Multiple Instance Real Boosting with Aggregation Functions

Hossein Hajimirsadeghi and Greg Mori
Simon Fraser University
hosseinh@sfu.ca, mori@cs.sfu.ca

Abstract

We introduce a boosting framework for multiple instance learning (MIL) with varied aggregation of instances. In this framework, a diverse set of aggregation functions can be used to refine the notion of a positive bag for multiple instance learning. We investigate the effect of a wide range of orness in aggregation, using ordered weighted averaging. Thus, we obtain a new notion of a positive bag, which can represent different levels of ambiguity. We evaluate the performance of the proposed algorithm on popular MIL datasets. The experimental results show that this algorithm outperforms the standard MILBoost algorithm.

1. Introduction

Multiple instance learning (MIL) is used to handle ambiguity in weakly supervised data. In MIL, training data are presented in positive and negative bags instead of individual instances. A positive bag label means that it contains at least one positive example, while in a negative bag all the instances are negative. The ambiguity in the examples is passed on to the learning algorithm, which should incorporate the information to find a suitable classifier. MIL has been extensively used in different applications, especially vision tasks. It has been successfully used to train classifiers for object detection [13], image categorization [4], image retrieval [10, 6], and object tracking [2] from weakly annotated data. For example, Viola et al. [13] use MIL to model imperfection in positive labels for face detection – a bag consists of a set of windows centered around a ground-truth face location. At least one of these windows should be a good ground truth face. Chen et al. [4] employ a diverse density (DD) function to map the instances of a bag into a bag-level feature vector. Then, the important features are chosen by 1-norm SVM and used for image categorization. Gehler and Chapelle [8] approach MIL with SVMs, using deterministic annealing based optimization. They also claim that different levels of ambiguity in bags can influence the performance of MIL-based methods. Hence, in their proposed algorithm they provide the possibility to encode prior knowledge about the dataset (i.e., fraction of positives in a bag). Bunescu and Mooney [3] use the framework of transductive SVMs to propose a MIL algorithm for sparse positive bags. They show that this algorithm is very effective for the tasks where there are few positive instances in the positive bags (e.g., image region classification). Duan et al. [6] and Li et al. [10] formulate text-based image retrieval as a MIL problem by treating the relevant and irrelevant clustered images as positive and negative bags. To come up with this problem, they introduce a generalized definition of MIL, where the bags contain at least a certain portion of positive instances. They use a SVM formulation with new constraints on instance labels of the bags to develop algorithms, which model the ambiguities in the instances.

In this work, we propose a novel algorithm called MIRealBoost to train a bag-level classifier. The main advantage of our framework is that a diverse set of aggregation functions can be used to model different levels of ambiguity in the data. Our notion of positive bag can range from at least one instance in the bag is positive to all instances are positive. This is different from algorithms such as [8, 6, 10], which need prior knowledge about fraction of positives inside bags. Instead, our proposed framework can roughly extract this knowledge by exploring different aggregation functions and directly optimizes the expected bag-level log likelihood to find the bag-level classifier. In addition, this algorithm has the general advantages of boosting algorithms like simple programming, few parameters for tunning, and ability of feature selection.

This paper is organized as follows. Section 2 describes our framework of multiple instance learning with aggregation functions. In particular, ordered weighted averaging and the proposed MIRealBoost algorithm are explained in this section. In Section 3 the experiments are presented, and MIRealBoost is compared with the state-of-the-art algoithms. Finally, the conclusions are drawn in Section 4.

2. Algorithm Design

In the MIL framework, training examples are not singletons. Instead, they are presented in bags (i.e. sets of instances), where the instances in a bag share a label. Let $X_i = \{x_{i1}, \cdots, x_{i|X_i|}\}$ denote a bag with $|X_i|$ instances and a binary label $Y_i \in \{-1, 1\}$. The whole data set is represented by $\{(X_1, Y_1), \cdots, (X_N, Y_N)\}$. In the MIL literature, a positive bag means at least one of the instances in the bag is positive. In a negative bag, all the instances are negative. Viola et al. [13] introduced an algorithm MILBoost, based on the AnyBoost framework [12]. This algorithm trains a boosting classifier which maximizes the log likelihood of the training bags:

$$\begin{aligned} L &= \sum_i \mathbf{1}(Y_i = 1) \log p(X_i) \\ &\quad + \mathbf{1}(Y_i = -1) \log (1 - p(X_i)), \end{aligned} \quad (1)$$

where $p(X_i)$ is the probability of the ith bag being positive and expressed in terms of its instances by the Noisy-OR (NOR) model:

$$p(X_i) = 1 - \prod_{x_{ij} \in X_i} (1 - p(x_{ij})). \quad (2)$$

The rationale for this model is that the probability of a bag being positive is high if at least one of the instances has high probability. In this paper, we propose an algorithm which maximizes the expected log likelihood of training examples based on RealBoost framework [7] by defining a function as the strong classifier for bags of any size. Moreover, besides the NOR model, we use a class of operators which can express different linguistic aggregation instructions. Hence, the concept of positive bags is extended to a wider range of definitions. For example, a bag might be called positive if *a few* instances inside the bag are positive, or *some of* the instances are positive or *half of* the instances are positive.

2.1. Ordered Weighted Averaging

Ordered Weighted Averaging (OWA) as an aggregation operator was proposed by Yager [14] . OWA is a mapping $\mathbf{owa} : [0, 1]^n \rightarrow [0, 1]$, which aggregates a list of arguments $A = \{a_1, a_2, \cdots, a_n\}(a_j \in [0, 1])$ with an associated weight vector $W = [w_1, w_2, \cdots, w_n]$ ($w_i \in [0, 1]$, $\sum w_i = 1$) according to (3).

$$\mathbf{owa}(a_1, a_2, \cdots, a_n) = \sum_{i=1}^{n} b_i w_i. \quad (3)$$

Where b_i is the ith largest of the a_j. OWA can be used to model a spectrum of linguistic aggregation instructions. The degree of orness or *optimism degree* (θ) for an OWA operator denotes its closeness to OR operator and is defined as follows:

$$\theta(w_1, w_2, \cdots, w_n) = \left(\frac{1}{n-1}\right) \sum_{i=1}^{n} ((n-i) w_i). \quad (4)$$

Using linguistic quantifiers is one of the approaches used to determine the weights of OWA operators. Here, we use the regular increasing monotonic (RIM) linguistic quantifier $Q : [0, 1] \rightarrow [0, 1]$ such that $Q(0) = 0$ and $Q(1) = 1$. Consequently, the OWA weight vector is computed from Q using (5).

$$w_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n}). \quad (5)$$

A popular form is $Q(p) = p^\alpha$, in which α is the parameter to be set. For this function, seven RIM quantifiers have been suggested [11, 14]: *At least one* ($\alpha \rightarrow 0$, i.e. *Max* function), *Few* ($\alpha = 0.1$), *Some* ($\alpha = 0.5$), *Half* ($\alpha = 1$), *Many* ($\alpha = 2$), *Most* ($\alpha = 10$), *All* ($\alpha \rightarrow \infty$), which we also demonstrate in Table 1.

Table 1. Family of RIM qunatifiers and their relevant values of α and θ

Linquistic quantifier	α	Orness (θ)
At least one of them	$\alpha \rightarrow 0$	0.999
Few of them	0.1	0.909
Some of them	0.5	0.667
Half of them	1	0.500
Many of them	2	0.333
Most of them	10	0.091
All of them	$\alpha \rightarrow \infty$	0.001

2.2. Multiple Instance RealBoost

In MIRealBoost algorithm, we define $H^b(X) = \mathbf{sign}\left(F^b(X)\right)$ as the strong classifier of the bag X, where $F^b(X)$ is the real-valued confidence (or score) of X being positive. Given the function $F^b(X)$, the binomial probability of a bag being positive is defined by the logistic function

$$p(X) = \frac{e^{F^b(X)}}{e^{F^b(X)} + e^{-F^b(X)}}. \quad (6)$$

Under this model, the binomial log-likelihood will be

$$\begin{aligned} l\left(Y, p(X)\right) &= \mathbf{1}(Y=1)\log p(X) \\ &\quad + \mathbf{1}(Y=-1)\log\left(1-p(X)\right) && (7) \\ &= -\log\left(1+e^{-2YF^b(X)}\right) && (8) \end{aligned}$$

Our goal is to maximize the expected log-likelihood $El(Y, p(X))$. It is proved in [7] that the maximizer of this function is $p(X) = P(Y=1|X)$ or equivalently:

$$F^b(X) = \frac{1}{2}\log\frac{P(Y=1|X)}{1-P(Y=1|X)}. \qquad (9)$$

In addition, we know that the probability of a bag can be expressed by aggregation of the probability of instances inside the bag:

$$P(Y=1|X) = \mathbf{agg}_{x\in X}(P(y=1|x)). \qquad (10)$$

The aggregation function $\mathbf{agg}$ can be the NOR model in (2) or the OWA operators in (3). On the other hand, if an instance classifier $H(x) = \mathbf{sign}\left(F(x)\right)$ is trained by the original RealBoost algorithm [7], the probability of each instance is given by

$$P(y=1|x) = \frac{e^{F(x)}}{e^{F(x)}+e^{-F(x)}}. \qquad (11)$$

Therefore, if we know the confidence of each instance inside the bag X, we can obtain $F^b(X)$ and classify the bag. In the rest of this section, we try to find $F(x)$.

The confidence function of the RealBoost strong classifier is defined as $F(x) = \sum_{m=1}^{M} f_m(x)$. At each step of the RealBoost algorithm, the weak classifier $f_m(x)$ is obtained by minimizing the stage-wise expected exponential cost:

$$Ee^{-y(F_{m-1}(x)+f_m(x))}. \qquad (12)$$

Setting the derivative w.r.t. $f_m(x)$ to zero, it can be shown that the minimizer is

$$f_m(x) = \frac{1}{2}\log\frac{P_w\left(y=1|x\right)}{P_w\left(y=-1|x\right)}, \qquad (13)$$

where P_w represents the probability distribution of y, given x weighted by $w(x,y) = e^{-yF_{m-1}(x)}$. Using Bayes' rule $P(y|x) \propto P(x|y)P(y)$ with the assumption $P(y=1) = P(y=-1)$, we get

$$f_m(x) = \frac{1}{2}\log\frac{P_w\left(x|y=1\right)}{P_w\left(x|y=-1\right)}. \qquad (14)$$

Algorithm 1 MIRealBoost algorithm

```
Input: Training set = $\{(X_1, Y_1), \cdots, (X_N, Y_N)\}$.
       $X_i = \{x_{i1}, \cdots, x_{i|X_i|}\}, i = 1, \cdots, N$.
       $M$ = number of weak classifiers.
  Initialize the weights $w^p_{ij} = 1/\sum_i(|X_i|)$, the pseudo-labels $y^p_{ij} = Y_i$, and the confidence of each instance $F\left(x_{ij}\right) = 0$.
  for $m = 1 \rightarrow M$ do
      for each available feature $h_k(.), k = 1 \rightarrow K$ do
          Compute weak classifier of each instance w.r.t. each feature $f^k_m(x_{ij}) = \frac{1}{2}\log\frac{\hat{P}_{\{h_k(x_{ij}), y^p_{ij}, w^p_{ij}\}}(h_k(x_{ij})|y=1)}{\hat{P}_{\{h_k(x_{ij}), y^p_{ij}, w^p_{ij}\}}(h_k(x_{ij})|y=-1)}$
          Compute the probability of each instance $p^k(x_{ij}) = \frac{e^{\left(F(x_{ij})+f^k_m(x_{ij})\right)}}{e^{(F(x_{ij})+f^k_m(x_{ij}))}+e^{-(F(x_{ij})+f^k_m(x_{ij}))}}$.
          Compute the probability of each bag $p^k(X_i) = \mathbf{agg}_{x_{ij}}\left(p^k\left(x_{ij}\right)\right)$.
          Compute the empirical log-likelihood $L^k = \sum_i \mathbf{1}(Y_i = 1)\log p^k(X_i) + \mathbf{1}(Y_i = -1)\log\left(1-p^k(X_i)\right)$
      end for
      Set $k^\star = \arg\max_k L^k$.
      Set $F\left(x_{ij}\right) \leftarrow F\left(x_{ij}\right) + f^{k^\star}_m\left(x_{ij}\right)$
      Compute confidence of each bag $F^b(X_i) = \frac{1}{2}\log\frac{p^{k^\star}(X_i)}{1-p^{k^\star}(X_i)}$.
      Update $w^p_{ij} \leftarrow e^{-Y_iF^b(X_i)}$, $i = 1, \cdots, N$, and normalize the weights such that $\sum_{ij} w^p_{ij} = 1$.
  end for
Output: The bag-classifier $\mathbf{sign}\left(F^b(X)\right)$.
```

In practice, the weak classifier is fit by approximating the class probability functions using weighted training instances. In our work, the weighted conditional probability functions for the positive and negative class are estimated by kernel smoothing density functions computed from the weighted voting of training examples. However, we cannot directly use the original training instances x_{ij} to approximate the class probability functions because we do not have the true label of instances inside positive bags. Indeed, we know x_{ij} and $F_{m-1}(x_{ij})$, but we do not know y_{ij}. On the other hand, we know the confidence of each bag (i.e. $F^b_{m-1}(X_i)$) and its label (i.e. Y_i). Consequently, we define new training pseudo-instances $\{x^p_{ij}, y^p_{ij}, w^p_{ij}\}$, where $x^p_{ij} = x_{ij}$, $y^p_{ij} = Y_i$, and $w^p_{ij} = e^{-Y_iF^b_{m-1}(X_i)}$. In fact, we have assumed uniform distribution over the instances of a bag in order to have all the instances compete to take part in prediction of the correct bag label. Thus, we finally get

$$f_m(x) = \frac{1}{2}\log\frac{\hat{P}_{\{x^p_{ij}, y^p_{ij}, w^p_{ij}\}}\left(x|y=1\right)}{\hat{P}_{\{x^p_{ij}, y^p_{ij}, w^p_{ij}\}}\left(x|y=-1\right)}. \qquad (15)$$

$$F_m(x_{ij}) = F_{m-1}(x_{ij}) + f_m(x_{ij}). \qquad (16)$$

Now that we have the confidence of all the instances, we can find the confidence of each bag by (11), (10), (9) and predict the class label of each bag.

The pseudocode of the proposed algorithm is shown in Algorithm 1. In this algorithm, each weak classifier is built from only one feature. Hence, the algorithm sequentially selects the weak classifiers, which maximize the empirical log-likelihood (1), from the pool of all weak classifiers in a stage-wise greedy approach. We found that using redundant features in computation of weak classifiers led to overfitting. Hence, at each iteration we pick the best feature among those which have not been used previously. Our experimental results verify that this approach is resistant to overfitting. In addition, in our experiments we considered each negative instance as a negative bag since there is no ambiguity about the label of the instances in a negative bag. Our investigations showed that using the original negative bags leads to similar results.

3. Experiments

We evaluate MIRealBoost with different aggregation functions on five well-known MIL data sets. These benchmark datasets are the *Elephant*, *Fox*, *Tiger* image retrieval datasets [1] and *Musk1* and *Musk2* drug activity prediction datasets [5]. In the image datasets, each bag represents an image and the instances inside the bag represent 230-D feature vectors of different segmented blobs of the image. The image datasets contain 100 positive and 100 negative bags. In the MUSK datasets, each bag describes a molecule, and the instances inside the bag represent 166-D feature vectors of the low-energy configurations of the molecule. Musk1 has 47 positive bags and 45 negative bags with about 5 instances per bag. Musk2 has 39 positive bags and 63 negative bags with variable number of instances in a bag, ranging from 1 to 1044 (average 64 instances per bag).

The classification accuracies for MIRealBoost with different aggregation functions are shown in Table 2. At each trial, we run the algorithm with 40 iterations (i.e. weak classifiers) for image datasets and 100 iterations for Musk datasets. It can be observed that for the image datasets NOR has the overall best performance. However, for Musk1 and Musk2 the *Many* and *Half* OWA operators outperform the others. The reason might be that in an image usually one of the segments is the true segment (positive instance). However, in the Musk datasets, more than one configuration of a molecule might be positive. In fact, it has been previously reported [8] that Musk1 dataset contains less ambiguity in positive bags, hence there are many positive instances in each bag. MIRealBoost, our algorithm, allows for exploring different aggregators for modeling ambiguity in the bags in order to enhance classification accuracy.

Table 2. MIRealBoost classification accuracy with different aggregation functions. Best methods are marked in bold face

Agg.	Elephant	Fox	Tiger	Musk1	Musk2
NOR	**83**	**63**	72	85	74
Max	77	58	68	85	74
Few	75	58	70	83	72
Some	75	57	**73**	85	75
Half	72	54	70	90	**77**
Many	67	52	67	**91**	75
Most	54	50	51	83	69
All	50	50	50	84	67

Next, we compare MIRealBoost with MILBoost, which is the closest method in feature pool, hypothesis space and structure. Table 3 shows that MIRealBoost algorithm always outperforms MILBoost algorithm. Note that since we run the experiments with the exact training and test sets[1] used in [9], we also report the MILBoost results from this paper.

Table 3. Comparison between MIRealBoost and MILBoost

Method	Elephant	Fox	Tiger	Musk1	Musk2
MIRealBoost	83	63	73	91	77
MILBoost	73	58	56	71	61

Finally, comparison of the best aggregator for MIRealBoost with state-of-the-art MIL methods is provided in Table 4. It can be observed that the performance of the methods varies depending on the dataset. However, MIRealBoost is comparable to the best methods in most cases (Elephant, Fox, and Musk1).

4. Conclusion

We proposed a novel framework for MIL based on boosting that can model different levels of ambiguity in data. Hence, it is more robust to the amount of ambiguity (i.e. true positive instances) in positive bags. To this end, we used OWA operators, which can represent

[1] These sets are available online at `http://www.ymer.org/amir/software/milforests/`

Table 4. Comparison between state-of-the-art MIL methods. Best methods are marked in bold face

Method	Elephant	Fox	Tiger	Musk1	Musk2
MIRealBoost	83	63	73	**91**	77
MIForest [9]	84	**64**	82	85	82
MI-Kernel [1]	84	60	84	88	89
MI-SVM [1]	81	59	84	78	84
mi-SVM [1]	82	58	79	87	84
MILES [4]	81	62	80	88	83
SIL-SVM [3]	85	53	77	88	87
AW-SVM [8]	82	**64**	83	86	84
AL-SVM [8]	79	63	78	86	83
EM-DD [15]	78	56	72	85	85
MIGraph [16]	85	61	82	90	**90**
miGraph [16]	**87**	62	**86**	90	**90**

different degrees of orness in aggregation. Experiments on standard MIL datasets showed that encoding degree of ambiguity in the classifier can influence the accuracy of prediction. MIRealBoost, the proposed algorithm, achieves state-of-the-art results and outperforms the MILBoost algorithm on these datasets.

Acknowledgement

This work was supported by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[2] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[3] R. Bunescu and R. Mooney. Multiple instance learning for sparse positive bags. In *Proceedings of the 24th international conference on Machine learning*, pages 105–112. ACM, 2007.

[4] Y. Chen, J. Bi, and J. Wang. Miles: Multiple-instance learning via embedded instance selection. *T-PAMI*, 28(12):1931–1947, 2006.

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

[6] L. Duan, W. Li, I. Tsang, and D. Xu. Improving web image search by bag-based re-ranking. *Image Processing, IEEE Transactions on*, (99):3280–3290, 2011.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The annals of statistics*, 28(2):337–407, 2000.

[8] P. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *AISTATS*, 2007.

[9] C. Leistner, A. Saffari, and H. Bischof. Miforests: Multiple-instance learning with randomized trees. In *ECCV*, 2010.

[10] W. Li, L. Duan, D. Xu, and I. Tsang. Text-based image retrieval using progressive multi-instance learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2049–2055. IEEE, 2011.

[11] J. Malczewski. Ordered weighted averaging with fuzzy quantifiers: Gis-based multicriteria evaluation for land-use suitability analysis. *Int. Jour. of Applied Earth Observation and Geoinformation*, 8(4):270–277, 2006.

[12] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. In *NIPS*, 1999.

[13] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2006.

[14] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems, Man and Cybernetics*, 18(1):183–190, 1988.

[15] Q. Zhang and S. Goldman. Em-dd: An improved multiple-instance learning technique. In *NIPS*, 2001.

[16] Z. Zhou, Y. Sun, and Y. Li. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1249–1256. ACM, 2009.