Multi-instance Support Vector Machine Based on Convex Combination *

Zhi-Xia Yang[1,2] Naiyang Deng[†3]

[1]College of Mathematics and System Science, Xinjiang University, Urumuchi 830046
[2]Academy of Mathematics and Systems Science, CAS, Beijing 100190
[3]College of Science, China Agricultural University, 100083, Beijing, China

Abstract

This paper presents a new formulation of multi-instance learning as maximum margin problem, which is an extension of the standard C-support vector classification. For linear classification, this extension leads to, instead of a mixed integer quadratic programming, a continuous optimization problem, where the objective function is convex quadratic and the constraints are either linear or bilinear. This optimization problem is solved by an iterative strategy solving a convex quadratic programming and a linear programming alternatively. For non-linear classification, the corresponding iterative strategy is also established, where the kernel is introduced and the related dual problems are solved. The preliminary numerical experiments show that our approach is competitive with the others.

Key words: Multi-instance classification; Support vector machine; Convex combination.

1 Introduction

Multi-instance learning (MIL) is a growing field of research in machine learning. The term multi-instance learning was coined by Dietterich et al.[1] when they were investigating the problem of drug activity prediction. In the MIL problem, the training set is composed of many bags each involves in many instances. A bag is positively labeled if it contains at least one positive instance; otherwise it is labeled as a negative bag. The task is to find some decision function from the training set for correctly labeling unseen bag.

So far, the MIL has been studied in many fields, e.g. in [2, 3, 4]. An increasing number of methods have been developed to solve MIL problems, e.g. [5, 6, 7]. In this paper, inspired by [8, 10], we propose a new formulation of MIL as maximum margin problem, which is an extension of the standard C-support vector classification (C-SVC). For linear classification, this extension leads to, instead of a mixed integer quadratic programming,

*This work is supported by the National Natural Science Foundation of China (No.10801112), the Key Project of the National Natural Science Foundation of China (No.10631070), the China Postdoctoral Science Foundation funded Project (No.20080432573) and the Ph.D Graduate Start Research Foundation of Xinjiang University Funded Project (No.BS080101).

†Corresponding author. E-mail: dengnaiyang@vip.163.com

a continuous optimization problem, where the objective function is convex quadratic and the constraints are either linear or bilinear. This optimization problem is solved by an iterative strategy solving a convex quadratic programming and a linear programming alternatively. For non-linear classification, the corresponding iterative strategy is also established, where the kernel is introduced and the related dual problems are solved. The preliminary numerical experiments show that our approach is competitive with other formulations.

2 Multi-instance SVM

Muliti-instance learning problem: Suppose there is a training set

$$\{(\mathscr{X}_1, y_1), \cdots, (\mathscr{X}_l, y_l)\}, \tag{1}$$

where, when $y_1 = 1$, $\mathscr{X}_i = \{x_{i1}, \cdots, x_{il}\}$ is call as positive bag and $(\mathscr{X}_i, y_i)$ implies that there exists at least one instance with positive label in $\mathscr{X}_i$; when $y_i = -1$, $\mathscr{X}_i = \{x_{i1}, \cdots, x_{il_i}\}$ is called as negative bag and there exists no any instance with positive label in $\mathscr{X}_i$. The task is to find a function $g(x)$ such that the label of any instance in R^n can be deduced by the decision function

$$f(x) = \mathrm{sgn}(g(x)). \tag{2}$$

2.1 Linear Multi-instance SVM

The training set (1) is represented as

$$T = \{(\mathscr{X}_1, y_1), \cdots, (\mathscr{X}_p, y_p), (x_{r+1}, y_{r+1}), \cdots, (x_{r+s}, y_{r+s})\}, \tag{3}$$

where $y_1 = \cdots = y_p = 1$, $y_{r+1} = \cdots = y_{r+q} = -1$, $(\mathscr{X}_i, y_i) = (\mathscr{X}_i, 1)$ implies that there exists at least one instance with positive label in $\mathscr{X}_i = \{x_j | j \in I(i)\}$ and $(x_i, y_i) = (x_i, -1)$ implies that the label of the instance x_i is negative.

Suppose the separating superplane is given by

$$(w \cdot x) + b = 0. \tag{4}$$

Note that it has been pointed out by [10] that the constraint

$$\max_{j \in I(i)} (w \cdot x_j) + b \geq 1 \tag{5}$$

is equivalent to the fact that there exist convex combination coefficients $v^i_j \geq 0$, $\sum\limits_{j \in I(i)} v^i_j = 1$ $\{v^i_j | j \in I(i)\}$, such that

$$(w \cdot \sum_{j \in I(i)} v^i_j x_j) + b \geq 1. \tag{6}$$

Thus, introducing slack variable $\xi = (\xi_1, \cdots, \xi_p, \xi_{r+1}, \cdots, \xi_{r+s})^T$ and the penalty parameters C_1 and C_2, maximum margin principle leads to

$$\min_{w,b,v,\xi} \quad \frac{1}{2}\|w\|^2 + C_1\sum_{i=1}^{p}\xi_i + C_2\sum_{i=r+1}^{r+s}\xi_i, \tag{7}$$

$$\text{s.t.} \quad (w\cdot\sum_{j\in I(i)} v_j^i x_j) + b \geq 1-\xi_i,\ i=1,\cdots,p, \tag{8}$$

$$(w\cdot x_i)+b \leq -1+\xi_i,\ i=r+1,\cdots,r+s, \tag{9}$$

$$\xi_i \geq 0,\ i=1,\cdots,p,r+1,\cdots,r+s, \tag{10}$$

$$v_j^i \geq 0, j\in I(i),\ i=1,\cdots,p, \tag{11}$$

$$\sum_{j\in I(i)} v_j^i = 1,\ i=1,\cdots,p. \tag{12}$$

For solving the problem (7)–(12), we get the following algorithm:

Algorithm 1. (Linear Multi-instance Support Vector Classification)

(1) Given a training set (3);

(2) Select proper penalty parameters $C_1, C_2 > 0$;

(3) Select initial values $\{v_j^i\}$, and denote it as $v_j^i(1)$, e.g select $v_j^i(1) = \frac{1}{|I(i)|}$, $j\in I(i)$, $j=1,\cdots,p$, where $|I(i)|$ stands for the number of the element in the set $I(i)$, i.e. the number of the instances in the positive bag $\mathscr{X}_i$. Set $k=1$;

(4) Compute $w(k)$ from $\{v_j^i(k)\}$: First, construct the series $\bar{x}_1,\cdots,\bar{x}_p,\bar{x}_{r+1},\cdots,\bar{x}_{r+s}$ by

$$\bar{x}_i = \sum_{j\in I(i)} v_j^i x_j,\ i=1,\cdots,p, \tag{13}$$

$$\bar{x}_i = x_i,\ i=r+1,\cdots,r+s, \tag{14}$$

where v_j^i is substituted by $v_j^i(k)$. Then solve the convex quadratic programming

$$\begin{aligned}\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} y_i y_j(\bar{x}_i\cdot\bar{x}_j)\alpha_i\alpha_j + \frac{1}{2}\sum_{i=1}^{p}\sum_{j=r+1}^{r+s} y_i y_j(\bar{x}_i\cdot\bar{x}_j)\alpha_i\alpha_j \\ & +\frac{1}{2}\sum_{i=r+1}^{r+s}\sum_{j=1}^{p} y_i y_j(\bar{x}_i\cdot\bar{x}_j)\alpha_i\alpha_j + \frac{1}{2}\sum_{i=r+1}^{r+s}\sum_{j=r+1}^{r+s} y_i y_j(\bar{x}_i\cdot\bar{x}_j)\alpha_i\alpha_j \\ & -\sum_{i=1}^{p}\alpha_j - \sum_{i=r+1}^{r+s}\alpha_j,\end{aligned} \tag{15}$$

$$\text{s.t.} \quad \sum_{i=1}^{p} y_i\alpha_i + \sum_{i=r+1}^{r+s} y_i\alpha_i = 0, \tag{16}$$

$$0 \leq \alpha_i \leq C_1, i=1,\cdots,p, \tag{17}$$

$$0 \leq \alpha_i \leq C_2, i=r+1,\cdots,r+s, \tag{18}$$

and get the solution $\bar{\alpha} = (\bar{\alpha}_1,\cdots,\bar{\alpha}_p,\bar{\alpha}_{r+1},\cdots,\bar{\alpha}_{r+s})^T$. At last, compute $\bar{w}$ by

$$\bar{w} = \sum_{i=1}^{p}\bar{\alpha}_i y_i\bar{x}_i + \sum_{i=r+1}^{r+s}\bar{\alpha}_i y_i\bar{x}_i, \tag{19}$$

and take it as $w(k)$;

(5) Compute $\{v_j^i(k+1)\}$ from $w(k)$: Solve the linear programming with the variable $\{v_j^i\}$, b and ξ obtained from problem (7)–(12) by taking $w = w(k)$. Take its solution $\{v_j^i\}$ with respect to v as $\{v_j^i(k+1)\}$;

(6) Compare $\{v_j^i(k+1)\}$ to $\{v_j^i(k)\}$. When their difference is small enough, construct the decision function

$$f(x) = \mathrm{sgn}((w^* \cdot x) + b^*), \tag{20}$$

where $w^* = w(k)$ and b^* is the just obtained solution $\bar{b}$ with respect to b to the linear programming in step (5). And stop; Otherwise, set $k = k+1$, go to step (4).

2.2 Nonlinear Multi-instance SVM

Using a kernel function $K(x, x')$, the general multi-instance support vector classification is obtained by extending the linear multi-instance support vector classification. In fact, in order to arrive nonlinear separation, introduce the map

$$\Phi: \quad \begin{array}{c} R^n \to \mathscr{H}, \\ x \to \mathrm{x} = \Phi(x). \end{array} \tag{21}$$

Let the separating superplane in the Hilbert space $\mathscr{H}$ be

$$(\mathrm{w} \cdot \mathrm{x}) + b = 0. \tag{22}$$

For a positive bag $\mathscr{X}_i$, the convex combination $\sum\limits_{j \in I(i)} v_j^i \Phi(x_j)$ of the images $\{\Phi(x_j) | j \in I(i)\}$ is considered. Thus, corresponding to the problem (7)–(12), we should solve the optimization problem

$$\min_{\mathrm{w}, b, v, \xi} \quad \frac{1}{2}\|\mathrm{w}\|^2 + C_1 \sum_{i=1}^{p} \xi_i + C_2 \sum_{i=r+1}^{r+s} \xi_i, \tag{23}$$

$$\text{s.t.} \quad (\mathrm{w} \cdot \sum_{j \in I(i)} v_j^i \Phi(x_j)) + b \geq 1 - \xi_i, \; i = 1, \cdots, p, \tag{24}$$

$$(\mathrm{w} \cdot \Phi(x_i)) + b \leq -1 + \xi_i, \; i = r+1, \cdots, r+s, \tag{25}$$

$$\xi_i \geq 0, \; i = 1, \cdots, p, r+1, \cdots, r+s, \tag{26}$$

$$v_j^i \geq 0, j \in I(i), \; i = 1, \cdots, p, \tag{27}$$

$$\sum_{j \in I(i)} v_j^i = 1, \; i = 1, \cdots, p. \tag{28}$$

For solving the above problem, we arrive the following algorithm.

Algorithm 2. (Multi-instance Support Vector Classification)

(1) Given a training set (3);

(2) Select proper kernel function $K(x, x')$ and panelty parameter $C_1, C_2 > 0$;

(3) Select initial values $\{v_j^i\}$, and denote it as v_j^i, e.g select $v_j^i = \frac{1}{|I(i)|}$, $j \in I(i)$, $j = 1, \cdots, p$, where $|I(i)|$ stands for the number of the elements in the set $I(i)$, i.e. the number of the instances in the positive bag $\mathscr{X}_i$;

(4) Compute $\bar{\alpha}$ from $\{v_j^i\}$: Solve the convex quadratic programming

$$\begin{aligned}
\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} y_i y_j \alpha_i \alpha_j K(x_i,x_j) + \frac{1}{2}\sum_{i=1}^{p}\sum_{j=r+1}^{r+s} y_i y_j \alpha_i \alpha_j (\sum_{k\in I(j)} v_k^j K(x_i,x_k)) \\
& + \frac{1}{2}\sum_{i=r+1}^{r+s}\sum_{j=1}^{p} y_i y_j \alpha_i \alpha_j (\sum_{k\in I(i)} v_k^i K(x_k,x_j)) \\
& + \frac{1}{2}\sum_{i=r+1}^{r+s}\sum_{j=r+1}^{r+s} y_i y_j \alpha_i \alpha_j (\sum_{k\in I(i)} v_k^i \sum_{l\in I(j)} v_l^j K(x_k,x_l)) \\
& - \sum_{i=1}^{p}\alpha_i - \sum_{i=r+1}^{r+s}\alpha_i, \qquad (29) \\
\text{s.t.} \quad & \sum_{i=1}^{p} y_i\alpha_i + \sum_{i=r+1}^{r+s} y_i\alpha_i = 0, \qquad (30) \\
& 0 \le \alpha_i \le C_1,\ i = 1,\cdots,p, \qquad (31) \\
& 0 \le \alpha_i \le C_2,\ i = r+1,\cdots,r+s, \qquad (32)
\end{aligned}$$

and get the solution $\bar{\alpha} = (\bar{\alpha}_1,\cdots,\bar{\alpha}_p,\bar{\alpha}_{r+1},\cdots,\bar{\alpha}_{r+s})^T$, For all i,j, set $\{\tilde{v}_j^i\} = \{v_j^i\}$;

(5) Compute $\{\bar{v}_j^i\}$ from $\bar{\alpha},\tilde{v}$: Solve the linear programming obtained from problem (23)–(28) with substituting w by

$$\bar{\mathrm{w}} \ = \ \sum_{i=1}^{p}\bar{\alpha}_i y_i \Phi(x_i) + \sum_{i=r+1}^{r+s}\bar{\alpha}_i y_i (\sum_{j\in I(i)} \tilde{v}_j^i \Phi(x_j)), \qquad (33)$$

and get its solution $\{\bar{v}_j^i\}$ with respect to $\{v_j^i\}$;

(6) Compare $\{\bar{v}_j^i\}$ to $\{\tilde{v}_j^i\}$. When their difference is small enough , construct the decision function

$$f(x) = \mathrm{sgn}(g(x)), \qquad (34)$$

where $g(x)$ is obtained by

$$g(x) = \sum_{i=1}^{p}\bar{\alpha}_i y_i K(x_i,x) + \sum_{i=r+1}^{r+s}\bar{\alpha}_i y_i (\sum_{j\in I(i)} v_j^i K(x_j,x)) + \bar{b}. \qquad (35)$$

and either

$$\bar{b} = y_j - \sum_{i=1}^{p} y_i\bar{\alpha}_i K(x_i,x_j) - \sum_{i=r+1}^{r+s} y_i\bar{\alpha}_i (\sum_{l\in I(i)} v_l^i K(x_l,x_j)); \qquad (36)$$

or

$$\bar{b} = y_j - \sum_{i=1}^{p} y_i\bar{\alpha}_i (\sum_{k\in I(j)} v_k^j K(x_i,x_k)) - \sum_{i=r+1}^{r+s} y_i\bar{\alpha}_i (\sum_{l\in I(i)} v_l^i \sum_{k\in I(j)} v_k^j K(x_l,x_k)). \qquad (37)$$

Here $\bar{\alpha}$ and $\{v_j^i\}$ are the latest corresponding values; Otherwise set $\{v_j^i\} = \{\bar{v}_j^i\}$, go to step (4).

3 Numerical Experiment

To estimate the capabilities of our proposed methods, we report the results on four public benchmark datasets, one from the UCI machine learning repository [11], and three from [8]. The datasets from [8] are used to evaluate our linear multi-instance support vector classification, where two datasets, “Elephant" and “Tiger", are from an image annotation task in which the goal is to determine whether or not a given animal is present in an image; the rest one dataset, “TST1", is from the OHSUMED data and the task is to learn binary concepts associated with the Medical Subject Headings of MEDLINE documents. The “Musk1" dataset from the UCI machine learning repository [11] is used to test our nonlinear multi-instance support vector classification, which involves bags of molecules and their activity levels and is commonly used in multi-instance classification. Detailed information about these datasets can be found in [10].

We use ‘linprog’ function in MATLAB to solve the linear programming in the two algorithms. For the quadratic programming, we use SVM*light*[12] and ‘quadprog’ function with MATLAB in Algorithm 1 and Algorithm 2 respectively. The testing accuracies for our method are calculated using standard 10-fold cross validation method[13]. The parameters, the regularization parameter C and the RBF kernel parameter γ, are selected from the set $\{2^i|i=-8,\cdots,8\}$ by 10-fold cross validation on the tuning set comprising of random 10% of the training data. Once the parameters are selected, the tuning set was returned to the training set to learn the final classifier. In Algorithm 1, the regularization parameters are set up $C_1=C_2=C$. In Algorithm 2, $C_1=\frac{n_{\mathrm{N-instance}}}{n_{\mathrm{P-bag}}+n_{\mathrm{N-instance}}}C$ and $C_2=\frac{n_{\mathrm{P-bag}}}{n_{\mathrm{P-bag}}+n_{\mathrm{N-instance}}}C$, where $n_{\mathrm{N-instance}}$ is the number of instances in all negative bags, $n_{\mathrm{P-bag}}$ is the number of all positive bags. Our algorithms are stopped if the difference between the convex combination coefficients v is less than $10^(-4)$ or if the iterations $k>80$.

We compare our results with MICA[10], SVM1[10], mi-SVM[8],MI-SVM[8],R-SVM[9] and EM-DD[5]. Our methods are denoted as SVM-CC since they are based on convex combination. The results of 10-fold cross validation accuracy are listed in Table 1. Correctness results for mi-SVM, MI-SVM and EM-DD are taken from [8] and for R-SVM from [9]. We see from Table 1 that our method has the best correctness on the ‘Musk1’ dataset. Although our method does not achieve the best accuracy results in the rest three datasets, they are comparable second bests in Table 1.

Table 1: Results for four models

Dataset	MICA	mi-SVM	MI-SVM	R-SVM	EM-DD	SVM-CC
Elephant	80.5%	82.2%	81.4%	**84.3%**	78.3%	81.5%
Tiger	82.6%	78.4%	**84%**	83.7%	72.1%	83%
TST1	94.5%	93.6%	93.9%	**95.1%**	85.8%	95%
Musk1	84.4%	87.4%	77.9%	85.8%	84.8%	**88.9%**

4 Conclusion

In this paper, we develop a novel approach to multi-instance learning problem based on convex combination for the instances in each positive bag and establish two algorithms for linear multi-instance SVC and nonlinear multi-instance SVC respectively. The key point is that the multi-instance learning is formulated, as an extension of the standard C-support vector classification, as a nonlinear optimization problem with continuous variables. This problem, in principle, is easy to be solved since it does not involved in integer variables like MI-SVM. It is interesting to solve this problem by a more reasonable and efficient way.

References

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

[2] T. Brow, B. Settles and M. Craven. Classifying biomedical articles by making localized decisions. In Proc. of the Fourteenth Text Retrieval Conference (TREC),2005.

[3] G. Carneiro, A. B. Chan, P. J. Moreno and N. Vas concelos. Supervised on pattern analysis and machine intelligence, 29(3):394–410,2007.

[4] D. R. Dooly, Q. Zhang, A. Goldman and R. A. Amar. Multiple instance learning of real-valued data. Journal of Machine Learning Research, 3, 651–678, 2002.

[5] Q. Zhang and S.Goldman. EM-DD: an improved multiple instance learning technique. In Neural Information Processing Systems,14,2001.

[6] J. Wang and J. D. Zucher. Solving the multiple-instance problem: a lazy learning approach. In Proceedings of the 17th International conference on Machine Learning, San Francisco, CA, 1119–1125, 2000.

[7] G. Ruffo. Learning single and multiple instance decision tree for computer security applications. PhD dissertation, Department of Computer Science, University of Turin, Torino, Italy, Feb. 2000.

[8] S. Andrews, I. Tsochantaridis, and T Hofmann. Support vector machines for multiple-instance learning. Advances in Neural Information Processing Systems 15, MIT Press, 561–568, 2003.

[9] O. Seref and O.E. Kundakcioglu. Multiple instance classification with relaxed support vector machines. Proceedings of the 3rd INFORMS Workshop on Data Mining and Health Informatics (DM-HI), 2008.

[10] O. L. Mangasarian, and E. W. Wild. Multiple instance classification via successive linear programming. Journal of Optimization Theory and Application, 137(1): 555–568, 2008.

[11] P. M. Murphy and D. W. Aha. UCI machine learning repository. www.ics.uci.edu/ mlearn/MLRepository.html,1992.

[12] T. Joachims. Making large-scale spupport vector machine learning practical. Advances in Kernel Methods—Support Vector Learning, B. Schölkopf,C.J.C. Burges, and A.J.Smola,eds,pp.169–184, Cambridge,Mass:MIT Press,1999.

[13] T.M. Mitchell. Machine learning. McGraw-Hill International, Singapore,1997.