Abstract In this paper, we formulated a novel method to solve the classification problem within the multiple instance learning (MIL) contexts by multiple kernel learning. Despite the large number of SVM models, there are only a few models that can solve general MIL problems well. To improve the classification precision of SVM method with regard to MIL problem, this paper introduced multiple kernel learning method to the process of multiple instance learning, and proposed a new SVW model (MKMI-SVM), which based on the MI-SVM model. The solution for this model was presented, and some numerical experiments on benchmark data were taken into this paper too. Computational results on a number of datasets indicate that the proposed algorithm is competitive with other SVM methods.

Keywords Multiple instance learning, Support vector machines, multiple kernel learning, multi-class SVM

1 Introduction

Literatures [1–3] gave us an introduction about multiple instance classification problems. In this paper we gave a method to solve the problem which is mentioned in literature [10]. The problem to consist of classifying positive and negative bags of points in the n-dimensional real space $\mathbb{R}^n$ on the following basis is considered. A bag is classified as a positive bag if one or more instances in that bag are positive, otherwise it is classified as a negative bag.

This problem was firstly analyzed by T. G. Dietterich et al. in the pharmed activity’s prediction in the 90s, 200th century. T. G. Dietterich considered every molecule as a bag in their analysis, and every low power shape represented an instance in the bag. This is the origin of multiple instance learning (MIL). The MIL problem has been existed for a long while; however, it is not a sudden result of pharmed activity’s prediction. Previous machine learning didn’t take this kind of problems’ property into consideration formally, and the problem hasn’t been exactly

---

* This work is supported by the Key Project of National Natural Science Foundation of China (No.10631070), the National Natural Science Foundation of China (No.10971223), the Chinese Universities Scientific Fund(No.2009-2-05), and the Graduate Basic Research Foundation of China Agricultural University Funded Project (No.15059208).

† Corresponding author. E-mail address: jingling_student@163.com.
defined until T. G. Dietterich’s work.

Later the analysis of this problem aroused a number of Machine learning researchers’ interest and a lot of research works have been done. Various methods for multiple instance classification problems have been proposed, including integer programming [6], expectation maximization [7], kernel formulations [8], and lazy learning [7]. Ray and Craven [9] provide an empirical comparison of several multiple instance classification algorithms and their non-multiple-instance counterparts. The classical SVM methods to solve MIL problem are MI-SVM and mi-SVM, which proposed by S. Andrews [6]. Based on their work, this paper added multiple kernel methods to the classical SVM methods, and gave a novel formulation for MIL problem. Meanwhile the strengths and the weakness about this new method have been discussed. Andrews et al. extend the single kernel SVM, while we begin with the multiple kernels SVM [10]. The use of the multiple kernels SVM allows us to get a better description of data’s distributing as opposed to single kernel SVM. We include results in Sect. 4 which demonstrate that multiple kernel methods are much more computationally efficient and faster than classical methods.

The paper is organized as follows. In Sect. 2 we give a review of some interrelated concepts. In Sect. 3 we introduce our formulation of the multiple instance classification problems and state its properties. In Sect. 4 we present our numerical tests on five datasets. Section 5 concludes the paper.

2 The background about SVM method for multiple instance learning and multiple kernel learning

As the background of this paper, this section will introduce MI-SVM which is the classical SVM method for MIL problem and the standard multiple kernel SVM method.

2.1 Support vector machines for multiple instance learning

In this part, a brief review about classical SVM methods for multiple instance learning, MI-SVM and mi-SVM, will be shown. And we are going to introduce the basic idea, model and solving algorithm respectively.

Andrews et al. [6] have previously investigated extending support vector machines to the multiple instance classification problems using mixed-integer programming. They use integer variables either to select the class of points in positive bags or to identify one point in each positive bag as a “witness” point that must be placed on the positive side of the decision boundary. Each of these representations leads to a natural heuristic for approximately solving the resulting mixed-integer program.

S. Andrews gave an alternative way [6] of applying maximum margin ideas which is the main ideas of SVM to the MIL setting. They extend the notion of a margin from individual patterns to set of patterns. It is natural to define the functional margin of a bag with respect to a hyperplane by \( \gamma_I = \max_{i=1}^{n_I} \langle w, x_i \rangle + b \). Therefore based on the this notion of a bag margin, the SVM model has been redefined into
\[
\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum \xi_i
\]
\[
\text{MI-SVM} \quad \text{s.t.} \quad \forall i \in I \quad Y_i \max_i \langle w, x_i \rangle + b \geq 1 - \xi_i
\]
\[
\xi_i \geq 0
\]

For solving this program, unfolding the max operations by introducing one inequality constraint for per instance has been done. For negative bags, the inequality constraints can be read as \(-\langle w, x_i \rangle - b \geq 1 - \xi_i, \forall i \in I\), where \(Y_i = -1\).

For positive bags, [6] introduces a selector variable
\[
s(I) \in I \quad \text{which denotes the instance selected as the positive “witness” in per positive bag } B_s. \quad \text{Meanwhile they gave two methods to select } s(I) \in I \text{ from } B_s, \text{ MI-SVM and mi-SVM.}
\]

Both of MI-SVM and mi-SVM are cast as mixed-integer programs. They will be shown in Algorithm 1 and Algorithm 2 in the following, respectively.

**Algorithm 1: MI-SVM algorithm**

| Initialize: \( s, Y \) for \( i \in I \) |
| REPEAT Compute SVM solution \( w, b \) for data with imputed labels |
| Compute outputs \( f_i(x) = \langle w, x_i \rangle + b \) for all \( x_i \) in positive bags |
| Set \( y_i = \text{sign}(f_i) \) for every \( i \), \( Y_i = 1 \) |
| FOR (every positive bag \( B_s \)) |
| IF \( \sum \langle w, x_i \rangle + b \geq 0 \) |
| Compute \( i = \arg \max_{i} f_i \) |
| Set \( y_i = 1 \) |
| END |
| END |
| WHILE (imputed labels have changed) |
| OUTPUT \( (w, b) \) |

**Algorithm 2: mi-SVM algorithm**

| Initialize: \( X, Y \) for all \( X, Y \) |
| REPEAT Compute QP solution \( w, b \) for data set with positive examples { \( X, Y \) : \( Y = 1 \) } |
| Compute outputs \( f_i(x) = \langle w, x_i \rangle + b \) for all \( x_i \) in positive bags |
| Set \( X_s, X_{s+} = \arg \max_{i} f_i \) for every \( i, Y_i = 1 \) |
| WHILE (selector variables \( s(I) \) have changed) |
| OUTPUT \( (w, b) \) |

### 2.2 Multiple kernel learning

MI-SVM model by Andrews et al. extend the classical simple kernel SVM, while we begin with the multiple kernels SVM [11]. In this part, we will give an introduction about multiple kernels SVM’s basic idea, model and main computing Algorithm.

Multiple kernels learning (MKL) aims at simultaneously learning a kernel and the associated predictor in supervised learning settings. Let \( \{x_i, y_i\}_{i=1}^n \) is the learning set, where \( x_i \) belongs to some input space \( X \) and \( y_i \) is the target value for pattern \( x_i \). For kernel algorithms in SVM, the solution of the learning problem is of the form

\[
f(x) = \sum_{i=1}^n a_i^* K(x, x_i) + b^*
\]

where \( a^* \) and \( b^* \) are some coefficients to be learned from examples, while \( K(\cdot, \cdot) \) is a given positive definite kernel associated with a reproducing kernel Hilbert space (RKHS) \( H \).

In some situations, a learning practitioner may be interested in more flexible
models. Recent applications have shown that using multiple kernels instead of a single one can enhance the interpretability of the decision function and improve performances \cite{11}. In such cases, a convenient approach is to consider that the kernel $K(x,x')$ is actually a convex combination of basis kernels:

$$K(x,x') = \sum_{m=1}^{M} d_m K_m(x,x'),$$

with $d_m \geq 0$, $\sum_{m=1}^{M} d_m = 1$

where $M$ is the total number of kernels. Each basis kernel $K_m$ may either use the full set of variables describing $x$ or subsets of variables stemming from different data sources \cite{11}. Alternatively, the kernels $K_m$ can simply be classical kernels (such as Gaussian kernels) with different parameters. Within this framework, the problem of data representation through the kernel is then transferred to the choice of weights $d_m$.

In the SVM methodology, the decision function is of the form $f(x) = \sum_{i=1}^{l} \alpha_i y_i \sum_{m=1}^{M} d_m K_m + b$, where the optimal parameters $d_m$, $\alpha_i$ and $b$ are obtained by solving the dual of the following optimization problem \cite{10}:

$$\min_{(d_m,b) \in \mathbb{R}^k} \frac{1}{2} \sum_{m=1}^{M} d_m^2 + C \sum_{i=1}^{l} \xi_i,$$

subject to $y_i \sum_{m=1}^{M} d_m K_m + b \geq 1 - \xi_i \quad \forall i$

$\xi_i \geq 0 \quad \forall i$

$\sum_{m=1}^{M} d_m = 1, \quad d_m \geq 0 \quad \forall m$

The MKL formulation introduced by Bach et al. \cite{12} and further developed by Sonnenburg et al. \cite{13} consists in solving an optimization problem expressed above. Nowadays an effective method to solve this optimization problem is proposed by Alain Rakotomamonjy et al. in 2008 \cite{11}. The main algorithm will be shown in algorithm 3 in following.

**Algorithm 3** Simple MKL algorithm

1. Set $d_m = \frac{1}{M}$ for $m = 1, \ldots, M$
2. While stopping criterion not met do
   3. Compute $J(d)$ by using an SVM solver with $K = \sum_{m} d_m K_m$
   4. Compute $\frac{\partial J}{\partial d_m}$ for $m = 1, \ldots, M$ and descent direction $D(2)$.
   5. Set $\mu = \arg \max_{a \in [0,1]}$.
   6. While $\mu \in (0,1)$ do
      7. Compute $\frac{\partial J}{\partial d_m}$.
      8. $\gamma = \mu d_m$
      9. $d_m = \alpha d_m + (1 - \alpha) d_m$
      10. $D' = D + \gamma D$
      11. $D'' = D' + D$
      12. $D_0 = D''$
      13. Compute $J'$ by using an SVM solver with $K = \sum_{m} d_m K_m$
   14. $d_m = d_m - d_m$
   15. End while
3. Line search along $D$ for $\gamma \in [0,1]$, [calls an SVM solver for each $\gamma$ trial value]
4. $d = d + \gamma D$
5. End while
3 MKMI-SVM Classification Model and Algorithm

Multiple kernel SVM is used for some situations where a machine learning practitioner may be interested in more flexible models. We can expect multiple kernel learning will have a better performance in MIL problem for two reasons. Obviously, since the highly complicated description about real object in MIL that the special problem, a flexible model is necessary for the learning task. Meanwhile the enhance about the interpretability of the decision function, more effective computation and higher predication accuracy not only can be expected, but also are our hopes in MIL. Therefore, it is significant to add multiple kernel method to MIL problem. In this section, the model and algorithm of MKMI-SVM will be given.

In MKMI-SVM method, we also defined the functional margin of a bag with respect to a hyperplane by \( \gamma_i = Y_i \max \langle w, x_i \rangle + b \). Based on this rules, the inequality constraints in multiple kernel SVM can be changed for solving MIL problem. Therefore the MKMI-SVM model can be expressed a new optimization problem showed in following:

\[
\min_{\{f_m, b, d_m\}} \frac{1}{2} \sum_{m} \sum_{\alpha} \| f_m \|_{D_m}^2 + C \sum_{\gamma} \xi_{\gamma} \\
\text{s.t.} \quad Y_i \max \left( \sum_{m} f_m(x_i) + b \right) \geq 1 - \xi_{\gamma}, \quad \forall I \\
\xi_{\gamma} \geq 0 \\
\sum_{m} d_m = 1, \quad d_m \geq 0 \quad \forall m
\]

Since the first constraint in our multiple instance formulation contains the max operations. We also unfolded this max operation as [6]. For negative bags, the inequality constraint can be read as \(- \langle w, x_i \rangle - b \geq 1 - \xi_{\gamma}, \forall i \in I\), where \( Y_i = -1 \). For positive bags, a selector variable \( s(I) \in I \) which denotes the instance selected as the positive instance in per positive bag \( I \), will be gave. For \( d_m \), \( s(I) \in I \) and \( \alpha, b \), alternately compute one set of variables when hold other sets. This leads to the successive solution of MKMI-SVM programs that underly our algorithm which we specify now.

---

**Algorithm 4** MKMI-SVM Algorithm

---

---

---

---
In practice, computing the MKMI-SVM Algorithm 4 may be faster than classical MI-SVM when you should change your kernel to check more kernel Hilbert space (RKHS). If the number of kernel you should compute is N, the classical SVM methods will compute their Algorithm 5 times. And the MKMI-SVM time is equal to its number of iterations time. In [11], a lot of numerical testing had been done to compare which is faster. Multiple kernel learning often has the better performances.

4 Numerical Experiments

In this section, some numerical experiments will be done for testing the MKMI-SVM’s capabilities in MIL problem. To evaluate the capabilities of MKMI-SVM method, we have performed some experiments on benchmark data. In this paper, we reported results on 5 datasets, two from the UCI machine learning repository [14], and three from [6]. Detailed information about these datasets is summarized in Table 1. We use the datasets from [6] to evaluate our multiple kernel classification algorithms. These three datasets 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 two datasets from the UCI repository [14] are the Musk datasets, which are commonly used in multiple instance classification.

Table 1 Description of the datasets used in the experiments. Elephant, Fox and Tiger datasets are used in [6], while Musk-1 and Musk-2 are available from [14]. +Bags denotes the number of positive bags in each dataset, while +Instances denotes the total number of instances in all the positive bags. Similarly, −Bags and −Instances denote corresponding quantities for the negative bags.

<table>
<thead>
<tr>
<th>Data set</th>
<th>+bag</th>
<th>+instances</th>
<th>−bag</th>
<th>−instances</th>
<th>features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elephant</td>
<td>100</td>
<td>762</td>
<td>100</td>
<td>629</td>
<td>143</td>
</tr>
<tr>
<td>Fox</td>
<td>100</td>
<td>647</td>
<td>100</td>
<td>673</td>
<td>143</td>
</tr>
<tr>
<td>Tiger</td>
<td>100</td>
<td>544</td>
<td>100</td>
<td>676</td>
<td>143</td>
</tr>
<tr>
<td>Musk-1</td>
<td>47</td>
<td>207</td>
<td>45</td>
<td>269</td>
<td>166</td>
</tr>
<tr>
<td>Musk-2</td>
<td>39</td>
<td>1017</td>
<td>63</td>
<td>5581</td>
<td>166</td>
</tr>
</tbody>
</table>

We compare our multiple kernels classification algorithm to the mi-SVM and MI-SVM [60] on three image datasets. Since Andrews et al. also report results on Zhang and Goldman’s expectation maximization approach EM-DD [7] on these datasets [6], we include those results here as well. Table 2 reports results comparing MKMI-SVM to mi-SVM, MI-SVM and EM-DD. Accuracy results for mi-SVM, MI-SVM and EM-DD were taken from [6]. Accuracy for MIMK-SVM was measured by averaging ten ten-fold cross validation runs. The multiple kernels for MKMI-SVM were selected by 10 different Gaussian kernel, kernel parameters form $2^{-5}$ to $2^5$. The parameters C for MKMI-SVM were selected from the set $\{2^i | i = -5, \ldots, 4\}$ by ten-fold cross validation on each training samples of the image datasets.
Table 2 MKMI-SVM, mi-SVM [6], MI-SVM [6] and EM-DD [7] testing accuracy used averaged over ten ten-fold cross validation experiments. The datasets are those used by Andrews et al. in [6]. Best accuracy on each dataset is in bold.

<table>
<thead>
<tr>
<th>Data set</th>
<th>MKMI-SVM</th>
<th>mi-SVM</th>
<th>MI-SVM</th>
<th>EM-DD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elephant</td>
<td>81.8%</td>
<td>82.2%</td>
<td>81.4%</td>
<td>78.3%</td>
</tr>
<tr>
<td>Fox</td>
<td>58.7%</td>
<td>58.2%</td>
<td>57.8%</td>
<td>56.1%</td>
</tr>
<tr>
<td>Tiger</td>
<td>84.0%</td>
<td>78.4%</td>
<td>84.0%</td>
<td>72.1%</td>
</tr>
</tbody>
</table>

In order to evaluate the difference between the algorithms more precisely, we used the Friedman test [17] on the results reported in Table 2. The Friedman test is a nonparametric test that compares the average ranks of the algorithms, where the algorithm with the highest accuracy on a dataset is given a rank of 1 on that dataset, and the algorithm with the worst accuracy is given a rank of 5. Therefore the average rank was 1.3 for MKMI-SVM, 1.5 for mi-SVM, 2.3 for MI-SVM, and 4 for EM-DD. The better performance by MKMI-SVM expressed on MKL problem was clearly showed.

Table 3 MKMI-SVM, mi-SVM [6], MI-SVM [6], EM-DD [7], DD [15], MI-NN [16], IAPR [11], and MIK [8] ten-fold testing accuracy on the Musk-1 and Musk-2 datasets. Best accuracy is in bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MKMI-SVM</th>
<th>mi-SVM</th>
<th>MI-SVM</th>
<th>EM-DD</th>
<th>DD</th>
<th>MI-NN</th>
<th>IAPR</th>
<th>MIK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musk-1</td>
<td>88.6%</td>
<td>87.4%</td>
<td>77.9%</td>
<td>84.8%</td>
<td>88.0%</td>
<td>88.9%</td>
<td>92.4%</td>
<td>91.6%</td>
</tr>
<tr>
<td>Musk-2</td>
<td>85.2%</td>
<td>83.6%</td>
<td>84.3%</td>
<td>84.9%</td>
<td>84.0%</td>
<td>82.5%</td>
<td>89.2%</td>
<td>88.0%</td>
</tr>
</tbody>
</table>

Table 3 gives ten-fold cross validation accuracy results for MKMI-SVM using the same test method on the Musk-1 and Musk-2 datasets which are available from the UCI repository [14]. In table 3, we can see that MKMI-SVM got the best accuracy in all SVM methods, but some simple method like IAPR methods, got the better result on the contrary. It showed that MKMI-SVM will just have a better performance on the complicated dataset of which the acting feature and its reciprocity in classification is not very clear, but for the ordinary datasets of which the acting feature and its reciprocity is clearly enough, it is proper to perform some simple methods. Obviously, this is in line with the principle of Occam's razor; meanwhile it can suggest us the type of dataset for which MKMI-SVM method is proper to be used.

5 Conclusion and Outlook

This paper has introduced a mathematical programming formulation of the multiple instance problems that has used multiple kernel learning. Results on previously published datasets indicate that our approach is effective at some situation where a machine learning practitioner may be interested in more flexible models. Furthermore, multiple kernels learning often cost less than simple kernel for learning in multiple kernel Hilbert space, and computing the MKMI-SVM maybe faster than classical simple kernel method in practice. Improvements in the mathematical
programming formulation and evaluation using a wide variety of datasets and algorithms, such as those in [17], are promising avenues of future research.

References


