data with 3 bags following the standard MIL assumption.

Although it is embraced by many methods, standard assumption is considered to be restrictive for some MIL applications. For example, consider a document retrieval application, where the bags are articles and multiple sections extracted from them are the instances. The aim is to detect whether an article is about a specific subject (e.g. finance) or not. A section including the predetermined words and word combinations makes this section a positive instance. However, articles that are not relevant may also contain these words in a particular section (e.g. including financial terms in the introduction). Thus, standard MIL assumption is not well suited to this problem. Generalized MIL [1,12,34] is formalized to describe MIL scenarios other than the standard MIL under various constraints [34]. Under generalized MIL, collective MIL assumption [12] models equal contribution of instances to the bag label. The idea is to derive a bag-level classifier from an instance-level decision function by averaging the learning results in underlying instance-feature space.

Instance Class	Instance ID	Feature 1	Feature 2	Feature 3
1	1	42	-198	-109
1	2	42	-191	-142
0	3	42	-191	-142
0	4	42	-198	-110
0	5	40	-39	30
0	6	36	-198	-166
0	7	43	69	30
0	8	36	-198	-166
0	9	24	80	29
0	10	38	-29	27
0	11	17	77	30
0	12	17	77	31

Regular

Bag Class	Bag ID	Feature 1	Feature 2	Feature 3
1	1	42	-198	-109
	1	42	-191	-142
	1	42	-191	-142
	1	42	-198	-110
0	2	40	-39	30
	2	36	-198	-166
	2	43	69	30
	2	36	-198	-166
0	3	24	80	29
	3	38	-29	27
	3	17	77	30
	3	17	77	31

Multiple Instance

Figure 2. Multiple instance data representation of one positive bag and 2 negative bags with 3 features.

We propose a novel Quadratic Programming-based Multiple Instance Learning (QP-MIL) framework. Our proposal is based on the idea of determining a simple linear function for discriminating positive and negative bag classes. We model MIL problem as a QP problem using the input data representation. An optimal solution of our QP formulation returns an instance-level scoring function. For an unlabeled bag, instance-level scores are averaged to assess the bag-level score. Finally, class label of the bag is determined according to the predetermined threshold value. Rather than selecting bag representatives as in standard MIL, QP-MIL regards collective MIL assumption because of its modeling capability of the standard assumption and coverage on other MIL assumptions by means of the smooth average of instance-level decisions [13].

The remainder of the paper is organized as follows: Sect. 2 summarizes the existing MIL methods and mathematical programming formulations of MIL. Sect. 3 introduces formal description of the MIL problem and provides an existing SVM-

based MIL formulation, MIHLSVM as a background. Sect. 4 describes the proposed QP-MIL framework. Sect. 5 provides insights resulting from the numerical comparisons of QP-MIL with MIHLSVM and presents the classification success and computational efficacy of QP-MIL with the experiments on a wide-range of MIL datasets. Conclusions and future extensions are discussed in Sect. 6.

2 Related Work

Previously, various data-mining and machine learning algorithms have been devised to solve the MIL problem. These approaches are heuristic algorithms and optimality of their solutions cannot be guaranteed. In this study, we focus on optimization-based approaches to solve MIL problem, and we refer the reader to comprehensive surveys [1,5] for other categories of MIL methods.

SVM classification is extended to MIL setting previously [2,20,22,23,25,37]. Table 1 describes and compares the Multiple Instance Support Vector Machine (MISVM) models in the literature. The level of the formulations indicates whether the misclassification penalties are incurred for bags or not. The assumptions are qualified as weak if only the standard MIL assumption holds. Otherwise, if there are additional restrictions reflected to the mathematical model, assumption status is entitled as strong.

In MISVM models, an instance is selected from a positive bag as a witness to represent that bag. Figure 3 illustrates standard SVM classification in instance space and bag-level separation. To classify bags, a witness instance is selected from a positive bag as shown in Figure 3. Witness instances are considered to be responsible from bag positivity and must be correctly classified.

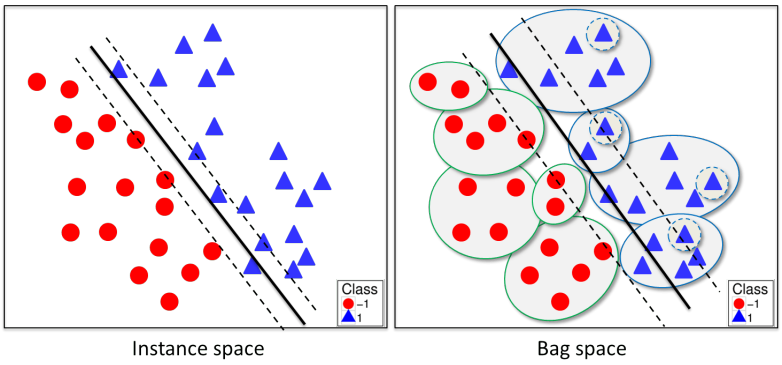

Figure 3. An illustration of witness selection in MISVM models. Red circles indicate instances in negative bags. In positive bags, instances are represented with blue triangles and witness instances are enclosed in dashed circles.

In mi-SVM and MI-SVM formulations [2], two types of constraints are added to the SVM formulation satisfying at least one sample in each positive bag has a label of one in mi-SVM and a witness instance is present for positive bags in MI-SVM. MissSVM [37] is formulated upon MI-SVM [2] with additional constraints

on the positive bags. Minimizing the misclassification error at either extreme, an instance of a positive bag is either positively or negatively labeled. Another method KI-SVM [22] selects witnesses from positive bags as key instances.

Sparse transductive MIL formulation (stMIL) [4] has an additional constraint that pulls all the negative instances in the bag closer to the hyperplane. An ℓ_1-norm SVM-based formulation [23] incorporating the assumption "arbitrary convex combination of instances in the positive bags represents each positive bag" is a linear program with bilinear constraints.

MIL problem is formulated as a mixed $0-1$ quadratic programming problem in [20], where MIL is reduced to instance-level learning, disregarding the bag information. Hard margin and soft margin maximization formulations of MIL, MIHMSVM and MIHLSVM [25] have additional bag-level misclassification penalties. A penalty is incurred if all instances in a positive bag are misclassified or at least one instance in the negative bag is misclassified. The resulting formulations are mixed integer quadratic programs (MIQPs), which are known to be NP-hard problems [20].

Most of the aforementioned MISVM models are analyzed in a recent survey [9]. It is emphasized in [9] that local convergence of the heuristic solution approaches for solving non-convex MISVM formulations leads to a sacrifice from the classification performance. The authors also discuss scalability of MISVM methods: Increased number of instances and bags affect model dimensionality and therefore increase both hyperparameter selection and model solution times.

When SVMs are tailored for MIL, specifically devised SVM solvers [11] can only be used solving subproblems of various heuristic solution algorithms [2, 20, 22, 23, 26, 37]. We propose a simplified QP formulation, which can be directly solved to optimality using any commercial QP solver. Instead of utilizing an iterative heuristic procedure, we are able to report exact solutions of each problem instance. Thus, repetition of the performed classification task is possible and the resulting classifier is reproducible in this way.

Table 1. The comparison of MIL formulations.

Formulation	Model type	Level	Assumptions	Solution approach	Main reference
mi-SVM	MIQP	instance	weak	mi-SVM opt. heuristic	[2]
MI-SVM	MIQP	bag	weak	MI-SVM opt. heuristic	[2]
stMIL	NC-MINLP	instance	strong	CCCP	[4]
MissSVM	NLP	instance	strong	CCCP	[37]
ℓ_1-norm SVM-MIL	LP/NLP	instance	strong	MICA	[23]
KI-SVM	MIQP	instance	strong	Cutting plane algorithm	[22]
Max-Margin MIL	0-1 MIQP	instance	weak	Branch and bound	[20]
MIHMSVM	MIQP	instance	strong	Three-phase heuristic	[25]
MIHLSVM	MIQP	bag	strong	Exact	[25]

Our study explores the utility of QP-MIL compared to the previous state-of-the-art MIL approaches. Leading methods in MIL literature are various machine learning-based approaches. We select several MIL algorithms as baseline methods to demonstrate success of the MIL classifiers. We carry out another comparison of QP-MIL considering SVM-based MIL, in terms of model building and classifier

testing. We experimented direct solution of a mixed integer quadratic programming (MIQP) formulation proposed in [25] for comparison.

3 Background

3.1 Problem Statement

Let $\mathbf{x}_i$ be a d-dimensional feature vector of instance i and $X = \{\mathbf{x}_i : i = 1, .., n\}$ be a set of instances. Also let y_i be a single, discrete-valued feature, specifically the label of instance i. Then, instance set $X = \{\mathbf{x}_i : i = 1, .., n\}$ forms the training set. This set can be labeled with y_i, $i = 1, ..., n$ or can be unlabeled. A bag B_j consists of a set of instances I_j formed by $\mathbf{x}_i$'s and n_j is the number of the instances in B_j. Therefore, $\chi = \{(B_j, l_j) : j = 1, .., m\}$ is a training bag set containing instances and a label l_j of each bag. Let an instance-based classifier be a function from instances to labels $f(\mathbf{x}_i) \rightarrow y_i$, and let $g(B_j) \rightarrow l_j$ be the function of a bag-based single classifier. Concisely, given a training set of bags with given label information $\chi = \{(B_j, l_j) : j = 1, .., m\}$, our MIL task is to learn a classifier $g(B_j)$ to predict the labels of input bags.

The sets, parameters and decision variables used in models are given as follows.
Indices:
$i = 1, 2, \ldots, n$: index for the instances
$j = 1, 2, \ldots, m$: index for the bags
Sets:
I_j: set of instances in bag
$J^+ = \{j : l_j = 1\}$: set of positive bags
$J^- = \{j : l_j = -1\}$: set of negative bags
$I^+ = \{i : i \in I_j \wedge j \in J^+\}$: set of instances in positive bags
$I^- = \{i : i \in I_j \wedge j \in J^-\}$: set of instances in negative bags
$I = I^+ \cup I^-$: set of all instances
Parameters:
$\mathbf{x}_i \in \Re^d$, $i = 1, 2, ..., n$: instance vectors
l_j: bag labels
C: trade-off parameter
Decision variables of QP-MIL:
$\mathbf{w}$: d-dimensional feature weight vector
m_i, $i = 1, 2, \ldots, n$: instance pseudo class memberships
β_j, $j = 1, 2, \ldots, m$: bag class memberships
δ_j^+, δ_j^-: slack variables for the positive and negative bag deviations
τ: decision threshold for bag classification
Decision variables of MIHLSVM [25]:
$\mathbf{w}$: d-dimensional feature weight vector
b: bias term
β_j, $j = 1, 2, \ldots, m$: bag class memberships
η_i, $i = 1, 2, \ldots, n$: variables identifying witness instances
z_i, $i = 1, 2, \ldots, n$: auxiliary variables replacing $\xi_i \eta_i$, $i = 1, 2, \ldots, n$.

3.2 A previous MIQP formulation: MIHLSVM [25]

Multiple Instance Hinge Loss Support Vector Machines (MIHLSVM) [25] extends traditional SVM for MIL. Unlike earlier SVM-based approaches to MIL, MIHLSVM defines bag-level hinge loss to penalize bag misclassifications. The proposed model handles the situation of nonlinearly seperable classes and the resulting formulation is a MIQP. The authors propose direct solution of MIHLSVM in [25] and do not present a heuristic algorithm as those in other MISVM studies [2,20, 22,23,37]. Still, it is difficult to get an exact solution to a MIHLSVM problem instance. We present our comparisons with MIHLSVM in Sect. 5.4.1.

A MIQP formulation of the described problem [25] is given as below

$$
\begin{aligned}
\text{(MIHLSVM)} \quad \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{\eta},\boldsymbol{z}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+\right) && \text{(1a)}\\
\text{st} \quad & -\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geq 1-\xi_i \quad \forall i \in I^- && \text{(1b)}\\
& \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1-\xi_i \quad \forall i \in I^+ && \text{(1c)}\\
& \sum_{i\in I_j}\eta_i = 1 \quad \forall j \in J^+ && \text{(1d)}\\
& \xi_i \leq \xi_j^- \quad \forall j \in J^-, \forall i \in I_j && \text{(1e)}\\
& \xi_j^+ = \sum_{i\in I_j} z_i \quad \forall j \in J^+ && \text{(1f)}\\
& z_i \geq \xi_i - M(1-\eta_i) \quad \forall i \in I^+ && \text{(1g)}\\
& z_i \leq \xi_i \quad \forall i \in I^+ && \text{(1h)}\\
& z_i \leq M\eta_i \quad \forall i \in I^+ && \text{(1i)}\\
& z_i \geq 0 \quad \forall i \in I^+ && \text{(1j)}\\
& \xi_i \geq 0 \quad \forall i \in I && \text{(1k)}\\
& \eta_i \in \{0,1\} \quad \forall i \in I^+. && \text{(1l)}
\end{aligned}
$$

In addition to maximization of the margin between bag classes, the objective function (1a) also minimizes bag misclassifications where a selected constant C controls the trade-off between two objectives. Constraints (1b) and (1c) are margin constraints enabling penalization of misclassification using slack variables ξ_i for misclassified instances. The weight vector $\mathbf{w}$ and the offset parameter b defines the instance-level separating hyperplane. Constraint (1d) forces a positive bag to have a positive instance as a witness. Negative bag misclassifications are represented by constraint (1e) using slack variables $\xi_j^-, \forall j \in J^-$. It is assumed that a negative bag is misclassified if all of its instances are misclassified.

Constraints (1g)–(1i) with the auxiliary variables $z_i \geq 0, \forall i \in I^+$ determine misclassification of a witness instance in a positive bag. Constraint (1f) assesses the misclassification of a positive bag as misclassification of its selected witness instance. Constraint (1l) imposes binary restrictions on witness variables and non-negativity restrictions on slack variables are introduced by constraint (1k).

After solving MIQP formulation, the following classifier can be used for bag classification

$$\operatorname{sgn}\left(\max_{i\in I_j}\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right)\right), \qquad j \in J. \tag{2}$$

We know that the MIHLSVM formulation given in (1) is a mixed integer quadratic program, and therefore, can be solved directly by commercial MIQP solvers. The efficiency of this approach along with QP-MIL is compared in Sect. 5.4.1 to verify the modeling and solution quality of the proposed MIL framework.

4 Quadratic Programming for Multiple Instance Learning

A bag classification rule can be found by solving the following optimization model:

$$\text{(QP)} \quad \min_{\mathbf{w},\boldsymbol{\beta},\mathbf{m},\tau,\boldsymbol{\delta}^+,\boldsymbol{\delta}^-} \quad \frac{1}{2}||\mathbf{w}||^2 - C\left(\frac{1}{m^+}\sum_{j\in J^+}\delta_j^+ + \frac{1}{m^-}\sum_{j\in J^-}\delta_j^-\right) \tag{3a}$$

$$\text{s.t.} \quad \langle \mathbf{w}, \mathbf{x}_i\rangle = m_i \quad \forall i \in I \tag{3b}$$

$$\beta_j = \frac{1}{n_j}\sum_{i\in I_j} m_i \quad \forall j \in J \tag{3c}$$

$$\beta_j \geq \tau + \delta_j^+ \quad \forall j \in J^+ \tag{3d}$$

$$\beta_j \leq \tau - \delta_j^- - \varepsilon \quad \forall j \in J^- \tag{3e}$$

$$0 \leq m_i \leq 1 \quad \forall i \in I \tag{3f}$$

$$0 \leq \delta_j^+ \leq 1 \quad \forall j \in J^+ \tag{3g}$$

$$0 \leq \delta_j^- \leq 1 \quad \forall j \in J^- \tag{3h}$$

$$0 \leq \tau \leq 1 \tag{3i}$$

Regularization processes are introduced to supervised learning problems for recovering the important features and for satisfying model generalizability. The quadratic objective function (3a) performs maximization of bag class membership margin together with a regularization of feature weights. In the first term of the objective function (3a), standard ℓ_2-norm of the weight coefficients $\mathbf{w}$ are minimized. Therefore, effect of redundant or uninformative features can also be controlled. The second term of the objective function (3a) maximizes the margin of bag class estimates formed by the threshold variable τ. In order to handle potential problems due to class imbalances, summations of the nonzero slack variables δ_j^+, $\forall j \in J^+$ and δ_j^-, $\forall j \in J^-$ in the objective function (3a) are normalized with the number of positive bags m^+, and the number of negative bags m^-, respectively. The hyperparameter C in the objective function (3a) tunes the trade-off between regularization of $\mathbf{w}$ and maximization of bag class membership estimate margin.

For each instance, an estimate of the class label is obtained as a pseudo class membership value. Constraint (3b) determines instance pseudo class memberships $m_i, \forall i = 1, \ldots, n$ using the coefficient vector $\mathbf{w}$ entry of which corresponds to the weight assigned to a feature of the input data. For each instance, Constraint (3c) maps bag-level class estimates $\beta_j, \forall j = 1, \ldots, m$ onto the $[0, 1]$ interval by averaging instance-level scores, which are forced to be between 0 and 1 by Constraint (3f).

Constraints (3d) and (3e) ensure that absolute difference between class membership estimate β_j and the threshold τ are maximized in the objective function for both positive and negative bags. Constraint (3i) restricts the decision threshold τ to be between 0 and 1. Similarly, slack variables δ_j^+, $\forall j \in J^+$ and δ_j^-, $\forall j \in J^-$ are restricted to be between 0 and 1 by Constraints (3g) and (3h). We set ε in Equation (3e) to a small positive value (10^{-6}) so that class membership value of a negative bag is strictly below the threshold τ.

QP-MIL models the contributions of all instances in a bag to the bag label collectively. Averages of pseudo-class membership estimates for instances determine the class membership estimates for the bags. A bag is positively labeled if its class membership value is above decision threshold τ, and negatively labeled otherwise. An optimal value of τ is adaptively identified in QP-MIL during the optimization process. This threshold is also applicable to the test bags. After solving the QP formulation in (3) on the training set, instance scores are calculated by Equation (3b) for each instance in a test bag and simply averaged in Equation (3c) to compute the bag-level score. If the output is below the optimal value of τ, the classifier produces a negative label, else a positive label.

The resulting bag-level classifier can be defined as

$$g(B_j) = \begin{cases} 1 & \text{if} \quad \beta_j \geq \tau, \\ -1 & \text{otherwise,} \end{cases}$$

where

$$\beta_j = \tfrac{1}{n_j} \textstyle\sum_{i \in I_j} m_i,$$

and

$$m_i = \langle \mathbf{w}, \mathbf{x}_i \rangle \quad \forall i \in I_j.$$

Our proposed MIL framework is independent of the underlying MIL assumptions. We seek to model bag structures by taking into account the reflection of instance scores to the bag labels. Since all instances contribute to the bag-level scoring, this paradigm resembles the collective MIL assumption [12]. It is shown in [13] that if an instance level separation can be performed in an embedding space $\mathcal{H}$ with a classifier f in a standard MIL problem, then the bags can also be separated in another embedding space $\mathcal{H}'$, which has a higher dimensionality than $\mathcal{H}$, by scoring each bag with the average of its instance-level estimates as $g(B_j) = \frac{1}{n_j} \sum_{i \in I_j} f(\mathbf{x}_i)$. Therefore, various MIL assumptions can be handled with a proper data representation and collective modeling of the bag structures.

In order to perform class separation by correct classification, having class membership values above the threshold for positive bags and below the threshold for negative bags is desirable. Therefore, we maximize summation of the absolute differences between bag class membership estimates. This paradigm defines the margin between positive and negative class membership estimates, as well. Thus, optimal value of decision threshold τ leaves the maximum margin between bag class membership estimates. Figure 4 illustrates a possible solution to the QP model (3). The selected value for decision threshold τ is 0.55 and the class memberships estimates for 3 positive and 3 negative bags are consistent with this threshold.

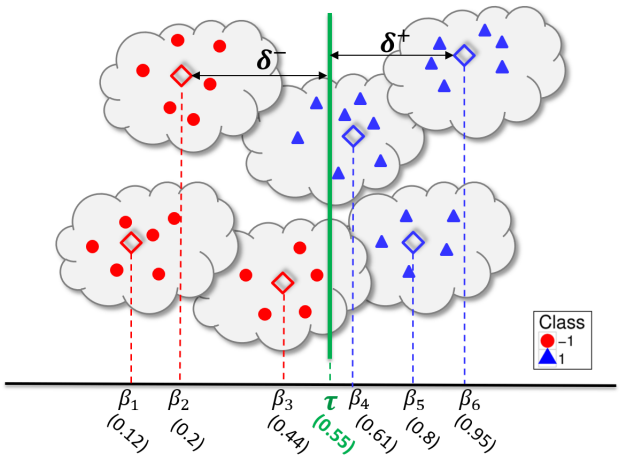

Figure 4. An illustration of a solution to QP model (3). Instance level scores are symbolized with red circles and blue triangles, for negative and positive bags, respectively. The vertical green line indicates the decision threshold and each dashed line maps bag class membership of a bag. For a positive and a negative bag, class membership margins are indicated with horizontal arrows.

5 Experiments

5.1 Data representation

In MIL, a specific data region representing the positive instance class is named as a *concept*. The concept instances are informative for class discrimination. Based on this idea, representative sets can be derived in many ways as prototypes to capture the informative instance relationships. Several MIL methods benefit from the dissimilarities to selected prototypes to represent the bags [6,7,10,14,21]. Moreover, a number of similar algorithms [31,38] utilize clustering to learn a target concept in MIL problems. Inspired by success of aforementioned methods, we attempt to perform MIL classification in a newly represented feature space. QP model (3) produces a linear classifier and success of this classifier is limited only to linearly separable data.

In QP-MIL, the relationships between instances can be implicitly modeled by preprocessing the input data. Instead of building a classifier in the original instance feature space, we attempt to represent the instances using dissimilarities to the selected prototypes. Aim of the representation is building a linear classifier, which is capable of class separation in a different space. We pool instances in bags and then group them by k-means clustering algorithm into an appropriate number of clusters. Then, the cluster centers are taken as the prototypes. The new features are simply constructed by calculating the Euclidean distances of each instance to these cluster centers. This way, protoypes are derived as a summarized representation of the original data and the linear classifier becomes applicable to the new features.

5.2 Multiple instance datasets

We evaluate our approach in image classification, molecular activity prediction, text categorization and audio classification tasks. The datasets are categorized in Table 2 based on their application domain and the dataset characteristics are also provided. The first category includes famous drug activity prediction tasks on Musks and Mutagenesis' datasets and a protein identification task. Image classification datasets constitute the second category containing the Corel image datasets, UCSB breast cancer dataset and other smaller sized benchmarks Elephant, Fox and Tiger. Positive class is considered as the target images and the remaining images determine the negative bag class.

Another dataset category covers web mining tasks on Newsgroups and Web recommendation datasets. In Newsgroups, blog posts are categorized into 20 groups based on their subjects where a bag is formed by a collection of multiple posts (i.e. the instances). In the positive class, the terms about a specific subject appears in a number of posts, and the bags with posts about other subjects constitute the negative bags. In Web recommendation, a web page in the user history is a bag and the web pages linked to that web page are the instances. Recommendations of a specific user form the positive class and the bags constituted by the remaining eight users are negatively labeled.

The last category is the bird song recordings from 13 different classes of birds, where a recording is bag and segments of recording are the instances. The target bird class is considered as positive, whereas the bags from the other classes are labeled as negative. We follow an effective experimentation strategy. Cross validation folds are generated by splitting the original dataset into the training set and the test set. We utilize the same splitting indices across both our proposed and the state-of-the-art methods from the literature to perform a comprehensive comparison. All the datasets and cross validation indices are available online at [19].

5.3 Experimental setting and performance criteria

Our experiments use a Windows 10 PC with 16 GB RAM, dual core CPU (Intel Core i7-7700HQ 2.8 GHz). For each dataset, a stratified cross validation scheme is conducted to assess the generalizability of the classifiers. Initially, we scale each feature to zero mean and unit variance. We obtain data representations in QP-MIL via the implementation in Python that uses scikit-learn [24] library. We model QP formulations using Gurobi Python interface and solve using barrier QP solver of Gurobi 8.0 [15]. The default parameters are accepted for the barrier algorithm except for the convergence tolerance, which is set to 0.01. QP-MIL has two parameters: number of clusters, κ in data representation and cost parameter C of QP model (3). In k-means clustering, necessarily enough number of clusters, κ is determined by using elbow approach [18]. Briefly, within cluster variance after k-means clustering is plotted along with increasing values of κ and the position of the elbow is identified to assign the corresponding value to κ. We run a nested cross-validation with an inner cross-validation loop to choose hyperparameter C from the set $\{0.01, 0.1, 1, 10, 100, 1000\}$. All of the instances of MIHLSVM formulation are also executed using Gurobi 8.0 [15].

Table 2. Common MIL datasets

Name	Instances	Min	Max	Features	Bags	+ bags	- bags
Musk 1 [8] ♣	476	2	40	166	92	47	45
Musk 2 [8] ♣	6598	1	1044	166	102	39	63
Mutagenesis 1 (easy) [27] ♣	10486	28	88	7	188	125	63
Mutagenesis 2 (hard) [27] ♣	2132	26	86	7	42	13	29
Protein [28] ♣	26611	35	189	8	193	25	168
Elephant [2] ♥	1391	2	13	230	200	100	100
Fox [2] ♥	1302	1	13	230	200	100	100
Tiger [2] ♥	1220	2	13	230	200	100	100
Corel, African [6] ♥	7947	2	13	9	2000	100	1900
Corel, Antique [6] ♥	7947	2	13	9	2000	100	1900
Corel, Battleships [6] ♥	7947	2	13	9	2000	100	1900
Corel, Beach [6] ♥	7947	2	13	9	2000	100	1900
Corel, Buses [6] ♥	7947	2	13	9	2000	100	1900
Corel, Cars [6] ♥	7947	2	13	9	2000	100	1900
Corel, Desserts [6] ♥	7947	2	13	9	2000	100	1900
Corel, Dinosaurs [6] ♥	7947	2	13	9	2000	100	1900
Corel, Dogs [6] ♥	7947	2	13	9	2000	100	1900
Corel, Elephants [6] ♥	7947	2	13	9	2000	100	1900
Corel, Fashion [6] ♥	7947	2	13	9	2000	100	1900
Corel, Flowers [6] ♥	7947	2	13	9	2000	100	1900
Corel, Food [6] ♥	7947	2	13	9	2000	100	1900
Corel, Historical [6] ♥	7947	2	13	9	2000	100	1900
Corel, Horses [6] ♥	7947	2	13	9	2000	100	1900
Corel, Lizards [6] ♥	7947	2	13	9	2000	100	1900
Corel, Mountains [6] ♥	7947	2	13	9	2000	100	1900
Corel, Skiing [6] ♥	7947	2	13	9	2000	100	1900
Corel, Sunset [6] ♥	7947	2	13	9	2000	100	1900
Corel, Waterfalls [6] ♥	7947	2	13	9	2000	100	1900
UCSB Breast Cancer [17] ♥	2002	21	40	708	58	26	32
Newsgroups 1, alt.atheism [36] ♠	5443	22	76	200	100	50	50
N.g. 2, comp.graphics [36] ♠	3094	12	58	200	100	50	50
N.g. 3, comp.os.ms-windows.misc [36] ♠	5175	25	82	200	100	50	50
N.g. 4, comp.sys.ibm.pc.hardware [36] ♠	4827	19	74	200	100	50	50
N.g. 5, comp.sys.mac.hardware [36] ♠	4473	17	71	200	100	50	50
N.g. 6, comp.windows.x [36] ♠	3110	12	54	200	100	50	50
N.g. 7, misc.forsale [36] ♠	5306	29	84	200	100	50	50
N.g. 8, rec.autos [36] ♠	3458	15	39	200	100	50	50
N.g. 9, rec.motorcycles [36] ♠	4730	22	73	200	100	50	50
N.g. 10, rec.sport.baseball [36] ♠	3358	15	58	200	100	50	50
N.g. 11, rec.sport.hockey [36] ♠	1982	8	38	200	100	50	50
N.g. 12, sci.crypt [36] ♠	4284	20	71	200	100	50	50
N.g. 13, sci.electronics [36] ♠	3192	12	58	200	100	50	50
N.g. 14, sci.med [36] ♠	3045	11	54	200	100	50	50
N.g. 15, sci.space [36] ♠	3655	16	59	200	100	50	50
N.g. 16, soc.religion.christian [36] ♠	4677	21	71	200	100	50	50
N.g. 17, talk.politics.guns [36] ♠	3558	13	59	200	100	50	50
N.g. 18, talk.politics.mideast [36] ♠	3376	15	55	200	100	50	50
N.g. 19, talk.politics.misc [36] ♠	4788	21	75	200	100	50	50
N.g. 20, talk.religion.misc [36] ♠	4606	25	79	200	100	50	50
Web recommendation 1 [35] ♠	2212	4	131	5863	75	17	58
Web recommendation 2 [35] ♠	2212	5	200	6519	75	18	57
Web recommendation 3 [35] ♠	2212	5	200	6306	75	14	61
Web recommendation 4 [35] ♠	2291	4	200	6059	75	55	20
Web recommendation 5 [35]	2546	5	200	6407	75	61	14
Web recommendation 6 [35] ♠	2462	4	200	6417	75	59	16
Web recommendation 7 [35] ♠	2400	4	200	6450	75	39	36
Web recommendation 8 [35] ♠	2183	4	200	5999	75	35	40
Web recommendation 9 [35] ♠	2321	5	200	6279	75	37	38
Birds, Brown creeper [3] ♦	10232	2	43	38	548	197	351
Birds, Chestnut-backed chickadee [3] ♦	10232	2	43	38	548	117	431
Birds, Dark-eyed junco [3] ♦	10232	2	43	38	548	20	528
Birds, Hammonds flycatcher [3] ♦	10232	2	43	38	548	103	445
Birds, Hermit thrush [3] ♦	10232	2	43	38	548	15	533

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

Table 2 continued.

Name	Instances	Min	Max	Features	Bags	+ bags	- bags
Birds, Hermit warbler [3] ♦	10232	2	43	38	548	63	485
Birds, Olive-sided flycatcher [3] ♦	10232	2	43	38	548	90	458
Birds, Pacific slope flycatcher [3] ♦	10232	2	43	38	548	165	383
Birds, Red-breasted nuthatch [3] ♦	10232	2	43	38	548	82	466
Birds, Swainsons thrush [3] ♦	10232	2	43	38	548	79	469
Birds, Varied thrush [3] ♦	10232	2	43	38	548	89	459
Birds, Western tanager [3] ♦	10232	2	43	38	548	46	502
Birds, Winter Wren [3] ♦	10232	2	43	38	548	109	439

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

The baseline MIL approaches selected for comparison are MILES [6], MInD [7] with bag dissimilarity representation $D_{meanmin}$ and miFV [33]. MILES iteratively measures similarities of bags to the training instances, and builds a linear SVM classifier along with ℓ_1-norm regularization at the same time. MInD defines a bag-level feature representation by using the bag-to-bag dissimilarity measure $D_{meanmin}$. miFV benefits from Fisher vectorial coding to map each bag to a single vector. Both MInD and miFV build a linear SVM classifier to classify bag vectors. We execute MILES [6] and MInD [7] using the MIL toolbox [29], and use a MATLAB [32] implementation to run miFV [33]. We accept the default parameters in the original paper for MILES [6]. We use the parameter setting proposed in [7] for MInD [7]. Following the authors' advice, we employ an inner ten-fold cross-validation to select the three parameters of miFV [33], which are enumerated as PCA energy, number of components and cost parameter of linear SVM. PCA energy attains values from the set $\{0.8, 0.9, 1\}$. The alternatives for the number of Gaussian components is selected from $\{1, 2, 3, 4, 5\}$. The cost parameter of the linear SVM classifier are $\{0.05, 1, 10\}$.

A receiver operation characteristics (ROC) curve visualizes the trade-off between percentage of true positive predictions and percentage of false positive predictions. Area under the ROC curve (AUC) is asserted to be a reliable metric for classification [16]. Larger AUC values indicate a better classifier. Another measure for classifier performance in MIL problems is classification accuracy. For a specific decision threshold value, such as the value of τ in QP-MIL after optimization, the bag classes are predicted and the accuracy of the classifier is computed. The class imbalance problem is seen in MIL tasks such as Corel, Web recommendation and Birds benchmarks. The value of τ is optimized on the training bags, and suffers from misleading accuracy when the bag classes are imbalanced. AUC is more effective under class imbalance since all possible thresholds are evaluated to report the classifier performance. Additionally, given the consistent performance of AUC on MIL datasets [30], we qualify AUC as a primary comparison metric in our study.

5.4 Experimental results

5.4.1 Comparison of QP-MIL with MIHLSVM

In this section, we present a comparison between QP-MIL and MIHLSVM formulation given in Sect. 1 in terms of computational efficiency and other indicators related to classification performance of the derived solutions. The clustering-based

	QP-MIL				MIHLSVM			
# / Dataset	Constraints	Cont. variables	Binary variables	Quad. terms	Constraints	Cont. variables	Binary variables	Quad. terms
Elephant	1611.9	1881.4	0	267.5	4055.4	2952.3	1251.9	267.5
Fox	1548.0	1827.5	0	277.5	3720.6	2834.5	1188.0	277.5
Musk 1	594.0	816.0	0	220.0	1314.0	1160.6	428.4	220.0
Musk 2	6121.8	6381.3	0	257.5	13777.2	12226.7	5938.2	257.5

Table 3. Model size summary of QP-MIL and MIHLSVM on problem instances of 4 datasets.

data representation described in Sect. 5.1 is considered as the input of all compared formulations.

Table 3 presents the overview of problem sizes on four moderate sized MIL datasets. All datasets are modelled using QP-MIL formulation in (3) and the MIHLSVM formulation in (1). For each dataset in Table 3, ten separate models of QP-MIL and MIHLSVM are built, where ten different partitioning of the original dataset form the input in each model. The averages of problem dimension properties for ten models are reported in Table 3. Formulations in (3) and (1) have quadratic objective functions and number of the quadratic terms are equal for both. Since we solve the formulations on a cluster center-based data representation, the number of quadratic terms is equal to the dimensionality of this representation.

In Table 4, we compare the performance of QP-MIL with the MIHLSVM. MIHLSVM is an MIQP and can be directly solved by standard MIQP solvers. We solve the MIHLSVM formulation in (1) and set the cost parameter C in the objective function (1a) to 1. It is plausible to tune up the appropriate value for C by a cross-validation procedure. However, the computation time of parameter selection in MIHLSVM is a limitation [25].

We are unable to report overall results for MIHLSVM since each cross-validation fold lasts longer than one day for relatively small datasets such as Elephant and Fox. Therefore, we do not carry out a cross-validation loop, and manifest only the model solution time for $C = 1$. In contrast with the described procedure in Sect. 4, we do not embed parameter selection into QP-MIL during comparisons of this section and the predetermined value of C is 1. The results in Table 4 are based on one repeat of a ten-fold cross validation. All methods are executed within a time limit of 1800 seconds. First column is the number of problem instance from each dataset that is solved to optimality until the time limit is reached. The mean percentage optimality gap [(upper bound- lower bound)/upper bound] is reported for each algorithm and the corresponding average model solution time in seconds is also presented. To observe generalizability of the learner, we evaluate obtained solutions on the test bags. Average accuracy and AUC values over ten experiments are reported for all three approaches.

Computational study demonstrates that QP-MIL is significantly more efficient and provides accurate solutions compared to the MIHLSVM formulation. All instances of QP-MIL can be solved exactly without a sacrifice in classification success as demonstrated by AUC and accuracy results in Table 4. Being the largest dataset in this comparison, Musk 2 requires an average solution time of 3 seconds to solve QP model (3) to optimality. On the other hand, only one MIHLSVM instance of Musk 1 dataset can be solved to optimality within the time limit. Ex-

	QP-MIL				
Dataset	Solved	Gap	Time	AUC	Accuracy
Elephant	10	0	1.6	93.7	83.5
Fox	10	0	1.5	70.3	65.0
Musk 1	10	0	0.3	96.5	89.0
Musk 2	10	0	3.0	94.7	88.3
	MIHLSVM				
Dataset	Solved	Gap	Time	AUC	Accuracy
Elephant	0	97.6	1800	87.3	63.5
Fox	0	98.6	1800	64.9	55.0
Musk 1	1	37.1	1721.4	89.3	71.7
Musk 2	0	91.0	1800	90.8	74.4

Table 4. Comparison of QP-MIL and MIHLSVM on problem instances of 4 datasets. 10 models of each formulation are built for each dataset, and the average values are reported.

cept for Musk 1, Gurobi is unable to reduce the optimality gap below 90%. For the sake of fairness, we do not include MIHLSVM in the overall comparison results in Sect. 5.4.2 due to the requirements of a higher runtime even for small/moderate sized datasets.

5.4.2 Comparison to baseline methods

Table 5 summarizes the performance of our proposed QP-MIL approach with MILES [6], MInD [7] with bag dissimilarity representation D_{meanmin} and miFV [33] on four different MIL application categories. Their descriptions and implementation details are provided in Sect. 5.3.

AUC and accuracy results of MIL classifiers in Table 5 are the averages of a ten-fold cross validation repeated for five times. The best result for each dataset is in boldface. In molecular activity prediction, the highest AUC results are obtained by QP-MIL in Musk 1, and by D_{meanmin} in Musk 2. Fisher vector based bag representation suits on Mutagenesis 1 dataset, where second best AUC and accuracy results are obtained by QP-MIL and miFV, respectively. In Protein, the leading method is MILES, which is followed by QP-MIL.

QP-MIL has the best image classification success in Elephant, Tiger and USCB Breast cancer datasets. The implicit instance selection mechanism of MILES is effective on Fox dataset and QP-MIL follows MILES on this dataset. In Corel image datasets, D_{meanmin} has the highest average performance, and QP-MIL performs very close to D_{meanmin}. Results of QP-MIL and D_{meanmin} are very close to each other on the average on 20 Newsgroups datasets. In Web recommendation, performance of QP-MIL falls behind miFV and D_{meanmin}. QP-MIL has the highest AUC and accuracy results in almost all Birds datasets.

The average testing results based on problem categories are reported in Table 6. For each problem category, results of the best method are in boldface, whereas the second best results are shown in italic. Average AUC and accuracy results in Table 6 demonstrate that QP-MIL is competitive with other algorithms across all application categories and provides the best classification results on some datasets. QP-MIL achieves the best or the second best average AUC and accuracy performance on molecular activity prediction datasets.

Table 5. AUC and accuracy results of four MIL methods with standard errors (× 100).

Dataset	AUC				Accuracy			
	MILES	$D_{meanmin}$	miFV	QP-MIL	MILES	$D_{meanmin}$	miFV	QP-MIL
Musk 1 ♣	93.5 (1.0)	94.5 (1.2)	94.1 (1.2)	**96.8 (0.8)**	79.6 (2.0)	84.1 (1.8)	85.2 (1.6)	**88.4 (1.4)**
Musk 2 ♣	96.7 (0.7)	**97.6 (0.8)**	94.7 (1.2)	94.5 (1.0)	86.3 (1.1)	**92.3 (1.1)**	87.7 (1.6)	87.8 (1.5)
Mutagenesis 1 ♣	78.0 (1.5)	85.1 (1.2)	**88.7 (1.2)**	85.5 (1.5)	79.8 (1.2)	77.4 (1.3)	**82.6 (1.2)**	77.6 (1.5)
Mutagenesis 2 ♣	62.7 (5.0)	64.7 (5.3)	68.3 (5.0)	**78.5 (3.8)**	70.9 (2.7)	70.4 (2.0)	**76.6 (2.2)**	71.2 (3.0)
Protein ♣	**95.3 (1.1)**	52.3 (3.7)	80.0 (1.9)	85.1 (1.6)	**94.9 (0.6)**	87.1 (0.3)	85.4 (0.8)	88.3 (0.9)
Elephant ♥	92.7 (0.7)	93.6 (0.9)	91.4 (0.9)	**94.1 (0.7)**	82.7 (1.0)	**86.2 (1.0)**	82.9 (1.1)	85.0 (0.9)
Fox ♥	**73.8 (1.6)**	61.2 (1.7)	67.5 (1.5)	67.7 (1.4)	**64.9 (1.4)**	57.8 (1.2)	62.3 (1.3)	63.4 (1.2)
Tiger ♥	86.8 (1.0)	85.3 (1.1)	87.5 (1.1)	**90.1 (1.0)**	79.2 (1.2)	77.7 (1.2)	80.4 (1.3)	**81.8 (1.2)**
Corel, African ♥	95.7 (0.5)	**96.7 (0.4)**	94.4 (0.6)	95.3 (0.5)	96.6 (0.2)	**97.3 (0.1)**	96.4 (0.1)	95.8 (0.2)
Corel, Antique ♥	87.3 (0.7)	**92.2 (0.6)**	90.8 (0.6)	87.1 (0.7)	93.8 (0.2)	**95.4 (0.1)**	95.0 (0.1)	92.7 (0.3)
Corel, Battleships ♥	94.9 (0.5)	**98.1 (0.2)**	92.9 (0.6)	95.8 (0.3)	96.1 (0.2)	**97.1 (0.1)**	96.2 (0.1)	96.0 (0.2)
Corel, Beach ♥	**99.3 (0.1)**	98.3 (0.4)	97.4 (0.4)	**99.3 (0.1)**	98.1 (0.1)	97.8 (0.1)	97.7 (0.1)	**98.4 (0.1)**
Corel, Buses ♥	**97.5 (0.4)**	97.3 (0.4)	94.0 (0.7)	96.0 (0.4)	**98.1 (0.1)**	97.7 (0.1)	97.2 (0.2)	96.8 (0.2)
Corel, Cars ♥	91.9 (0.7)	**94.8 (0.5)**	91.7 (0.7)	91.2 (0.7)	95.4 (0.2)	**97.3 (0.1)**	96.5 (0.1)	95.2 (0.2)
Corel, Desserts ♥	93.9 (0.7)	**97.4 (0.3)**	97.3 (0.4)	96.9 (0.3)	96.5 (0.1)	**97.7 (0.1)**	97.4 (0.1)	96.9 (0.2)
Corel, Dinosaurs ♥	97.5 (0.3)	**98.3 (0.2)**	94.4 (0.5)	96.4 (0.4)	97.5 (0.1)	**97.9 (0.1)**	96.6 (0.1)	96.3 (0.2)
Corel, Dogs ♥	87.4 (1.1)	**91.9 (0.7)**	86.4 (1.2)	86.3 (0.9)	94.8 (0.2)	**96.3 (0.1)**	95.6 (0.1)	93.4 (0.3)
Corel, Elephants ♥	95.7 (0.4)	**98.2 (0.2)**	95.7 (0.4)	96.8 (0.3)	96.0 (0.1)	**96.9 (0.1)**	96.5 (0.1)	96.2 (0.1)
Corel, Fashion ♥	99.0 (0.1)	**99.0 (0.1)**	98.9 (0.2)	98.6 (0.2)	98.3 (0.1)	**98.6 (0.1)**	98.1 (0.1)	98.1 (0.1)
Corel, Flowers ♥	94.3 (0.6)	**94.7 (0.6)**	93.8 (0.6)	93.8 (0.5)	96.1 (0.2)	**97.0 (0.1)**	96.4 (0.2)	95.5 (0.2)
Corel, Food ♥	99.4 (0.1)	**99.8 (0.1)**	98.7 (0.1)	99.2 (0.1)	98.6 (0.1)	**98.9 (0.1)**	97.9 (0.2)	97.9 (0.1)
Corel, Historical ♥	99.3 (0.1)	**99.8 (0.0)**	98.5 (0.3)	99.4 (0.1)	98.7 (0.1)	**98.9 (0.1)**	97.8 (0.1)	98.6 (0.1)
Corel, Horses ♥	89.6 (0.7)	**92.0 (0.6)**	88.9 (0.8)	86.0 (0.8)	94.6 (0.2)	**96.2 (0.1)**	95.9 (0.1)	93.1 (0.2)
Corel, Lizards ♥	97.0 (0.4)	**98.0 (0.3)**	95.8 (0.5)	97.3 (0.3)	**97.8 (0.1)**	**97.8 (0.1)**	97.0 (0.1)	97.2 (0.2)
Corel, Mountains ♥	99.9 (0.0)	**100 (0.0)**	99.9 (0.0)	99.8 (0.0)	99.3 (0.1)	**99.5 (0.1)**	**99.5 (0.1)**	99.1 (0.1)
Corel, Skiing ♥	94.7 (0.4)	**96.0 (0.3)**	95.9 (0.4)	**96.0 (0.3)**	**96.4 (0.1)**	**96.4 (0.1)**	96.3 (0.1)	96.0 (0.1)
Corel, Sunset ♥	76.3 (1.2)	**83.7 (1.0)**	77.1 (1.3)	73.9 (1.2)	92.4 (0.2)	**95.2 (0.1)**	95.1 (0.1)	90.5 (0.4)
Corel, Waterfalls ♥	94.5 (0.5)	**97.5 (0.2)**	93.4 (0.5)	95.5 (0.3)	96.0 (0.2)	**97.0 (0.2)**	95.8 (0.1)	95.5 (0.2)
UCSB Breast Cancer ♥	83.3 (2.6)	83.1 (2.7)	86.8 (2.5)	**88.8 (2.2)**	75.8 (2.2)	72.2 (2.3)	79.6 (2.4)	**81.7 (2.0)**
Newsgroups 1, alt.atheism ♠	41.6 (2.3)	**94.1 (1.0)**	91.1 (1.2)	93.3 (1.2)	41.8 (1.8)	**85.6 (1.5)**	81.2 (1.4)	82.0 (1.6)
N.g. 2, comp.graphics ♠	52.8 (2.2)	**89.8 (1.6)**	57.2 (3.2)	83.1 (1.8)	52.2 (2.0)	**79.0 (1.4)**	53.4 (1.2)	75.0 (1.8)
N.g. 3, comp.os.ms-windows.misc ♠	47.9 (2.7)	**81.0 (2.1)**	66.8 (2.2)	77.8 (2.1)	46.6 (2.2)	54.0 (0.9)	55.2 (1.7)	**71.2 (1.7)**
N.g. 4, comp.sys.ibm.pc.hardware ♠	68.2 (2.4)	**85.7 (2.2)**	69.5 (2.4)	82.0 (1.9)	62.4 (2.5)	**75.4 (1.6)**	65.0 (2.1)	74.4 (1.9)
N.g. 5, comp.sys.mac.hardware ♠	58.9 (2.8)	**85.2 (1.6)**	65.0 (2.6)	81.2 (1.6)	56.4 (2.4)	**79.6 (1.2)**	59.6 (1.6)	73.4 (1.6)
N.g. 6, comp.windows.x ♠	61.0 (2.4)	**89.0 (1.7)**	82.2 (2.0)	84.9 (2.2)	56.8 (2.0)	66.8 (1.5)	75.0 (1.9)	**76.4 (2.0)**

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

Table 5 continued.

Dataset	AUC				Accuracy			
	MILES	$D_{meanmin}$	miFV	QP-MIL	MILES	$D_{meanmin}$	miFV	QP-MIL
N.g. 7, misc.forsale ♠	51.1 (2.4)	79.0 (2.0)	72.6 (2.5)	**81.6 (1.8)**	53.0 (2.3)	53.2 (1.5)	60.0 (1.8)	**70.0 (2.1)**
N.g. 8, rec.autos ♠	49.8 (2.5)	**87.0 (1.7)**	72.7 (2.5)	81.8 (1.9)	50.2 (1.7)	**77.0 (1.6)**	63.0 (1.9)	69.0 (2.0)
N.g. 9, rec.motorcycles ♠	52.2 (2.6)	32.6 (3.2)	81.2 (2.4)	**85.0 (1.9)**	51.8 (1.9)	51.0 (0.5)	74.0 (2.1)	**74.8 (1.9)**
N.g. 10, rec.sport.baseball ♠	51.0 (2.7)	**91.4 (1.4)**	86.4 (1.8)	88.0 (1.8)	50.6 (1.9)	**80.0 (1.6)**	77.2 (1.5)	75.8 (1.7)
N.g. 11, rec.sport.hockey ♠	37.4 (2.2)	**95.8 (0.8)**	87.9 (1.5)	92.3 (1.4)	42.0 (2.1)	**85.8 (1.5)**	77.0 (1.5)	82.6 (1.8)
N.g. 12, sci.crypt ♠	46.2 (2.7)	84.0 (1.9)	85.1 (1.8)	**86.5 (2.0)**	47.2 (2.0)	62.4 (1.6)	**77.6 (2.1)**	75.6 (2.1)
N.g. 13, sci.electronics ♠	48.7 (2.2)	94.6 (1.0)	61.6 (2.6)	**94.9 (0.8)**	46.6 (1.7)	**89.0 (1.3)**	57.4 (1.6)	87.4 (1.2)
N.g. 14, sci.med ♠	49.8 (2.4)	**94.2 (0.8)**	84.3 (1.7)	88.8 (1.5)	52.0 (1.9)	**80.0 (1.4)**	73.2 (1.5)	78.6 (1.4)
N.g. 15, sci.space ♠	43.6 (2.5)	90.5 (1.4)	82.9 (1.9)	**92.1 (1.1)**	45.2 (2.1)	78.4 (1.5)	77.4 (1.9)	**83.6 (1.4)**
N.g. 16, soc.religion.christian ♠	46.1 (2.4)	**89.8 (1.4)**	84.9 (1.5)	82.2 (2.0)	45.0 (1.9)	**83.8 (1.4)**	76.0 (1.4)	70.2 (1.7)
N.g. 17, talk.politics.guns ♠	49.8 (2.5)	**87.4 (1.5)**	82.7 (2.0)	79.6 (1.9)	48.4 (2.1)	**77.8 (1.6)**	73.4 (1.7)	66.0 (1.9)
N.g. 18, talk.politics.mideast ♠	54.6 (3.0)	**87.4 (1.7)**	85.8 (1.9)	85.2 (1.5)	53.8 (2.1)	78.2 (1.3)	**79.0 (1.5)**	76.6 (1.7)
N.g. 19, talk.politics.misc ♠	55.0 (2.3)	80.2 (1.9)	67.2 (2.9)	**81.4 (2.0)**	51.8 (2.1)	68.6 (1.7)	60.0 (2.2)	**72.2 (1.8)**
N.g. 20, talk.religion.misc ♠	56.0 (2.7)	**83.4 (2.2)**	80.9 (2.3)	81.3 (2.3)	52.8 (2.3)	62.4 (1.3)	69.2 (2.2)	**70.8 (1.9)**
Web 1 ♠	73.2 (3.0)	63.4 (4.2)	**83.2 (2.3)**	65.5 (3.4)	**75.5 (1.5)**	69.9 (0.9)	74.9 (1.3)	68.1 (2.0)
Web 2 ♠	**54.4 (3.9)**	47.4 (4.2)	37.1 (2.5)	53.7 (4.4)	75.0 (1.1)	**75.3 (0.9)**	73.4 (1.2)	68.1 (2.3)
Web 3 ♠	67.1 (4.4)	70.8 (4.6)	**73.3 (3.6)**	66.1 (4.1)	**85.1 (1.4)**	81.6 (1.0)	82.1 (1.1)	74.9 (2.1)
Web 4 ♠	74.3 (3.5)	79.9 (3.6)	**81.2 (3.4)**	62.7 (3.3)	76.5 (1.8)	75.8 (0.9)	**82.5 (1.7)**	69.6 (1.6)
Web 5 ♠	**74.3 (3.4)**	71.1 (3.7)	68.7 (3.4)	55.2 (3.8)	**82.1 (1.3)**	79.7 (1.1)	79.8 (1.1)	78.0 (1.8)
Web 6 ♠	55.0 (3.4)	52.5 (4.2)	64.6 (3.6)	**65.0 (3.6)**	**81.5 (1.0)**	78.9 (0.8)	74.7 (1.3)	75.1 (1.7)
Web 7 ♠	62.5 (2.6)	69.0 (2.8)	**69.7 (3.4)**	54.5 (3.2)	54.3 (1.9)	63.1 (2.3)	**65.1 (2.8)**	51.3 (2.7)
Web 8 ♠	51.1 (3.1)	40.9 (2.6)	**53.7 (2.4)**	53.0 (3.0)	50.1 (2.3)	50.3 (1.4)	**53.3 (2.1)**	51.7 (2.7)
Web 9 ♠	68.7 (2.3)	**73.5 (2.7)**	68.5 (3.1)	50.3 (3.2)	61.8 (2.1)	**69.3 (2.2)**	65.9 (2.0)	49.8 (2.5)
Birds, Brown creeper ♦	97.4 (0.3)	89.9 (0.5)	98.8 (0.2)	**99.0 (0.1)**	92.4 (0.5)	81.8 (0.7)	95.1 (0.5)	**95.8 (0.4)**
Birds, Chestnut-backed chickadee ♦	80.1 (1.3)	85.3 (0.8)	**92.3 (0.8)**	91.7 (0.5)	80.6 (0.8)	88.4 (0.5)	**91.1 (0.4)**	86.9 (0.5)
Birds, Dark-eyed junco ♦	89.1 (1.2)	85.6 (1.3)	88.1 (1.2)	**93.2 (0.6)**	**95.8 (0.3)**	**95.8 (0.2)**	95.2 (0.3)	94.6 (0.3)
Birds, Hammonds flycatcher ♦	93.9 (0.8)	94.4 (0.7)	94.0 (0.7)	**100.0 (0.0)**	91.1 (0.8)	90.8 (0.4)	92.6 (0.4)	**99.6 (0.1)**
Birds, Hermit thrush ♦	68.2 (3.0)	57.8 (4.4)	66.2 (3.1)	**90.3 (1.1)**	96.7 (0.2)	**97.2 (0.1)**	97.0 (0.1)	95.6 (0.3)
Birds, Hermit warbler ♦	90.4 (1.3)	78.1 (1.5)	94.0 (0.6)	**98.4 (0.3)**	90.4 (0.5)	91.6 (0.3)	93.8 (0.4)	**95.2 (0.5)**
Birds, Olive-sided flycatcher ♦	92.0 (0.5)	89.6 (0.6)	95.9 (0.4)	**96.7 (0.3)**	88.7 (0.5)	84.3 (0.2)	91.3 (0.5)	**91.4 (0.5)**
Birds, Pacificslope flycatcher ♦	84.8 (0.8)	75.4 (1.0)	**98.6 (0.2)**	94.3 (0.4)	79.6 (0.8)	77.6 (0.5)	**95.4 (0.4)**	86.4 (0.6)
Birds, Red-breasted nuthatch ♦	90.7 (0.7)	87.6 (0.7)	94.6 (0.5)	**97.1 (0.3)**	88.4 (0.6)	85.0 (0.2)	90.3 (0.5)	**92.7 (0.5)**
Birds, Swainsons thrush ♦	80.4 (1.7)	76.7 (1.7)	91.4 (1.0)	**97.6 (0.3)**	86.7 (0.7)	91.4 (0.3)	93.4 (0.4)	**94.2 (0.5)**
Birds, Varied thrush ♦	95.1 (0.6)	84.0 (1.2)	93.0 (0.7)	**99.7 (0.2)**	92.7 (0.6)	88.1 (0.3)	91.4 (0.4)	**98.5 (0.2)**
Birds, Western tanager ♦	89.4 (1.6)	84.9 (1.8)	**98.9 (0.2)**	97.3 (0.3)	93.5 (0.5)	94.7 (0.3)	**97.9 (0.2)**	94.6 (0.4)
Birds, Winter wren ♦	94.6 (0.4)	93.1 (0.7)	**99.7 (0.1)**	98.8 (0.1)	91.2 (0.4)	93.4 (0.4)	**97.6 (0.3)**	94.7 (0.4)

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

Table 6. Average AUC and accuracy results of four MIL methods based on problem categories.

Dataset	**AUC**				**Accuracy**			
	MILES	$D_{meanmin}$	miFV	QP-MIL	MILES	$D_{meanmin}$	miFV	QP-MIL
Musk ♣	95.1	**96.1**	94.4	*95.7*	83.0	**88.2**	*86.5*	**88.1**
Mutagenesis ♣	70.4	74.9	*78.5*	**82.0**	*75.4*	73.9	**79.6**	74.4
Protein ♣	**95.3**	52.3	80.0	*85.1*	**94.9**	87.1	85.4	*88.3*
Elephant, Fox, Tiger ♥	**84.4**	80.0	82.1	*84.0*	*75.6*	73.9	75.2	**76.7**
Corel ♥	*94.3*	**96.2**	93.8	94.0	96.6	**97.3**	*96.7*	96.0
UCSB Breast Cancer ♥	83.3	83.1	*86.8*	**88.8**	75.8	72.2	*79.6*	**81.7**
Newsgroups ♠	51.1	**85.1**	*77.4*	**85.2**	50.3	*73.4*	69.2	**75.3**
Web recommendation ♠	*64.5*	63.2	**66.7**	58.4	71.3	*71.5*	**72.4**	65.2
Birds ♦	88.2	83.3	*92.7*	**96.5**	*89.8*	89.2	**94.0**	**93.9**
Avg.	80.7	79.3	*83.6*	**85.5**	79.2	80.8	*82.1*	**82.2**

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

Table 7. Average time results of QP-MIL.

Dataset	Instances	Features	Bags	RL time	CV time	Solution time
Musk 1 ♣	476	166	92	5.3	15.4	0.3
Musk 2 ♣	6598	166	102	59.4	187.0	2.7
Mutagenesis 1 ♣	10486	7	188	47.8	844.1	27.4
Mutagenesis 2 ♣	2132	7	42	25.7	570.8	20.0
Protein ♣	26611	8	193	125.2	782.7	13.2
Elephant ♥	1391	230	200	17.8	77.5	1.5
Fox ♥	1302	230	200	19.5	77.1	1.6
Tiger ♥	1220	230	200	16.1	69.9	1.6
Corel, African ♥	7947	9	2000	34.2	294.6	5.0
Corel, Antique ♥	7947	9	2000	36.6	346.2	7.4
Corel, Battleships ♥	7947	9	2000	34.7	339.3	6.1
Corel, Beach ♥	7947	9	2000	30.8	321.0	5.8
Corel, Buses ♥	7947	9	2000	31.7	325.8	5.8
Corel, Cars ♥	7947	9	2000	34.1	350.1	6.1
Corel, Desserts ♥	7947	9	2000	37.0	345.9	5.7
Corel, Dinosaurs ♥	7947	9	2000	35.5	329.8	6.0
Corel, Dogs ♥	7947	9	2000	35.7	338.2	6.8
Corel, Elephants ♥	7947	9	2000	32.6	341.0	6.0
Corel, Fashion ♥	7947	9	2000	37.1	350.3	6.3
Corel, Flowers ♥	7947	9	2000	41.7	336.2	6.2
Corel, Food ♥	7947	9	2000	39.8	333.7	5.8
Corel, Historical ♥	7947	9	2000	42.5	330.8	6.1
Corel, Horses ♥	7947	9	2000	37.9	330.2	6.8
Corel, Lizards ♥	7947	9	2000	32.4	321.0	5.7
Corel, Mountains ♥	7947	9	2000	34.8	370.1	6.0
Corel, Skiing ♥	7947	9	2000	40.0	316.1	5.6
Corel, Sunset ♥	7947	9	2000	41.9	362.0	12.0
Corel, Waterfalls ♥	7947	9	2000	28.4	348.6	6.6
UCSB Breast Cancer ♥	2002	708	58	33.6	31.0	0.5
Newsgroups 1, alt.atheism ♠	5443	200	100	41.2	129.6	1.8
N.g. 2, comp.graphics ♠	3094	200	100	57.1	303.1	4.5
N.g. 3, comp.os.ms-windows.misc ♠	5175	200	100	79.4	172.0	2.7
N.g. 4, comp.sys.ibm.pc.hardware ♠	4827	200	100	85.3	179.8	2.7
N.g. 5, comp.sys.mac.hardware ♠	4473	200	100	83.5	239.1	2.9
N.g. 6, comp.windows.x ♠	3110	200	100	45.2	217.6	2.8
N.g. 7, misc.forsale ♠	5306	200	100	75.7	162.9	2.4
N.g. 8, rec.autos ♠	3458	200	100	53.7	282.2	3.0
N.g. 9, rec.motorcycles ♠	4730	200	100	47.3	128.1	1.9
N.g. 10, rec.sport.baseball ♠	3358	200	100	49.4	265.0	4.0
N.g. 11, rec.sport.hockey ♠	1982	200	100	32.8	176.0	3.8
N.g. 12, sci.crypt ♠	4284	200	100	30.4	97.5	1.3
N.g. 13, sci.electronics ♠	3192	200	100	65.0	380.6	6.4
N.g. 14, sci.med ♠	3045	200	100	26.1	143.9	2.0

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

Table 7 continued.

Dataset	Instances	Features	Bags	RL time	CV time	Solution time
N.g. 15, sci.space ♠	3655	200	100	32.1	146.6	2.3
N.g. 16, soc.religion.christian ♠	4677	200	100	35.9	112.1	1.6
N.g. 17, talk.politics.guns ♠	3558	200	100	27.0	107.6	1.6
N.g. 18, talk.politics.mideast ♠	3376	200	100	40.9	181.3	2.3
N.g. 19, talk.politics.misc ♠	4788	200	100	39.3	108.7	1.5
N.g. 20, talk.religion.misc ♠	4606	200	100	36.6	113.9	1.6
Web 1 ♠	2212	5863	75	143.3	29.1	0.4
Web 2 ♠	2212	6519	75	144.8	26.1	0.4
Web 3 ♠	2212	6306	75	154.7	31.5	0.4
Web 4 ♠	2291	6059	75	142.4	26.9	0.4
Web 5 ♠	2546	6407	75	158.7	33.4	0.5
Web 6 ♠	2462	6417	75	156.7	26.2	0.4
Web 7 ♠	2400	6450	75	151.6	27.1	0.4
Web 8 ♠	2183	5999	75	137.2	23.1	0.4
Web 9 ♠	2321	6279	75	149.6	28.2	0.4
Birds, Brown creeper ♦	10232	38	548	43.3	211.8	3.6
Birds, Chestnut-backed chickadee ♦	10232	38	548	43.8	212.7	3.6
Birds, Dark-eyed junco ♦	10232	38	548	39.3	241.7	4.3
Birds, Hammonds flycatcher ♦	10232	38	548	40.9	220.2	4.3
Birds, Hermit thrush ♦	10232	38	548	48.9	228.9	3.9
Birds, Hermit warbler ♦	10232	38	548	47.6	229.7	4.0
Birds, Olive-sided flycatcher ♦	10232	38	548	46.4	232.4	4.1
Birds, Pacificslope flycatcher ♦	10232	38	548	47.4	232.6	3.9
Birds, Red-breasted nuthatch ♦	10232	38	548	46.9	225.1	3.9
Birds, Swainsons thrush ♦	10232	38	548	43.5	234.9	3.9
Birds, Varied thrush ♦	10232	38	548	49.5	237.2	4.0
Birds, Western tanager ♦	10232	38	548	48.5	241.1	4.3
Birds, Winter wren ♦	10232	38	548	44.7	239.6	4.1

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

Image classification results in Table 6 reveal that QP-MIL is broadly comparable with the competitors in all benchmarks. In text categorization, performance of QP-MIL is competitive in Newsgroups datasets and miFV is the leading method in Web recommendation datasets. QP-MIL yields the best average AUC and accuracy results in audio recording classification as verified by the reported results on Birds competition. Finally, QP-MIL has the best overall average AUC and accuracy results.

Both miFV and D_{meanmin} are bag-level methods and they are mostly tuned for computer vision and bioinformatics applications of MIL. However, QP-MIL is not tailored for a certain MIL application and overall results of this section confirm generalizability of our approach to various application domains. Without forcing the standard MIL assumption, QP-MIL matches or outperforms the state-of-the-art algorithms on a broad range of applications.

Table 7 shows the time taken up by experiments of QP-MIL on 71 datasets. Again, reported results are the averages after 5 repeats of a ten-fold cross validation. We divide the total time spent by QP-MIL into three main parts: representation learning (RL) time, inner cross-validation (CV) time and model solution time. At first, we obtain clustering-based data representation. We determine the required number of clusters on the training instances and use the resulting cluster centers to represent the training bags. Compared to the computational time on the training instances, RL time for the test bags is negligible. Therefore, we only report the RL time consumed on the training set. As described in Sect. 5.3, we report classification results after a nested cross-validation procedure. The time spent for inner cross-validation loop is the CV time. After parameter selection,

we solve QP model and record execution time of barrier algorithm as the model solution time.

Table 7 reveals that QP models are solved efficiently regardless of the dataset dimensionality. Due to the repeated solution of the QP model within each inner fold, significant amount of time is spent on parameter selection. However, RL times are considerably longer compared to CV times in Web datasets since large number of features complicates the dissimilarity calculations in data representation phase. In Mutagenesis datasets, predetermined value of the threshold controlling parameter ε may cause infeasibility in QP models. If infeasibility is detected, we solve an auxiliary optimization problem to deal with this situation. Specifically, by keeping the original constraints of (3), we convert ε into a decision variable and maximize its value. This way, a suitable value of ε is derived. Then, QP model (3) is solved after stating the selected ε value. This process increases both the CV time and model solution time on these datasets as seen in Table 7. QP-MIL provides an efficient learning approach concerning different MIL application categories. In the light of parameter sensitivity discussions in Sect. 5.4.4, QP-MIL can be implemented without parameter selection to gain from the execution time.

5.4.3 Contribution of threshold selection to model robustness

To make classification more robust, QP-MIL selects the decision threshold automatically. After each experiment, optimal decision threshold is returned with the QP solution as the value of variable τ. To observe the robustness of accuracy results of QP-MIL, we conduct a comparison via solving an extra alternative formulation. We describe another QP, QP without τ, where only variable τ is excluded and the remaining variables and constraints are the same with the original QP. After solving QP without τ, optimal decision threshold is selected on the training set. Then, testing accuracy is calculated using this threshold value.

Table 9 shows the testing accuracy results after solving both formulations for 3 different datasets. These results imply that including τ as a variable in QP elicits only negligible differences on accuracy and hence the resulting classifier. We also compare solutions of the original QP formulation and QP without τ in terms of variance. In Figure 5, the boxplots of testing accuracies on 3 datasets are provided. Figure 5 demonstrates that QP solutions with a threshold have lower variance compared to QP solutions without τ. Namely, QP-MIL results with similar accuracy and lower variance than QP without τ. Overall, QP with τ generates robust results and the embedded threshold selection is a particular advantage of the proposed method.

	Accuracy	
Dataset	QP-MIL with τ	QP-MIL without τ
Musk 1	88.4	88.4
Elephant	85.0	84.7
Fox	64.1	63.4

Table 9. Comparison of the testing accuracy results on 3 datasets computed with two different QP solutions depending on whether threshold parameter τ is included in the model or not.

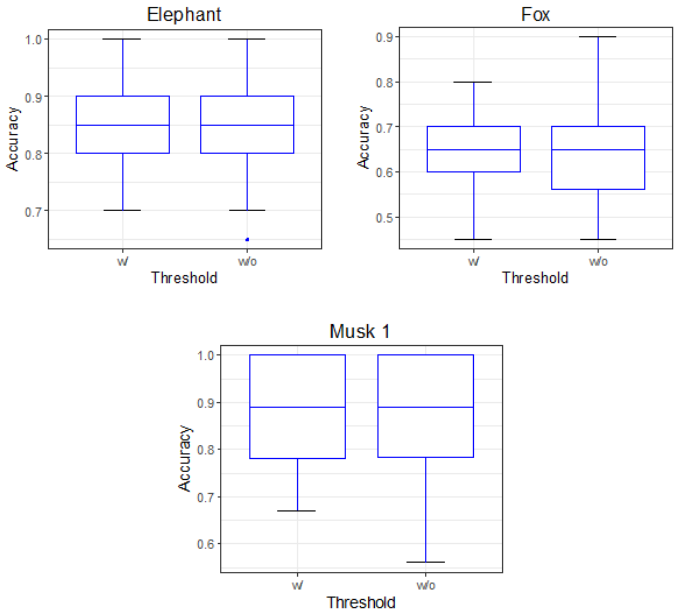

Figure 5. Boxplots of pairwise accuracy comparison of QP solution with a threshold variable τ, and QP solution without τ on 3 datasets.

5.4.4 Parameter sensitivity

In this section, we conduct experiments on four real-world datasets to examine the sensitivity of QP-MIL to C setting. Six different values of C are tested with 50 replicates of the experiments. We select the tuning set of C as $\{0.01, 0.1, 1, 10, 100, 1000\}$. We execute data representation and model solving as described in Sect. 5.3 except for the inner cross validation. For each level of C, we solve QP model (3) and record the classification results for the test bags. Figure 6 presents the behavior of the QP-MIL classifier on four datasets. For each dataset, boxplots show the AUC values for different levels of C. For Musk 2, value of C does not have a significant effect on the AUC performance. Corresponding boxplots in Figure 6 show that smaller C values yield slightly better AUC results in Elephant dataset. Finally, analysis with the boxplots in Figure 6 demonstrates that changing value of C does not significantly affect the AUC performance for other datasets.

The reported results of the comparisons with baseline approaches are provided after a cross-validation procedure in Sect. 5.4.2. The trade-off between maximization of bag class membership margin and sparsity of the weighting vector can be considered as a practically dispensable criterion for learning. Since most of the computation time is consumed by parameter selection as reported in Table 4, value of C can be fixed initially for run-time considerations. Setting a higher value of C introduces potential risk of overfitting, and therefore may reduce generalization to unknown objects. As shown in the boxplots of Figure 6, small C values yield higher

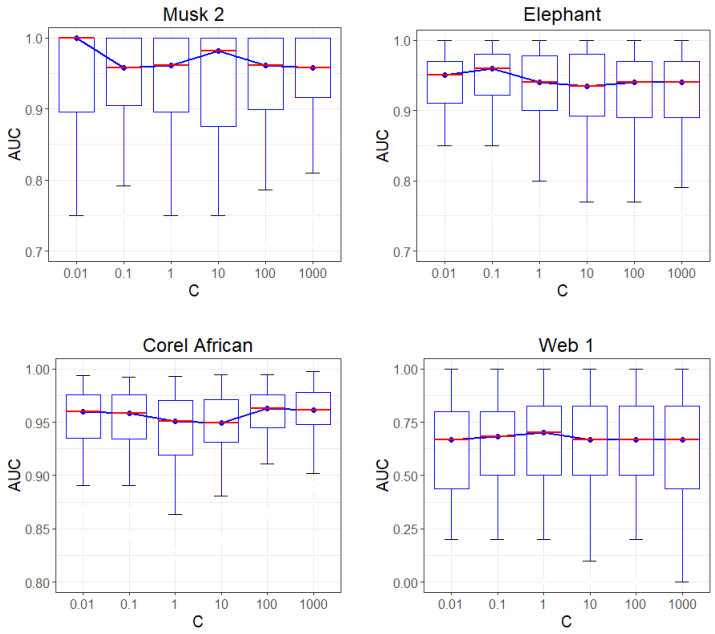

Figure 6. Sensitivity of the QP-MIL to different values for C on 4 real-world datasets.

AUC values in both Musk 2 and Elephant. Therefore, if the parameter selection phase is skipped, we suggest to use small values of C to obtain satisfactory results.

6 Conclusions

In this paper, we propose an optimization-based method, QP-MIL, to solve multiple instance classification problem, where a bag of instances are classified instead of single instances. Our algorithm is based on an quadratic programming (QP) formulation, which performs classification without imposing additional constraints on relating instance labels to the bag labels. Solving QP problem produces a decision function, which computes a bag class membership score by aggregating instance-level scores. Instance-level scores are obtained by a linear function of feature values. This way, all instances contribute to the bag label and their contributions are modeled by specifying the feature weights. The optimization process outputs a bag-level decision threshold to classify new bags together with the decision function. Distances of bag class memberships to the threshold value are maximized and the sparseness of feature weight vector is controlled by a cost parameter.

We have tested our approach on a wide range of datasets from various categories such as drug activity prediction, image categorization, text mining and audio recording classification. In order to support further research on this area, we serve the used datasets, codes and configurations on our supporting page [19]. We compared the performance of our approach to state-of-the-art machine learn-

ing based approaches. To model instance relationships, cluster centers are selected as prototypes and input features are the instance-to-prototype distances. For each dataset, generated problem instances can be easily solved to optimality in seconds. Our experiments on 71 datasets indicate that QP-MIL is competitive with the recent successful heuristic algorithms, and provides the best classification results on a variety of datasets.

Since this study focuses on optimization-based MIL, we also performed comparisons with a recent method MIHLSVM in terms of problem size and computation time. MIHLSVM solves mixed integer quadratic programs to learn a bag classifier. Our comparisons between QP-MIL and MIHLSVM indicate that MIHLSVM problem instances have difficulties to scale to large datasets. Our computational results show that direct solution of MIHLSVM is able to retrieve satisfactory solutions to MIL problem within a reasonable amount of time. Finally, we examined the effect of the cost parameter and illustrated that the classification performance does not excessively depend on adjustment of the cost parameter. Our MIL approach offers an efficient solution to MIL problem in terms of classification accuracy and model solution time, and can be extended to large real-world challenges as a future work.

Acknowledgements Z. Caner Taşkın's research was partially supported by Turkish Science Academy BAGEP award.

References

1. Amores, J.: Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence **201**, 81–105 (2013)
2. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems 15, pp. 561–568. MIT Press (2003)
3. Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X.Z., Raich, R., Hadley, S.J., Hadley, A.S., Betts, M.G.: Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. The Journal of the Acoustical Society of America **131**(6), 4640–4650 (2012)
4. Bunescu, R.C., Mooney, R.J.: Multiple instance learning for sparse positive bags. In: Proceedings of the 24th international conference on Machine learning, pp. 105–112. ACM (2007)
5. Carbonneau, M.A., Cheplygina, V., Granger, E., Gagnon, G.: Multiple instance learning: A survey of problem characteristics and applications. Pattern Recognition **77**, 329–353 (2018)
6. Chen, Y., Bi, J., Wang, J.Z.: MILES: Multiple-instance learning via embedded instance selection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(12), 1931–1947 (2006)
7. Cheplygina, V., Tax, D.M., Loog, M.: Multiple instance learning with bag dissimilarities. Pattern Recognition **48**(1), 264–275 (2015)
8. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence **89**(1), 31–71 (1997)
9. Doran, G., Ray, S.: A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. Machine Learning **97**(1-2), 79–102 (2014)
10. Erdem, A., Erdem, E.: Multiple-instance learning with instance selection via dominant sets. In: Similarity-Based Pattern Recognition, pp. 177–191. Springer (2011)
11. Fischetti, M.: Fast training of support vector machines with gaussian kernel. Discrete Optimization **22**, 183–194 (2016)
12. Foulds, J., Frank, E.: A review of multi-instance learning assumptions. The Knowledge Engineering Review **25**(01), 1–25 (2010)

13. Fu, Z., Lu, G., Ting, K.M., Zhang, D.: Learning sparse kernel classifiers for multi-instance classification. IEEE transactions on neural networks and learning systems **24**(9), 1377–1389 (2013)
14. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: Multiple instance learning with instance selection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(5), 958–977 (2011)
15. Gurobi Optimization, Inc.: Gurobi optimizer reference manual (2018). URL `http://www.gurobi.com`
16. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering **17**(3), 299–310 (2005)
17. Kandemir, M., Zhang, C., Hamprecht, F.A.: Empowering multiple instance histopathology cancer diagnosis by cell graphs. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014, pp. 228–235. Springer (2014)
18. Ketchen Jr, D.J., Shook, C.L.: The application of cluster analysis in strategic management research: an analysis and critique. Strategic management journal pp. 441–458 (1996)
19. Kucukasci, E.S., Baydogan, M.G.: Bag-level representations for multiple instance learning (2018). URL `http://ww3.ticaret.edu.tr/eskucukasci/multiple-instance-learning/`
20. Kundakcioglu, O.E., Seref, O., Pardalos, P.M.: Multiple instance learning via margin maximization. Applied Numerical Mathematics **60**(4), 358–369 (2010)
21. Li, W.J., Yeung, D.Y.: MILD: Multiple-instance learning via disambiguation. Knowledge and Data Engineering, IEEE Transactions on **22**(1), 76–89 (2010)
22. Li, Y.F., Kwok, J., Tsang, I., Zhou, Z.H.: A convex method for locating regions of interest with multi-instance learning. Machine learning and knowledge discovery in databases pp. 15–30 (2009)
23. Mangasarian, O.L., Wild, E.W.: Multiple instance classification via successive linear programming. Journal of Optimization Theory and Applications **137**(3), 555–568 (2008)
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**(Oct), 2825–2830 (2011)
25. Poursaeidi, M.H., Kundakcioglu, O.E.: Robust support vector machines for multiple instance learning. Annals of Operations Research **216**(1), 205–227 (2014)
26. Şeref, O., Chaovalitwongse, W.A., Brooks, J.P.: Relaxing support vectors for classification. Annals of Operations Research **216**(1), 229–255 (2014)
27. Srinivasan, A., Muggleton, S., King, R.: Comparing the use of background knowledge by inductive logic programming systems. In: Proceedings of the 5th International Workshop on Inductive Logic Programming, pp. 199–230. Department of Computer Science, Katholieke Universiteit Leuven (1995)
28. Tao, Q., Scott, S., Vinodchandran, N., Osugi, T.T.: SVM-based generalized multiple-instance learning via approximate box counting. In: Proceedings of the twenty-first international conference on Machine learning, p. 101. ACM (2004)
29. Tax David M. J., C.V.: MIL, A Matlab toolbox for multiple instance learning (2015). URL `http://prlab.tudelft.nl/david-tax/mil.html`. Version 1.1.0
30. Tax, D.M., Duin, R.P.: Learning curves for the analysis of multiple instance classifiers. In: Structural, Syntactic, and Statistical Pattern Recognition, pp. 724–733. Springer (2008)
31. Tax, D.M., Hendriks, E., Valstar, M.F., Pantic, M.: The detection of concept frames using clustering multi-instance learning. In: Pattern Recognition (ICPR), 2010 20th International Conference on, pp. 2917–2920. IEEE (2010)
32. The Mathworks, I.: MATLAB version 8.5.0.197613 (R2015a). Natick, Massachusetts (2015)
33. Wei, X.S., Wu, J., Zhou, Z.H.: Scalable algorithms for multi-instance learning. IEEE transactions on neural networks and learning systems **28**(4), 975–987 (2017)
34. Weidmann, N., Frank, E., Pfahringer, B.: A two-level learning method for generalized multi-instance problems. In: Machine Learning: ECML 2003, pp. 468–479. Springer (2003)
35. Zhou, Z.H., Jiang, K., Li, M.: Multi-instance learning based web mining. Applied Intelligence **22**(2), 135–147 (2005)
36. Zhou, Z.H., Sun, Y.Y., Li, Y.F.: Multi-instance learning by treating instances as non-iid samples. In: Proceedings of the 26th annual international conference on machine learning, pp. 1249–1256. ACM (2009)
37. Zhou, Z.H., Xu, J.M.: On the relation between multi-instance learning and semi-supervised learning. In: Proceedings of the 24th international conference on Machine learning, pp. 1167–1174. ACM (2007)
38. Zhou, Z.H., Zhang, M.L.: Solving multi-instance problems with classifier ensemble based on constructive clustering. Knowledge and Information Systems **11**(2), 155–170 (2007)