Improving Multiple-Instance Learning via Disambiguation by Considering Generalization

Lu Zhao[1(✉)], Youjian Yu[1], Hao Chen[1], and Liming Yuan[2]

[1] Tianjin Chengjian University, Tianjin 300384, China
zhaolu6892@163.com
[2] Tianjin University of Technology, Tianjin 300384, China

Abstract. Multiple-instance learning (MIL) is a variant of the traditional supervised learning. In MIL training examples are bags of instances and labels are associated with bags rather than individual instances. The standard MIL assumption indicates that a bag is labeled positive if at least one of its instances is labeled positive, and otherwise labeled negative. However, many MIL problems do not satisfy this assumption but the more general one that the class of a bag is jointly determined by multiple instances of the bag. To solve such problems, the authors of MILD proposed an efficient disambiguation method to identify the most discriminative instances in training bags and then converted MIL to the standard supervised learning. Nevertheless, MILD does not consider the generalization ability of its disambiguation method, leading to inferior performance compared to other baselines. In this paper, we try to improve the performance of MILD by considering the discrimination of its disambiguation method on the validation set. We have performed extensive experiments on the drug activity prediction and region-based image categorization tasks. The experimental results demonstrate that MILD outperforms other similar MIL algorithms by taking into account the generalization capability of its disambiguation method.

Keywords: Multiple-instance learning · Disambiguation
Generalization ability

1 Introduction

Multiple-instance learning (MIL) copes with the classification of training bags each of which is composed of one or more training instances [5,7]. Labels are associated with bags rather than any individual instance. The standard multiple-instance assumption indicates that a bag is labeled positive if *at least* one of its instances is positive, and otherwise labeled negative.

Several researchers have made a more general multiple-instance assumption that a bag is labeled as a ceratin class only when several different instances co-appear in the bag. Under this general assumption, they further proposed

C. Li and S. Mao (Eds.): WiCON 2017, LNICST 230, pp. 419–429, 2018.
https://doi.org/10.1007/978-3-319-90802-1_37

several embedded-space MIL algorithms [3,4,8,10]. The basic idea can be summarized as: (1) selecting some instance prototypes from the training set to form an embedded-space, (2) embedding every bag into the embedded-space by computing the distances/similarities between this bag and those instance prototypes, (3) using the new bag-level features for training bags to learn an support vector machine (SVM).

MILD is a very efficient and robust embedded-space MIL algorithm, which has been demonstrated by Li and Yeung [10]. MILD focuses on the ability of a candidate instance prototype in separating positive and negative training bags. However, it ignores the discriminative ability of an instance on the validation set, or in other words does not consider the generalization capability of its disambiguation method, leading to inferior performance as compared to other similar algorithms.

In this paper, we attempt to improve the performance of MILD by taking into account the generalization ability of its disambiguation method. The main idea is dividing the training set into a training set and a validation set, and using the discrimination of a candidate instance prototype on the validation set as the evaluation standard of its discriminability. We name the new variant of MILD Multiple-Instance Learning via Generalized Disambiguation (MILGD). The experimental results show that MILGD outperforms other embedded-space MIL algorithms with respect to classification accuracy and robustness to labeling noise.

The remainder of the paper is organized as follows. In Sect. 2, we review some work related to our research. In Sect. 3, we first analyse the characteristics of MILD and then propose our new algorithm. We then compare our MILGD algorithm with other baselines using two kinds of MIL data sets in Sect. 4. We conclude this paper in the last section.

2 Related Work

Dietterich et al. [6] proposed the first MIL algorithm called Axis-Parallel Rectangle (APR). The main idea is trying to find an APR in the feature space, which includes *at least* one instance from every positive training bag but excludes all instances from the negative training bags. Before long, Maron and Lozano-Prez [12] presented a similar concept named Diverse Density (DD) to solve the MIL problem. DD actually describes the likelihood that a possible concept appears in all positive bags and does not appear in any negative bag at the same time. Zhang and Goldman [19] extended the DD concept into the Expectation Maximization (EM) framework and proposed the EM-DD algorithm in order to locate the target concept in a more efficient way. Since learning a single concept may be insufficient to capture the multi-modal distribution, GEM-DD applies the Quasi-Newton approaches to search for a group of concepts in an iterative way [14].

Ramon and De Raedt [15] adapted the neural networks to the MIL context. Later on, Zhang and Zhou [18] derived a similar framework to tackle the MIL

problem. Wang and Zucker [17] used the Hausdorff distance to measure the distances between different bags and extended the standard k-Nearest Neighbor (kNN) algorithm into the multiple-instance setting. Gärtner et al. [9] developed a multiple-instance kernel function such that SVMs can be applied directly for the training bags. Andrews et al. [1] regarded the unobservable instance labels as hidden variables and formulated MIL as mixed integer quadratic programs. Settles et al. [16] constructed a multiple-instance framework with active learning and showed that instance labels are beneficial for improving the performance of an MIL learner. Motivated by the subgradient-based approaches for SVMs, Bergeron et al. [2] proposed a non-convex bundle method for optimizing the multiple-instance objective. Li et al. [11] assume that the distribution of instances is a mixture of the concept and non-concept. Under this assumption, they constructed an ensemble of several classifiers for classifying bags. Nguyen et al. [13] provided a generic framework used to convert the rule-based algorithms into MIL algorithms.

Several researchers have attempted to use the embedded-space algorithms to solve the MIL problem. Various embedded-space algorithms are different with each other in that they choose from training bags the instance prototypes. Specifically, DD-SVM [4] depends on DD for choosing the instance prototypes. DD-SVM regards the instances with the local maximal DD value as prototypes. MILES [3] regards all instances in the training set as valid prototypes and selects a subset of them via learning a 1-norm linear SVM that is known to produce sparse solutions for feature weights [20]. MILD [10] depends on a conditional probability model for the instance selection. The instance possessing the highest capability in classifying the training bags is considered as a prototype. MILIS [8] achieves the initial instance selection by modeling the distribution of the negative population with a Gaussian-kernel-based kernel density estimator. Then it depends on an iterative optimization framework for the instance selection and classifier learning.

3 MILGD: A Variant of MILD

In this section, we first analyse the characteristics and bias of the MILD algorithm in order to introduce our MILGD algorithm. Then MILGD algorithm is proposed to improve the performance of MILD by considering the discrimination of its disambiguation method on the validation set. Let B^+ denote all positive training bags and B^- all negative training bags. m^+ and m^- are the size of B^+ and B^-, respectively. B is the union of B^+ and B^-, and m is the sum of m^+ and m^-. We denote the i^{th} positive bag in B^+ as B_i^+ and the j^{th} instance in that bag as B_{ij}^+. The bag B_i^+ is composed of n_i^+ instances B_{ij}^+, $j = 1, \ldots, n_i^+$. When the label of a bag does not matter, we simply denote the bag as B_i and its instances as B_{ij}. $l(B_i) \in \{+1, -1\}$ denotes the label of B_i and $l(B_{ij}) \in \{+1, -1\}$ that of B_{ij}. Note that the instance labels are not directly observable.

3.1 Analysis of MILD Characteristics

From the MIL formulation, we know that instances in each negative bag are all negative, so for negative bags there is no ambiguity on the labels of instances. However, a positive bag may contain not only positive instances but also negative instances and the labels of instances are unknown therein. Therefore, the ambiguity in instance labels in MIL arises in the positive bags and MILD is thus aimed at identifying the *true* positive instances in the positive bags.

Assumption 1. *Given a true positive instance t, the* ***probability*** *that an instance B_{ij} is positive is calculated as*

$$\Pr(l(B_{ij}) = +1 \mid t) = \exp(-\frac{\parallel t - B_{ij} \parallel^2}{\sigma_t^2}), \tag{1}$$

where $\parallel x \parallel \triangleq \sqrt{\sum_k x_k^2}$ denotes the 2-norm of the vector x, and σ_t is a parameter larger than 0.

Assumption 1 is used to compute a conditional probability. From (1), we can easily see that $0 \leq \Pr(l(B_{ij}) = +1 \mid t) \leq 1$, $\Pr(l(B_{ij}) = +1 \mid t) = 0$ when $\parallel t - B_{ij} \parallel = +\infty$ and $\Pr(l(B_{ij}) = +1 \mid t) = 1$ when $\parallel t - B_{ij} \parallel = 0$. This is well consistent with our intuition. If t is a true positive instance and $\parallel t - B_{ij} \parallel = 0$, B_{ij} will definitely be a true positive instance since B_{ij} is just equal to t, which indicates that $\Pr(l(B_{ij}) = +1 \mid t) = 1$ is reasonable. Similarly, if $\parallel t - B_{ij} \parallel = +\infty$, B_{ij} will be infinitely far away from the true positive instance t, which means that $\Pr(l(B_{ij}) = +1 \mid t) = 0$ is also reasonable. The farther B_{ij} is away from t, the lower is the probability that B_{ij} is positive given t, which is reasonable based on our intuition. Based on Assumption 1, MILD defines the probability that a bag is positive as follows.

Definition 1. *The* ***most-likely-cause estimator*** *for estimating the probability that a bag B_i is positive given a true positive instance t is defined as*

$$\begin{aligned} Pr(l(B_i) = +1 \mid t) &= \max_{B_{ij} \in B_i} Pr(l(B_{ij}) = +1 \mid t) \\ &= \max_{B_{ij} \in B_i} \exp(-\frac{\parallel t - B_{ij} \parallel^2}{\sigma_t^2}) \\ &= \exp(-\frac{d^2(t, B_i)}{\sigma_t^2}), \end{aligned} \tag{2}$$

where

$$d(t, B_i) = \min_{B_{ij} \in B_i} \parallel t - B_{ij} \parallel . \tag{3}$$

In other words, the distance $d(t, B_i)$ between an instance t and a bag B_i is simply equal to the distance between t and its nearest instance in B_i.

The definition of the most-likely-cause estimator implies that the label of a bag is most probably determined by a specific instance in it which is nearest to

the true positive instance t. In general, the larger $d(t, B_i)$ is, the lower is the probability that B_i is positive given the true positive instance t. Based on the definition of the most-likely-cause estimator, MILD gives the following theorem.

Theorem 1. *Given a true positive instance t, there exists a threshold θ_t which makes the decision function defined in (4) label the bags according to the Bayes decision rule.*

$$h^t_{\theta_t}(B_i) = \begin{cases} +1, \ d(t, B_i) \leq \theta_t, \\ -1, \ otherwise. \end{cases} \tag{4}$$

For simplicity, we ignore the proof of Theorem 1 and refer the interested readers to [10] for details. Therefore, if t is a true positive instance, there must exist a decision function as defined in (4) to label the bags well, implying that the distances from the true positive instance t to the positive bags are expected to be less than those to the negative bags. Since the positive bags may also contain negative instances just like the negative bags, the distances from a negative instance to the positive bags may be as random as those to the negative bags. Thus, for a negative instance its distances to the positive and negative bags do not exhibit the same distribution as those from t. MILD thus uses this distributional difference to identify the true positive instances. The following definition and theorem form the basis of its disambiguation method.

Definition 2. *The **empirical precision** of the decision function in (4) is defined as*

$$P_t(\theta_t) = \frac{1}{m} \sum_{i=1}^{m} \frac{1 + h^t_{\theta_t}(B_i) l(B_i)}{2}. \tag{5}$$

The empirical precision essentially measures how well the decision function $h^t_{\theta_t}(\cdot)$ with threshold θ_t mimics $l(\cdot)$ in predicting the bag labels. If t is a true positive instance, it can label the bags well according to Theorem 1, and thus the best (maximum) empirical precision $P^*(t)$ for t will be high. In contrast, if t is a negative instance, it cannot label the bags well, and thus $P^*(t)$ for t will be low. In essence, $P^*(t)$ reflects the ability of instance t in discriminating the training bags. The larger $P^*(t)$ is, the more likely t is a true positive instance. The remaining issue is how to compute $P^*(t)$ given an instance t. Theorem 2 provides the solution to this problem. Note that

$$P^*(t) = \max_{\theta_t} P_t(\theta_t), \tag{6}$$

$$\theta^*_t = \arg\max_{\theta_t} P_t(\theta_t). \tag{7}$$

Theorem 2. *The best empirical precision $P^*(t)$ for t is achieved when θ_t is an element in the set $\{d(t, B^+_i) \mid B^+_i \in B^+\}$.*

Therefore, to obtain the best empirical precision $P^*(t)$ given an instance t, we only need to compute the distance from t to every positive training bag. Given all

of the above knowledge, Li and Yeung [10] proposed their MILD algorithm. In their algorithm, they select from every positive training bag the instance with the largest P^* value as a candidate true positive instance (instance prototype).

When the disambiguation process is completed, MILD maps every training bag B_i to a point $D(B_i)$ in the embedded-space composed of all the instance prototypes, and then learns a SVM with a Gaussian kernel on all new features for bags. The new bag-level features for a bag B_i is defined as

$$D(B_i) = [d(t_1, B_i), \ldots, d(t_{m^+}, B_i)]^{\mathrm{T}}, \tag{8}$$

where $t_k \in T$ and T is the set of instance prototypes.

3.2 MILGD

Following the above description, we know that $P^*(t)$ describes the ability of an instance t in classifying the training bags. MILD just uses $P^*(t)$ as its instance selection principle. However, MILD computes $P^*(t)$ for an instance t with all the training examples and does not consider the discriminative ability of t on unknown examples. As we know, this kind of practice cannot guarantee the generalization ability of a method, specifically, the disambiguation method of MILD herein.

To solve this problem, we can group all the training bags into a training set and a validation set. Given an instance t, we first compute the best threshold θ_t^* on the training set, which corresponds to the maximum empirical precision $P^*(t)$. Then we compute the value of $P_t(\theta_t^*)$ on the validation set. This process can be considered as one fold of n-fold cross-validation. When the cross-validation approach is applied, we use the mean of $P_t(\theta_t^*)$ on all the folds to estimate the discriminability of the instance t. This is the main idea of our MILGD algorithm. Algorithm 1 summarizes the disambiguation process presented here. Note that MILGD assumes that a target concept (instance prototype) can be related to either positive bags or negative bags, whereas, in MILD the target concept is defined for positive bags only. As in DD-SVM [4], negative instance prototypes can be computed in exactly the same fashion after negating the labels of positive and negative bags.

Following Algorithm 1, we know that the main difference of MILD and MILGD lies in the instance selection standard. MILD regards the discriminative ability of an instance t on the training set as its instance selection standard, and thus MILD does not consider the discrimination of unknown examples. In contrast, MILGD regards the discriminability of t on the validation set as its instance selection standard, and hence MILGD takes into account the generalization ability of its disambiguation method. As shown in Sect. 4, this transition of instance selection standard can lead to improved performance and robustness to labeling noise. As for the bag-level feature mapping and classifier learning, there is no difference between MILGD and MILD.

Algorithm 1. Instance-Selection Method of MILGD

Input: Set of training bags B and fold number n
Output: Set of instance prototypes T

T_p = **LearnIPs**(B, n)
Negate labels of all bags in B
T_n = **LearnIPs**(B, n)
$T = T_p \cup T_n$

LearnIPs

Partition B into n subsets $\{B_1, \ldots, B_n\}$
for $B_i^+ \in B^+$ **do**
  for $B_{ij}^+ \in B_i^+$ **do**
    for $k = 1$ to n **do**
      Compute $\theta^*_{B_{ij}^+}$ on $\{B_1, \ldots, B_{k-1}, B_{k+1}, \ldots, B_n\}$ according to (7)
      Compute $P^k_{B_{ij}^+}(\theta^*_{B_{ij}^+})$ on B_k according to (5)
    end for
    $P(B_{ij}^+) = \frac{1}{n}\sum_{k=1}^{n} P^k_{B_{ij}^+}$
  end for
  $t = arg\,max_{B_{ij}^+ \in B_i^+} P(B_{ij}^+)$
  Add t to T
end for

4 Experiments

In this section, we compare the performance and efficiency of the proposed MILGD algorithm with that of other MIL algorithms using two kinds of data sets.

4.1 Drug Activity Prediction

Experimental Setup. The MUSK data sets, MUSK1 and MUSK2, are standard benchmarks for MIL, which are publicly available from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). These data sets consist of descriptions of molecules and the task is to predict whether a given molecule is active or inactive. Each molecule is viewed as a bag, the instances of which are the different low-energy conformations of the molecule. Surface properties of a conformation are extracted as its feature vector that has 166 dimensions. If one of the conformations of a molecule binds well to the target protein, the molecule is said active, and otherwise inactive. MUSK1 contains 47 positive bags and 45 negative bags, with an average of 5.17 instances per bag. MUSK2 contains 39 positive bags and 63 negative bags, with 64.69 instances per bag on average. MUSK2 shares 72 molecules with MUSK1, but contains more conformations for those shared molecules.

The parameter n (fold number in Algorithm 1) was set to be 2 for MILGD. We used LIBSVM (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) to train all the SVMs in our experiments. We chose the regularization parameter C and

Gaussian kernel parameter γ from $\{2^{-10}, 2^{-8}, \ldots, 2^{10}\}$ using a twofold cross-validation on the training bags. As for other embedded-space MIL algorithms used for comparison, we asided by the same parameter selecting principle described here.

For the MUSK data sets, we applied ten different random runs of tenfold cross-validation to test various embedded-space MIL algorithms. We thus reported the mean and 95% confidence interval of the results of ten runs of tenfold cross-validation.

Classification Results. Table 1 reports the classification accuracies for different embedded-space MIL algorithms on the MUSK data sets. We also list some other results on the same data sets for comparison. From Table 1, we can see that APR gives the best performance on MUSK1 and MUSK2. However, the APR algorithm chooses the parameters to maximize the performance on the test set rather than the training set, and thus the superiority of APR should not be interpreted as a failure. It can also be observed from Table 1 that our MILGD algorithm is superior to other algorithms in terms of the average prediction accuracy over the two data sets, in particular, the embedded-space MIL algorithms. Furthermore, the classification accuracies of MILGD are much higher than those of MILD, which demonstrates that considering the generalization ability is indeed very helpful for MILD.

Table 1. Classification accuracies (%) for various MIL algorithms on MUSK.

Algorithm	MUSK1	MUSK2	Avg.
MILGD	87.7:[86.2, 89.2]	88.1:[86.6, 89.5]	87.9
MILD [10]	85.0:[82.8, 87.1]	85.0:[83.6, 86.5]	85.0
DD-SVM [4]	85.6:[83.9, 87.2]	87.3:[86.3, 88.2]	86.5
MILES [3]	86.6:[84.9, 88.4]	88.3:[86.8, 89.9]	87.5
MILIS [8]	86.4:[84.6, 88.2]	88.3:[87.2, 89.5]	87.4
APR [6]	92.4	89.2	90.8
DD [12]	88.9	82.5	85.7
EM-DD [19]	84.8	84.9	84.9
MI-SVM [1]	77.9	84.3	81.1
mi-SVM [1]	87.4	83.6	85.5

Computation Time. Then we evaluate the efficiency of our MILGD algorithm. Following the description of MILGD, we know that MILGD divides the whole training set into different parts (or folds) for instance selection, and then uses one part for validating and the remaining parts for training in each iteration. Therefore, the more the fold number n is, the slower MILGD is. In general, MILGD is less efficient than MILD. However, the disambiguation process of

Table 2. Computation time (minutes) for various embedded-space MIL algorithms on MUSK.

Algorithm	MUSK1	MUSK2
MILGD	0.06	2.70
MILD [10]	0.03	0.42
DD-SVM [4]	8.74	122.57
MILES [3]	0.11	4.14
MILIS [8]	6.02	3091.39

MILD itself is very fast, and thus MILGD can still accomplish the instance selection process very quickly. Table 2 reports the computation time of tenfold cross-validation for different embedded-space MIL algorithms on the MUSK data sets. From Table 2, we can see that MILD is the most efficient one while DD-SVM and MILIS are the least efficient ones among all the embedded-space MIL algorithms. But other than that, the computation time of other algorithms (i.e., MILGD and MILES) is on the same order of magnitude.

4.2 Region-Based Image Categorization

Experimental Setup. The COREL data sets have been widely used for region-based image categorization. The data sets contain 20 thematically diverse image categories with 100 images of size 384×256 or 256×384 in each category. Each image is segmented into several local regions and features are extracted from each region. The data sets and extracted features are available at http://www.cs.olemiss.edu/~ychen/ddsvm.html. Details of segmentation and feature extraction are beyond the scope of this paper and interested readers are referred to [4] for further information.

Two tests have been performed for the COREL data sets, i.e., 10-category and 20-category image categorizations. In the first test, we used the first 10 categories. In the second test, we used all 20 categories. we randomly chose from each category half of images as the training bags, and we used the remaining half as the test bags. The SVM parameters were tuned in the same way as for the MUSK data sets and the fold number n was still set to be 2. We repeated the above procedure five different times. Thus, we reported the classification accuracy over five different test sets and the corresponding 95% confidence interval. Since this is a classification problem for multi-class, the simple one-against-the-rest strategy is applied for training 10/20 binary SVMs. Therefore, the category having the largest decision value given by the SVMs is assigned to the unknown bag.

Categorization Results. We provide the classification accuracies for MILGD and other embedded-space MIL algorithms on the COREL data sets in Table 3. On both data sets, the performance of MILGD is better or highly comparable with that of other algorithms. With respect to the average accuracy over the two

Table 3. Classification accuracies (%) for various embedded-space MIL algorithms on COREL.

Algorithm	COREL_{10}	COREL_{20}	Avg.
MILGD	83.2:[81.4, 85.0]	69.9:[68.8, 71.0]	76.6
MILD [10]	80.1:[77.9, 82.3]	66.8:[65.5, 68.1]	73.5
DD-SVM [4]	73.0:[71.8, 74.1]	54.3:[51.0, 57.7]	63.7
MILES [3]	82.0:[81.2, 82.9]	69.9:[68.3, 71.6]	76.0
MILIS [8]	81.2:[79.3, 83.2]	69.7:[67.2, 72.1]	75.5

tests, MILGD outperforms all the other embedded-space MIL algorithms. Particularly, the performance of MILGD is significantly better than that of MILD, which indicates again that taking into account the generalization ability is very important for the MILD algorithm. Based on the better results on the COREL data sets, we can conclude that our MILGD algorithm is very promising for the applications satisfying the general multiple-instance assumption mentioned in Sect. 1.

5 Conclusions

In this paper, we have proposed a variant of MILD called MILGD. The goal of the study was to improve the performance of MILD via the consideration of the generalization capability of its disambiguation method. The experimental results indicate that its prediction ability can be significantly improved when taking into account the generalization capability. Moreover, due to the transition of instance selection principle (from focusing on the discriminative ability on the training set to focusing on that on the validation set), MILGD achieves the best performance as compared to other state-of-the-art embedded-space MIL algorithms.

Acknowledgements. This research has been supported by the Open Project of Key Laboratory from Ministry of Education (TJUT-CVS20170001), the Tianjin Technology Project (14ZCZDGX00868), Science and Technology Transformation Award Special Fund Project of Tianjin Chengjian University in 2017 (KJZH-A1-1709), and the Basic Research Foundation of Tianjin Chengjian University (2016CJ11).

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems, pp. 561–568. MIT Press (2003)
2. Bergeron, C., Moore, G., Zaretzki, J., Breneman, C.M., Bennett, K.P.: Fast bundle algorithm for multiple-instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **34**(6), 1068–1079 (2012)

3. Chen, Y., Bi, J., Wang, J.Z.: MILES: multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 1931–1947 (2006)
4. Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. J. Mach. Learn. Res. **5**, 913–939 (2004)
5. Cinbis, R.G., Verbeek, J., Schmid, C.: Weakly supervised object localization with multi-fold multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **39**(1), 189–203 (2016)
6. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. **89**(1), 31–71 (1997)
7. Durand, T., Thome, N., Cord, M.: WELDON: weakly supervised learning of deep convolutional neural networks. In: Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, pp. 4743–4752. IEEE, Washington (2016)
8. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: multiple instance learning with instance selection. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 958–977 (2011)
9. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: International Conference on Machine Learning, pp. 179–186. Morgan Kaufmann (2002)
10. Li, W.J., Yeung, D.Y.: MILD: multiple-instance learning via disambiguation. IEEE Trans. Knowl. Data Eng. **22**(1), 76–89 (2010)
11. Li, Y., Tax, D.M.J., Duin, R.P.W., Loog, M.: Multiple-instance learning as a classifier combining problem. Pattern Recogn. **46**(3), 865–874 (2013)
12. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Advances in Neural Information Processing Systems, pp. 570–576 (1998)
13. Nguyen, D.T., Nguyen, C.D., Hargraves, R., Kurgan, L.A., Cios, K.J.: mi-DS: multiple-instance learning algorithm. IEEE Trans. Syst. Man Cybern. Part B-Cybern. **43**(1), 143–154 (2013)
14. Rahmani, R., Goldman, S.A., Zhang, H., Cholleti, S.R., Fritts, J.E.: Localized content-based image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1902–1912 (2008)
15. Ramon, J., De Raedt, L.: Multi instance neural networks. In: International Conference on Machine Learning Workshop on Attribute-Value and Relational Learning (2000)
16. Settles, B., Craven, M., Ray, S.: Multiple-instance active learning. In: Advances in Neural Information Processing Systems, pp. 1289–1296. MIT Press (2008)
17. Wang, J., Zucker, J.D.: Solving multiple-instance problem: a lazy learning approach. In: International Conference on Machine learning, pp. 1119–1126. Morgan Kaufmann (2000)
18. Zhang, M.L., Zhou, Z.H.: Improve multi-instance neural networks through feature selection. Neural Process. Lett. **19**(1), 1–10 (2004)
19. Zhang, Q., Goldman, S.A.: EM-DD: an improved multiple-instance learning technique. In: Advances in Neural Information Processing Systems, pp. 1073–1080. MIT Press (2001)
20. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. In: Advances in Neural Information Processing Systems, pp. 49–56. MIT Press (2004)