Joint Kernel Learning and Multi-Instance Classification

Hossein Hajimirsadeghi
School of Computing Science
Simon Fraser University
hosseinh@sfu.ca

Greg Mori
School of Computing Science
Simon Fraser University
mori@cs.sfu.ca

Abstract

This paper presents a structured kernel learning approach to multi-instance classification, where the bags are modeled as hidden conditional random fields (hCRFs) with cardinality potentials. A kernel, parameterized by the corresponding hCRF cardinality model, is jointly trained with the max-margin classifier in a regularized risk optimization problem. The integrated learning forms the instance-level labeling and the bag-level discriminative classification together, with a direct goal of improved bag classification The proposed method is evaluated on standard multi-instance learning data sets.

1 Introduction

Multiple instance learning (MIL) approaches are powerful because they can leverage a bag of instances to make a decision. They can handle weakly labeled data and aggregate information from a set of related instances to make a classification decision. However, squeezing the available utility from a bag of instances requires two key components. First, modeling the varying relationship between instance-level labels and a bag label is crucial. Second, holistic information about an entire bag should be captured. In this paper we propose a principled, unified framework for capturing these two sources of information. We present a method for learning powerful kernels that operate over graphical models describing bags and their counts of instance-level labels. These learned kernels can adapt to a particular problem, determining the right combination of instance-level analysis, aggregation, and bag-level representation appropriate for classification.

Multi-instance classification algorithms can be categorized into instance-level and bag-level methods based on the space or level in which the discriminative information lies [19, 1]. In the instance-level paradigm, an instance classifier is trained to classify positive and negative instances, and based on the instance-level predictions a bag classifier is obtained. Thus in these methods, the mechanism which determines how the instances contribute to bag prediction is important. This mechanism is usually designed based on multi-instance assumptions. The standard assumption (i.e., a bag is positive if at least one of the instances is positive) was proposed for the early applications of MIL. However, this assumption discards information useful in many MIL applications. For example, in image retrieval most top-ranked training images are truly relevant to the query, i.e., they are true positives and not just additional irrelevant elements in a bag [20]. Stronger classifiers can be trained by employing better assumptions that extract more information from a bag. Recently, generalized assumptions have been studied, such as *ratio-based* assumptions [11, 20, 9, 24, 12], where the ratio of positive instances in a bag determines the bag label. According to [12, 13, 11] encoding the level of ambiguity of instance labels (e.g., the portion of positive instances in a bag) in the classifier can significantly improve the accuracy of classification. However, instance-level methods lack the ability to extract holistic information from the bag.

On the other hand, in the bag-level paradigm, a classifier is trained directly on bags by extracting discriminative global information from the whole bag. For example, each bag is mapped to a single feature vector, which summarizes the information about a whole bag, and a standard single-instance learner is used to classify the bags in the resulting vectorial embedding space. The bag-level approach can be more effective for bag classification [1]. By defining appropriate kernel, distance or mapping functions, the bag-level methods extract unified bag metadata, which can improve classification accuracy. Most bag-level methods find a summary of the bag without considering the instance labels. This limits the ability of a method to discern and model contributions of individual instances to classification.

The Cardinality Kernel proposed in [15] introduced a novel framework which has the advantages of both classes of methods: 1) encoding the level of ambiguity of instance labels with ratio-based assumptions and 2) defining a kernel which can extract discriminative bag-level information from data. In this framework, first a bag is modeled as a hidden conditional random field (hCRF), namely Cardinality Model [12, 14], to capture the count-based relations between hidden instance labels. The parameters of the Cardinality model can be learned by either likelihood maximization or max-margin optimization. Next, a kernel is defined on bags based on the trained Cardinality Model for the purpose of bag classification. The proposed kernel can give a holistic discriminative model of weakly annotated graph-structured data in an embedded space of even infinite dimensionality, and simply be incorporated in a variety of machine learning techniques by using the kernel trick.

This paper extends our previous works on Cardinality Models [12, 14] and Cardinality Kernels [15], and proposes a unified algorithm for joint kernel learning and classifier training. The parameters of the Cardinality Kernel (which are indeed the hCRF parameters) are learned inside a regularized SVM optimization, and consequently the kernel can faithfully adapt to the target multi-instance classification task.

This paper is organized as follows. Section 2 reviews related work. Section 3 provides some preliminaries to introduce the topic, notation, and base model. In Section 4 our framework is described, including the proposed kernel and the kernel learning algorithm. The algorithms are also analyzed in terms of computational complexity. Section 5 shows the results of evaluating the proposed methods and summarizes the experimental comparisons to the state-of-the-art MIC algorithms. We conclude in Section 6.

2 Related Work

Gärtner et al. [10] defined a class of kernels for multi-instance classification. In these kernels all instances of a bag contribute to bag classification equally, although they are not equally important in practice. To alleviate this problem, Kwok and Cheung [16] proposed marginalized multiple instance kernels. These kernels specify the importance of an instance pair of two bags according to the consistency of their probabilistic instance labels. In our work, we also use the idea of marginalizing joint kernels, but we propose a unified framework to combine instance label inference and bag classification within an adaptive probabilistic graph-structured kernel. MIGraph and miGraph [28] are two other kernel-based MIL methods. Both algorithms work by mapping a bag into an undirected graph and designing a kernel between graph pairs. Bag classification is fulfilled by plugging the kernels in an SVM. The PPMM kernel [26] is an example of adaptive kernels for multi-instance classification. Each bag is represented by aggregate posteriors of instances in a mixture model, which summarize the frequencies of patterns in the bag. Given this, a parameterized kernel is defined, where the parameter controls the importance of major and minor patterns in the application domain. This parameter is learned by maximizing the alignment between the proposed kernel and the ideal kernel in an exhaustive search procedure. Thus, in this method the objective function for learning the kernel is different from the objective function for classification, while in our proposed framework the kernel learning is integrated in a single, unified optimization problem.

Probabilistic graphical models are powerful tools for multi-instance classification, given their ability to model the structured relationships between the instances in a bag. Deselaers and Ferrari [7] proposed MI-CRF. In this method, the bags are modelled as nodes in a CRF, where each node can take one of the instances of the bag as its state. Bags are jointly trained and classified in this model. The model uses instance classifiers as unary terms and dissimilarity measures between witness instances as pairwise terms. Warrell and Torr [27] proposed another CRF-based method. This method provides a structured bag model, by constructing a CRF over the instances, instance labels, and the bag label. In this CRF, hard and soft MIL constraints are incorporated in the model by defining energy functions between the labels. Inference of the proposed CRF is performed *approximately* by dual decomposition, and the models are trained by deterministic annealing.

Tarlow et al. [24] proposed a model to approach MIL by CRFs with cardinality potentials defined on instance labels. Learning is performed by regularized likelihood maximization using gradient ascent algorithms proposed for hidden CRFs. Hajimirsadeghi et al. [12] proposed a framework to model different multiple instance assumptions by parameterizing cardinality potentials in a CRF. Learning is formulated as a discriminative max-margin optimization problem.

In this paper, we build on this line of graphical model-based multi-instance classification. We learn an adaptive kernel on hCRFs modeling bags (c.f. kernel conditional random fields [17]). This allows the direct incorporation of powerful kernels that can model not only the relations between instances and instance labels, but also the level of ambiguity of instance labels and a task-specific bag-level kernel classifier.

3 Preliminaries

In this section, we define our MIL notation and provide a description of the cardinality potential model for multi-instance classification.

3.1 Multi-Instance Learning Notation

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$ denote a bag with m instances and a binary bag label $Y \in \{-1, +1\}$. Each instance is represented by a fixed-size feature vector $\mathbf{x}_i = [x_{i1}, \cdots, x_{iD}] \in \mathbb{R}^D$. The instances also have unknown binary labels $y_i \in \{0, 1\}$ (where 0 represents a negative label). Given this, the collective binary instance labels of a bag are denoted by $\mathbf{y} = \{y_1, \cdots, y_m\}$. The whole set of training data is represented by $\{(\mathbf{X}_1, Y_1), \cdots, (\mathbf{X}_N, y_N)\}$. Finally, the goal is to learn a bag classification function $f(\mathbf{X})$, which specifies the score of $\mathbf{X}$ being positive.

3.2 Cardinality Potential Model for Multi-Instance Learning

A cardinality potential is defined in terms of counts of variables which take some particular values. For example, with binary variables, it is defined in terms of the number of positive and negative labeled variables. Given a set of binary random variables $\mathbf{y} = \{y_1, y_2, \cdots, y_m\}$ ($y_i \in \{0, 1\}$), the standard cardinality potential model is described by the joint probability

$$P(\mathbf{y}) = \frac{C(\sum_i y_i) \prod_i \exp(\varphi_i y_i)}{\sum_{\mathbf{y}} C(\sum_i y_i) \prod_i \exp(\varphi_i y_i)}, \tag{1}$$

which consists of one cardinality potential $C(\cdot)$ over all the variables and unary potentials $\exp(\varphi_i y_i)$ on each single variable. Tarlow et al. [24] proposed an exact algorithm to perform sum-product inference and compute all marginal probabilities of this model in $O\left(m \log^2 m\right)$ time.

Multiple instance assumptions are usually defined on the counts of positive labeled instances in a bag. For example, the standard assumption states that at least one instance in a positive bag is positive. So, it is intuitive that the MIL constraints can be modeled by a cardinality potential over the instance labels. Using the standard cardinality potential model as the core, a conditional random field (CRF) is constructed to model the likelihood of a bag $\mathbf{X}$ with the bag label Y and the instance labels $y_i \in \{0, 1\}$, which factorizes as [24]:

$$P(Y, \mathbf{y} | \mathbf{X}; \boldsymbol{\theta}) \propto \phi^C(Y, \mathbf{y}) \prod_i \phi^I_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i). \tag{2}$$

A graphical representation of the model is shown in Figure 1. In our framework, we call this the multi-instance cardinality potential model (MICPM). In this model, $\phi^C(Y, \mathbf{y})$ is a clique potential over all the instance labels and the bag label. It is used to model assumptions and formulated as $\phi^C(Y, \mathbf{y}) = C^{(Y)}(\sum_i y_i)$. $C^{(+1)}$ and $C^{(-1)}$ are cardinality potentials for positive and negative bag labels, and in general could be expressed by any cardinality function which models MIL constraints. However, in this paper we only work with the Normal model in (3) and the Majority model in (4).

$$C^{(+1)}(c) = \exp\left(-(\frac{c}{m} - \mu)^2 / 2\sigma^2\right) \text{ and } C^{(-1)}(c) = \exp\left(-(\frac{c}{m})^2 / 2\sigma^2\right). \tag{3}$$

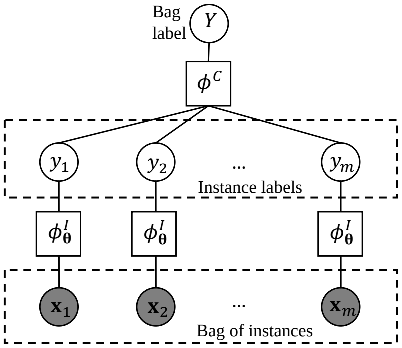

Figure 1: Graphical illustration of the proposed model for multiple instance learning. Potential functions relate instances $\mathbf{x}_i$ to labels y_i. A clique relates all instance labels y_i to the bag label Y.

$$C^{(+1)}(c) = \mathbb{1}(\frac{c}{m} >= 0.5) \text{ and } C^{(-1)}(c) = \mathbb{1}(\frac{c}{m} < 0.5). \tag{4}$$

The parameter μ in the Normal model controls the ratio of positive labeled instances in a bag.

$\phi^I_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i)$ represents the potential between each instance and its label, and it is parameterized as:

$$\phi^I_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i) = \exp\left(\boldsymbol{\theta}^t \mathbf{x}_i\, y_i\right) \tag{5}$$

Given (3) and (5), the joint probability of the CRF in (2) can be rewritten as

$$P(Y, \mathbf{y}|\mathbf{X}; \boldsymbol{\theta}) \propto C^{(Y)}(\sum_i y_i) \prod_i \exp\left(\boldsymbol{\theta}^t \mathbf{x}_i\, y_i\right). \tag{6}$$

Hence, the bag label likelihood alone is given by

$$P(Y|\mathbf{X}; \boldsymbol{\theta}) = \sum_{\mathbf{y}} P(Y, \mathbf{y}|\mathbf{X}; \boldsymbol{\theta}) = \frac{Z^{(Y)}}{\sum_{Y'} Z^{(Y')}}, \tag{7}$$

$$\text{where} \qquad Z^{(Y)} = \sum_{\mathbf{y}} \left(C^{(Y)}(\sum_i y_i) \prod_i \exp\left(\boldsymbol{\theta}^t \mathbf{x}_i\, y_i\right) \right) \tag{8}$$

is the partition function of a standard cardinality potential model, which can be computed by the proposed sum-product inference algorithm of [24] in $O\left(m \log^2 m\right)$ time.

Finally, the maximum a posteriori estimation of the learning parameters given the parameter prior distributions is obtained by maximizing the following log likelihood function:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_i \log P(Y|\mathbf{X}; \boldsymbol{\theta}) - \lambda\, r(\boldsymbol{\theta}). \tag{9}$$

This is a maximum likelihood optimization of a hidden conditional random field (hCRF) [22] with parameter regularization ($r(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_n$ for L_n-norm regularization). Gradient ascent is used to find the optimal parameters, where the gradients are obtained efficiently in terms of marginal probabilities.

4 Cardinality Potential Kernel for Multi-Instance Classification

In this section, we explain our proposed framework for multi-instance classification. At a high level, we define a kernel between multi-instance cardinality potential models of bags. We start by deriving a kernel based on marginalization over instance labels, starting from an existing probabilistic model for bags (Sec. 4.1). We then show how to jointly learn bag probabilistic model parameters and classifier parameters (Sec. 4.2). This joint learning algorithm sets instance-level labeling parameters and bag kernel parameters together, with a direct goal of improved classification performance. In experiments, we show that this joint learning can improve classification results.

4.1 Multi-Instance Cardinality Potential Kernel SVM

Given two bags $\mathbf{X}_p$ and $\mathbf{X}_q$ or their equivalent MICPMs described in Section 3.2, the joint kernel is defined between the combined variables $\mathbf{z}_p = (\mathbf{X}_p, \mathbf{y}_p)$ and $\mathbf{z}_q = (\mathbf{X}_q, \mathbf{y}_q)$:

$$k_z(\mathbf{z}_p, \mathbf{z}_q) = \sum_{i=1}^{m_p} \sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) k_y(y_{pi}, y_{qj}), \tag{10}$$

where $k_x(\cdot, \cdot)$ is a standard kernel between single instances, and $k_y(\cdot, \cdot)$ is a kernel defined on discrete instance labels. By marginalizing the joint kernel w.r.t. the hidden instance labels and with independence assumption between the bags, a kernel is defined on the bags as:

$$\tilde{k}(\mathbf{X}_p, \mathbf{X}_q) = \sum_{\mathbf{y}_p, \mathbf{y}_q} P(\mathbf{y}_p|\mathbf{X}_p) P(\mathbf{y}_q|\mathbf{X}_q) k_z(\mathbf{z}_p, \mathbf{z}_q). \tag{11}$$

After plugging (10) into (11), it can be shown that the marginalized joint kernel is reduced to

$$\sum_{i=1}^{m_p} \sum_{j=1}^{m_q} \sum_{\mathbf{y}_p, \mathbf{y}_q} \big(k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) k_y(y_{pi}, y_{qj}) P(y_{pi}|\mathbf{X}_p) P(y_{qj}|\mathbf{X}_q) \big). \tag{12}$$

In our proposed framework, $P(y_{pi}|\mathbf{X}_p)$ and $P(y_{qj}|\mathbf{X}_q)$ are obtained by

$$P(y_i|\mathbf{X}) = \sum_{Y} P(y_i|Y, \mathbf{X})\, P(Y|\mathbf{X}), \tag{13}$$

where $P(y_i|Y, \mathbf{X})$ are the marginal probabilities of a standard cardinality potential model, which can be computed efficiently in $O(m \log^2 m)$ time. Also $P(Y|\mathbf{X})$ is the bag label likelihood introduced in (7). In our proposed algorithm first the parameters $\boldsymbol{\theta}$ of the MICP model are learned using the likelihood maximization approach explained in Section 3.2. Next, the marginals are inferred and plugged into the kernel function of (12).

In general, any kernel for discrete spaces can be used as k_y. However, throughout this paper k_y is assumed to be

$$k_y(y_{pi}, y_{qj}) = \mathbb{1}(y_{pi} = y_{qj}). \tag{14}$$

Using this, the kernel in (12) is simplified as:

$$\begin{aligned} \tilde{k}(\mathbf{X}_p, \mathbf{X}_q) = & \sum_{i=1}^{m_p} \sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) P(y_{pi} = 1|\mathbf{X}_p) P(y_{qj} = 1|\mathbf{X}_q) \\ & + \sum_{i=1}^{m_p} \sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) P(y_{pi} = 0|\mathbf{X}_p) P(y_{qj} = 0|\mathbf{X}_q). \end{aligned} \tag{15}$$

Finally, to avoid bias towards the bags with large numbers of instances, the kernel is normalized as [10]:

$$k(\mathbf{X}_p, \mathbf{X}_q) = \frac{\tilde{k}(\mathbf{X}_p, \mathbf{X}_q)}{\sqrt{\tilde{k}(\mathbf{X}_p, \mathbf{X}_p)} \sqrt{\tilde{k}(\mathbf{X}_q, \mathbf{X}_q)}}. \tag{16}$$

We call the resulting kernel the multi-instance cardinality potential (MICP) kernel. By using this kernel in the standard kernel SVM, we propose the multi-instance cardinality potential kernel SVM (MICPK-SVM) method for multi-instance classification.

4.2 Multi-Instance Cardinality Potential Kernel Learning

In this section, we show how to learn the parameters of the MICP model $\boldsymbol{\theta}$ in the proposed MICP kernel, integrated in a kernel SVM classifier. As a result, instead of generative pre-learning of a kernel by likelihood maximization of the MICP model, the MICP kernel is specifically trained for the target classification task. Given a parameterized kernel $k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q) = \Phi_{\boldsymbol{\theta}}(\mathbf{X}_p).\Phi_{\boldsymbol{\theta}}(\mathbf{X}_q)$, the goal is to learn a bag classification function $f(\mathbf{X}) = \mathbf{w}^t \Phi_{\boldsymbol{\theta}}(\mathbf{X}) + b$ to predict the binary bag label $Y = \operatorname{sign}(f(\mathbf{X})) \in \{-1, +1\}$. To this end, we follow the generalized multiple kernel learning framework [25] to optimize the SVM primal objective function w.r.t. the classifier parameters $\mathbf{w}$ and b, and kernel parameters $\boldsymbol{\theta}$. First the SVM primal is rewritten as a nested optimization

$$\begin{aligned} &\min_{\boldsymbol{\theta}} T(\boldsymbol{\theta}), \\ \text{where } \ &T(\boldsymbol{\theta}) = \min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_n \max\left(0, 1 - Y_n f(\mathbf{X}_n)\right) + r(\boldsymbol{\theta}). \end{aligned} \tag{17}$$

In the outer optimization the kernel parameters $\boldsymbol{\theta}$ are optimized, and in the inner optimization the SVM learning weights are estimated. To solve this problem in a gradient descent approach, it is required to calculate $\nabla_{\boldsymbol{\theta}} T$. Using the duality theorem, it is shown that $\nabla_{\boldsymbol{\theta}} T = \nabla_{\boldsymbol{\theta}} W$ [25], where W is the dual formulation of T:

$$\begin{gathered} W(\boldsymbol{\theta}) = \max_{\boldsymbol{\alpha}} \ \mathbf{1}^t\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^t\mathbf{Y}\mathbf{K}_{\boldsymbol{\theta}}\mathbf{Y}\boldsymbol{\alpha} + r(\boldsymbol{\theta}), \\ \text{subject to } \ \mathbf{1}^t\mathbf{Y}\boldsymbol{\alpha} = 0, \ \ 0 \le \boldsymbol{\alpha} \le C, \end{gathered} \tag{18}$$

and $\boldsymbol{\alpha}$ is the vector of dual variables, $\mathbf{Y}$ is a diagonal matrix made up of all training bag labels, and $\mathbf{K}_{\boldsymbol{\theta}}$ is the kernel matrix of all training bag pairs for a given $\boldsymbol{\theta}$. It is proven in [6, 4] that if k, r, $\nabla_{\boldsymbol{\theta}} k$ and $\nabla_{\boldsymbol{\theta}} r$ are smooth functions of $\boldsymbol{\theta}$ and if $\boldsymbol{\alpha}^*$, which is the solution to the dual maximization problem in (18), is unique, $\nabla_{\boldsymbol{\theta}} W$ exists, and the derivatives are expressed by

$$\frac{\partial T}{\partial \theta_d} = \frac{\partial W}{\partial \theta_d} = \frac{r}{\partial \theta_d} - \frac{1}{2}\boldsymbol{\alpha}^{*t}\mathbf{Y}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_d}\mathbf{Y}\boldsymbol{\alpha}^*. \tag{19}$$

Note that α_n^* is zero except for the support vectors, and consequently, $\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_d}$ is only required to be computed for the support vectors. This can significantly reduces the computational cost of the algorithm, especially, if it is integrated with ideas such as reduced support vector machines (RSVM) [18]. Using the derivatives in a coordinate descent approach, learning is an iterative procedure of alternating between finding $\boldsymbol{\alpha}^*$ by a standard kernel SVM dual optimization in (18) given $\boldsymbol{\theta}$ fixed, and next updating $\boldsymbol{\theta}$ by moving in the direction of derivities calculated in (19) given $\boldsymbol{\alpha}^*$ from the previous step. Note that if the L_1 regularization function $r(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_1$ is desired, the smooth L_1-norm approximation [23] should be employed to satisfy the necessary conditions.

To use this algorithm, the key issue is to calculate the derivatives of the kernel matrix $\mathbf{K}_{\boldsymbol{\theta}} = [k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q)]_{N\times N}$ w.r.t. the learning parameters θ_d:

$$\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_d} = \left[\frac{\partial k_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q)\right]_{N\times N}. \tag{20}$$

The derivatives of each element of the kernel matrix are given by

$$\begin{aligned} \frac{\partial k_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q) = &\frac{\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q)}{\sqrt{\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_p)}\sqrt{\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_q, \mathbf{X}_q)}} \\ &- \frac{k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q)\,\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_p)}{2\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_p)} - \frac{k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q)\,\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_q, \mathbf{X}_q)}{2\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_q, \mathbf{X}_q)}, \end{aligned} \tag{21}$$

where

$$\begin{aligned} &\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q) = \sum_{i=1}^{m_p}\sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) \\ &\left(\frac{\partial P_{\boldsymbol{\theta}}(y_{pi}|\mathbf{X}_p)}{\partial \theta_d}\Big|_{y_{pi}=1} \cdot P_{\boldsymbol{\theta}}(y_{qj}=1|\mathbf{X}_q) + P_{\boldsymbol{\theta}}(y_{pi}=1|\mathbf{X}_p) \cdot \frac{\partial P_{\boldsymbol{\theta}}(y_{qj}|\mathbf{X}_q)}{\partial \theta_d}\Big|_{y_{qj}=1}\right. \\ &\left. + \frac{\partial P_{\boldsymbol{\theta}}(y_{pi}|\mathbf{X}_p)}{\partial \theta_d}\Big|_{y_{pi}=0} \cdot P_{\boldsymbol{\theta}}(y_{qj}=0|\mathbf{X}_q) + P_{\boldsymbol{\theta}}(y_{pi}=0|\mathbf{X}_p) \cdot \frac{\partial P_{\boldsymbol{\theta}}(y_{qj}|\mathbf{X}_q)}{\partial \theta_d}\Big|_{y_{qj}=0}\right). \end{aligned} \tag{22}$$

So, the only thing needed is to find $\frac{\partial P_{\boldsymbol{\theta}}(y_i|\mathbf{X})}{\partial \theta_d}$ for a given y_i in a bag $\mathbf{X}$. Actually, calculating these derivities is not straightforward, but taking derivities of $\log P(y_i|\mathbf{X})$ is quite similar to taking the derivatives of the log likelihood function in hCRFs [22]. Thus, by exploiting the relations in [22] and the chain rule $\frac{\partial \log P_{\boldsymbol{\theta}}}{\partial \theta_d} = \frac{1}{P_{\boldsymbol{\theta}}} \frac{\partial P_{\boldsymbol{\theta}}}{\partial \theta_d}$, it can be shown that

$$\frac{\partial P_{\boldsymbol{\theta}}(y_i|\mathbf{X})}{\partial \theta_d} = P_{\boldsymbol{\theta}}(y_i|\mathbf{X}) \Big(\sum_{i'} \sum_{y_{i'}} P_{\boldsymbol{\theta}}(y_{i'}|y_i, \mathbf{X})\, x_{i'd}\, y_{i'} - \sum_{i'} \sum_{y_{i'}} P_{\boldsymbol{\theta}}(y_{i'}|\mathbf{X})\, x_{i'd}\, y_{i'} \Big). \tag{23}$$

The calculation of $P_{\boldsymbol{\theta}}(y_i|\mathbf{X})$ is described in (13). $P_{\boldsymbol{\theta}}(y_{i'}|y_i, \mathbf{X})$ can be calculated in the same way except that one of the hidden variables has been observed (i.e., canceled out), and consequently the cardinality potential of the resulting model (which has $m-1$ unobserved variables) has been modified accordingly. Thus, for each i and all i', $P_{\boldsymbol{\theta}}(y_{i'}|y_i, \mathbf{X})$ is computed in $O\left(m \log^2 m\right)$ time.

Putting all the above together we propose a new algorithm, namely multi-instance cardinality potential kernel learning (MICPKL). The pseudo-code of this algorithm is provided in Algorithm 1.

Algorithm 1 Multiple Instance Cardinality Potential Kernel Learning

```
Input: Training data $\{(\mathbf{X}_n, Y_n)\}_{n=1}^{N}$, Cardinality potential parameters $\mu$ and $\sigma$, Regularization parameters $\lambda$ and $C$, Maximum number of iterations.
Initialize $\boldsymbol{\theta}$ randomly.
repeat
  $\mathbf{K} = [k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q)]_{N \times N}$.
  Find $\boldsymbol{\alpha}^*$ by solving the standard kernel SVM dual optimization in (18) with $\mathbf{K}$.
  Find $\frac{\partial \mathbf{K}}{\partial \theta_d}$ using (20), (21), (22), (23).
  $\theta_d = \theta_d - \eta(\frac{r}{\partial \theta_d} - \frac{1}{2}\boldsymbol{\alpha}^{*t}\mathbf{Y}\frac{\partial \mathbf{K}}{\partial \theta_d}\mathbf{Y}\boldsymbol{\alpha}^*)$.
until converged or maximum number of iterations
```

4.3 Computational Complexity

We analyze the time complexity of the MICP kernel. Assuming that evaluation of the primitive kernel k_x takes $O(d)$ time, $k_x(\cdot, \cdot)$ between all instance pairs of two bags X_p and X_q can be computed in $O(m_p m_q d)$. As we explained in Section 4.1, the time complexity of computing $P(y_i|Y, \mathbf{X})$ is $O\left(m \log^2 m\right)$. So, the kernel in (15) can be evaluated in $O(m_p m_q d + m_p \log^2 m_p + m_q \log^2 m_q)$ time. As a result, the computational complexity of prediction for a single bag $\mathbf{X}_p$ with this kernel is $O(N_{sv}\, \bar{m}\, m_p\, d + N_{sv}\, \bar{m}\, \log^2 \bar{m} + m_p\, \log^2 m_p)$, where N_{sv} is the number of support vectors and $\bar{m}$ is the maximum number of instances in a training bag.

Second, we analyze the computational complexity of the MICPK-SVM algorithm. First, the parameters of an MICPM should be learned. Training this hCRF with likelihood maximization takes $O(N_{iter}\, N\, \bar{m} \log^2 \bar{m} + N_{iter}\, N\, \bar{m}\, d)$ time, where N_{iter} is the number of iterations of the gradient ascent algorithm. The kernel matrix can be computed in $O(N^2\, \bar{m}^2\, d + N\, \bar{m}\, \log^2 \bar{m})$ time. Finally, assuming the quadratic programming to solve the SVM dual takes $O(N^3)$ time[1], the computational complexity of the entire algorithm is $O(N_{iter}\, N\, \bar{m} \log^2 \bar{m} + N_{iter}\, N\, \bar{m}\, d + N^2\, \bar{m}^2\, d + N^3)$.

Third, we show the computational complexity of MICPKL algorithm. According to what was explained in Section 4.2, for a given bag X, computing $P(y_{i'}|y_i, \mathbf{X})$ for all the instances takes $O(m^2 \log^2 m)$ time, and so computation of all the derivatives in (23) takes $O(m^2 \log^2 m + m^2 d)$. Consequently, the time complexity of finding the kernel derivatives in (22) and (21) is $O(m_p\, m_q\, d + m_p^2 \log^2 m_p + m_p^2\, d + m_q^2 \log^2 m_q + m_q^2\, d)$. Using this, the kernel matrix derivatives are computed in $O(N_{sv}^2\, \bar{m}^2\, d + N_{sv}\, \bar{m}^2 \log^2 \bar{m})$ time, where N_{sv} is the number of support vectors. Finally, with the assumption that the quadratic programming in (18) takes $O(N^3)$, the computational complexity of MICPKL is given by $O(N_{iter}\, N_{sv}^2\, \bar{m}^2\, d + N_{iter}\, N_{sv}\, d\, \bar{m}^2 \log^2 \bar{m} + N_{iter}\, N^3)$ for N_{iter} iterations.

5 Experiments

In this section, the performance of the proposed methods is evaluated on different datasets.

[1] In our experiments, we used LIBSVM [3] solver, which can be much more efficient than $O(N^3)$ in practice.

5.1 MIL Benchmark Datasets

The standard MIL benchmark datasets are the *Elephant*, *Fox*, *Tiger* image categorization datasets [2], and the *Musk1* and *Musk2* drug activity prediction datasets [8]. Though dated, these are the standard benchmark on which MIL algorithms are evaluated. In all the experiments, we have preprocessed datasets by scaling the features of the original datasets to the range $[0, 1]$. We run our methods with Normal cardinality potentials where σ is set to 0.1 and μ was estimated by grid search in $\{0.1, 0.2, \cdots, 1.0\}$ for MICPM (the same values were then used for MICPK-SVM and MICPKL). The regularization weights λ and C are also roughly optimized on 10-fold cross-validation accuracy. For MICPK-SVM and MICPKL, similar to the MI-Kernel setting in [10], we use RBF kernels as the instance kernels. The results are reported based on 10-fold cross-validation classification accuracy and compared with the state-of-the-art MIL methods in Table 1. It can be seen that MICPKL performs well compared to the other methods. More specifically, it achieves the best accuracy on the Elephant, Fox, and Tiger data sets. In addition, MICPK-SVM is by and large comparable to the best methods although it is more computationally efficient than MICPKL.

Table 1: Comparison between state-of-the-art MIL methods. The best and second best results are highlighted in bold and italic face respectively.

Method	Elephant	Fox	Tiger	Musk1	Musk2	Average
MICPM	84	65	86	81	83	79.8
MICPK-SVM	*88*	63	*87*	89	*89*	*83.2*
MICPKL	**89**	**71**	**88**	89	*89*	**85.2**
MIMN [12]	**89**	64	*87*	86	**90**	*83.2*
MIRealBoost [13]	83	63	73	**91**	77	77.4
ClassSetMaxRBM$^{\mathrm{XOR}}$ [21]	*88*	60	83	84	84	79.8
MI-CRF [7]	85	*68*	83	88	85	81.8
MIGraph [28]	85	61	82	*90*	**90**	81.6
miGraph [28]	87	62	86	*90*	**90**	83.0
ALP-SVM [11]	84	66	86	86	86	81.6
MILES [5]	81	62	80	88	83	78.8
MI-Kernel [10]	84	60	84	88	*89*	81.0
mi-SVM [2]	82	58	79	87	84	78.0
MI-SVM [2]	81	59	84	78	84	77.2

6 Conclusion

We proposed a novel kernel learning framework for multi-instance classification. This framework is constructed based on a multi-instance cardinality potential model, which can explore different levels of ambiguity in instance labels and model different cardinality-based assumptions. In addition, the proposed adaptive kernel can help to perform classification within a bag-level paradigm and in a task-specific discriminative embedded space of even infinite dimensionality. The results of our experiments on standard MIL benchmark datasets showed the efficacy of the proposed kernel learning approach for multi-instance classification. This kernel learning approach can be extended to different variants of conditional random fields for other structured prediction tasks.

References

[1] J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

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

[3] C. Chang and C. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[4] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

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

[6] J. M. Danskin. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer-Verlag New York, 1967.

[7] T. Deselaers and V. Ferrari. A conditional random field for multiple-instance learning. In *International Conference on Machine Learning (ICML)*, 2010.

[8] 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.

[9] L. Duan, W. Li, I. Tsang, and D. Xu. Improving web image search by bag-based re-ranking. *IEEE Transactions on Image Processing*, 20(11):3280–3290, 2011.

[10] T. Gärtner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *International Conference on Machine Learning (ICML)*, pages 179–186, 2002.

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

[12] H. Hajimirsadeghi, J. Li, G. Mori, M. Zaki, and T. Sayed. Multiple instance learning by discriminative training of markov networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

[13] H. Hajimirsadeghi and G. Mori. Multiple instance real boosting with aggregation functions. In *International Conference on Pattern Recognition (ICPR)*, 2012.

[14] H. Hajimirsadeghi and G. Mori. Multi-instance classification by max-margin training of cardinality-based markov networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1839–1852, 2017.

[15] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] J. T. Kwok and P.-M. Cheung. Marginalized multi-instance kernels. In *IJCAI*, pages 901–906, 2007.

[17] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *International Conference on Machine Learning (ICML)*. ACM, 2004.

[18] Y.-J. Lee and O. L. Mangasarian. Rsvm: Reduced support vector machines. In *SIAM International Conference on Data Mining*, 2001.

[19] W. Li, L. Duan, I. W.-H. Tsang, and D. Xu. Batch mode adaptive multiple instance learning for computer vision tasks. In *CVPR*, 2012.

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

[21] J. Louradour and H. Larochelle. Classification of sets using restricted boltzmann machines. In *Uncertainty in Artificial Intelligence (UAI-11)*, 2011.

[22] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(10):1848–1852, 2007.

[23] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Machine Learning: ECML 2007*, pages 286–297. Springer, 2007.

[24] D. Tarlow, K. Swersky, R. Zemel, R. Adams, and B. Frey. Fast exact inference for recursive cardinality models. In *Uncertainty in Artificial Intelligence (UAI-12)*, 2012.

[25] M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *International Conference on Machine Learning (ICML)*, pages 1065–1072. ACM, 2009.

[26] H.-Y. Wang, Q. Yang, and H. Zha. Adaptive p-posterior mixture-model kernels for multiple instance learning. In *International Conference on Machine Learning (ICML)*, 2008.

[27] J. Warrell and P. H. Torr. Multiple-instance learning with structured bag models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 369–384. Springer, 2011.

[28] Z. Zhou, Y. Sun, and Y. Li. Multi-instance learning by treating instances as non-iid samples. In *International Conference on Machine Learning (ICML)*, 2009.