Multiple Instance Learning for Visual Recognition: Learning Latent Probabilistic Models

by

Hossein Hajimirsadeghi

M.Sc., University of Tehran, 2010
B.Sc., University of Tehran, 2008

Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2015

Approval

Name:	Hossein Hajimirsadeghi
Degree:	Doctor of Philosophy (Computing Science)
Title:	*Multiple Instance Learning for Visual Recognition: Learning Latent Probabilistic Models*
Examining Committee:	**Dr. Ze-Nian Li** (chair) Professor

Dr. Greg Mori
Senior Supervisor
Professor _____

Dr. Anoop Sarkar
Supervisor
Professor _____

Dr. Ping Tan
Internal Examiner
Assistant Professor _____

Dr. Ming-Hsuan Yang
External Examiner
Associate Professor
Electrical Engineering and Computer Science
University of California, Merced _____

Date Defended: 8 September 2015

Abstract

Many visual recognition tasks can be represented as multiple instance problems. Two examples are image categorization and video classification, where the instances are the image segments and video frames, respectively. In this regard, detecting and counting the instances of interest can help to improve recognition in a variety of applications. For example, classifying the collective activity of a group of people can be directed by counting the actions of individual people. Further, encoding the cardinality-based relationships can reduce sensitivity to clutter or ambiguity, in the form of individuals not involved in a group activity or irrelevant segments/frames in an image/video.

Multiple instance learning (MIL) aims to use these counting relations in order to recognize patterns from weakly supervised data. Contrary to standard supervised learning, where each training instance is labeled, in the MIL paradigm a bag of instances share a label, and the instance labels are hidden. This weak supervision significantly reduces the cost of full annotation in many recognition tasks. However, it makes learning and recognition more challenging. In a general MIL problem, three major issues usually emerge: how to infer instance labels without full supervision; how the cardinality relations between instance labels contribute to predict the bag label; how the the bag as a whole entity which integrates the instances is labeled. In this thesis, we try to address all these challenges.

To this end, first we propose a boosting framework for MIL, which can model a wide range of soft and linguistic cardinality relations. Next, a probabilistic graphical model is proposed to capture the interactions and interrelations between instances, instance labels, and the whole bag. This is a general and flexible model, which can encode any cardinality-based relations. For training this model, we introduce novel algorithms based on latent max-margin classification, kernel learning, and gradient boosting. Thus, very rich and high-capacity models are obtained for bag classification. We evaluate our proposed methods in various applications such as image classification, human group activity recognition, human action recognition, video recognition, unconstrained video event detection, and video summarization.

Keywords: Multiple instance learning; probabilistic graphical models; latent structured models; visual recognition

Table of Contents

List of Tables

List of Figures

Chapter 1

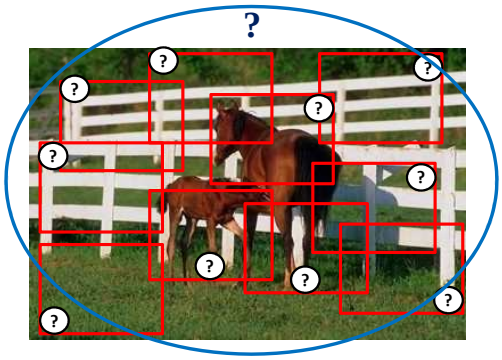
Introduction

In the standard supervised learning, training data is given as labeled instances, and the goal is to train a classifier to predict the labels of new instances. However, this full supervision might be costly in some practical applications. For example, full annotation of objects or people in all training images needs a lot of human labor. Further, this annotation might be noisy or inaccurate because of human misjudgment or fatigue. Contrary to standard supervised learning, multiple instance learning (MIL) is a type of weakly supervised learning, where training data is given as *bags* (i.e., sets) of *instances*. In MIL, the bag labels are given for training, but the instance labels are missing. As illustrated in Fig. 1.1, a bag could be an image with a set of patches, segments, or people in it as instances; or a bag could be a long video sequence containing a set of frames or smaller temporal clips as instances.

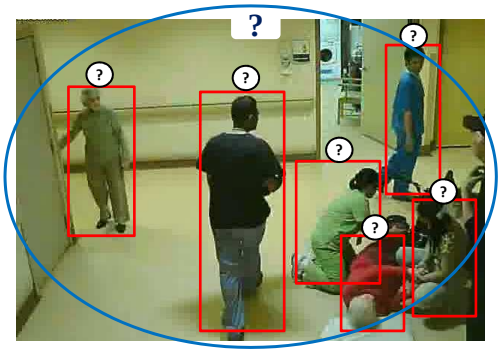
Multiple instance learning defines the new notion of labeled bags. For example, in the binary case, the data is represented as positive and negative bags. To define positive or negative bags, some *cardinality* (i.e., count-based) relations are used, which are also known as multi-instance (MI) assumptions. For example, a fall event in a nursing home (Fig. 1.1b) can be detected by using the following count-based assumption: "in a fall event at least one person is fallen." Actually, this is the most traditional multi-instance assumption in MIL literature, which states a bag is positive if at least one of the instances in the bag is positive. However, this assumption is not effective in all MIL application. For example, in group activity recognition (Fig. 1.1e) the collective activity of a group of individuals is determined by the *majority* of people involved in the activity.

These count-based assumptions not only help to capture some intrinsic or intuitive relations in the problem but also enhances robustness against noise, ambiguity, or clutter, incurred by low-quality representation/annotation of input data or imperfect intermediate processing stages. For example, in group activity recognition, there might be clutter in the form of people in a scene performing unrelated actions, or noise in the form imperfect person detection or noisy tracking. Another example is unconstrained internet video analysis. Detecting events in internet videos (Fig. 1.1c) or determining whether part of a video is

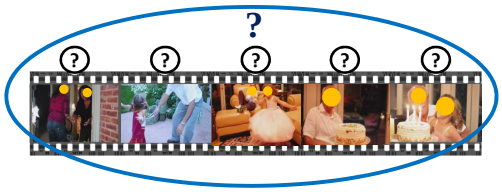
interesting (Fig. 1.1d) are challenging for many reasons, including temporal clutter – videos often contain frames unrelated to the event of interest or that are difficult to classify. In this dissertation, we present frameworks built on probabilistic multi-instance models to encode these count-based relations and deal with the ambiguity or clutter in data.



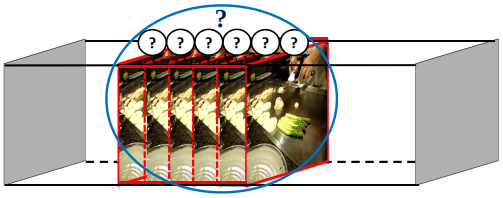
(a) What is the image category?



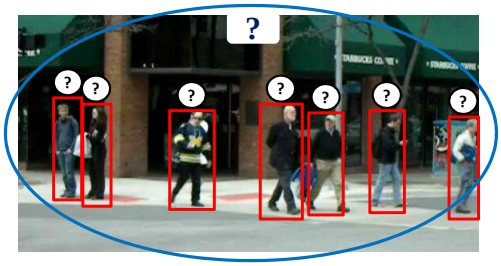
(b) Is this a fall scene?



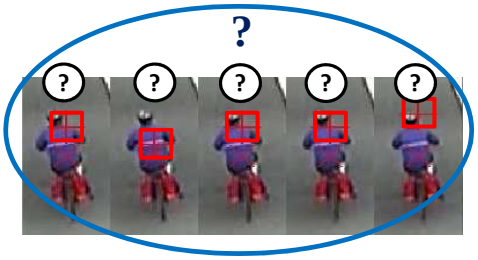
(c) What is the video event?



(d) Is this video segment interesting?



(e) What is the collective activity?



(f) Is the cyclist wearing helmet?

Figure 1.1: In multiple instance learning framework, data is given as bags of instances. Multiple instance learning can be applied to a variety of visual recognition problems. Here, some examples are presented. (a) Image categorization: an image is represented as a bag of image patches extracted from regions of interests in the image. (b) Fall detection in a nursing home and (e) human group activity recognition: a scene is represented as a bag of individuals doing different actions. (c) Video event detection: a video is represented as a bag of frames or temporal segments. (d) Detecting interesting video segments for video summarization: a video segment is represented as a bag of frames. (f) Cyclist helmet recognition: a cyclist track is represented as a bag of automatically extracted windows around the cyclist's head position estimate. The multi-instance assumption varies in different applications. One person is enough to detect a fall scene, while the majority of people are involved in a collective activity.

MIL methods aim to train a classifier to discriminate between positive and negative bags. In this regard, MIL algorithms are categorized into instance-space and bag-space methods. In the instance-space 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 instance labels contribute to bag prediction is important. This mechanism is usually built on the multi-instance assumptions. The standard MI assumption states that a bag is positive if it contains at least one positive instance, while in a negative bag all the instances are negative. This ambiguity in the instance labels is passed on to the learning algorithm, which should incorporate the information to classify bags. The standard assumption was proposed for the early applications of MIL (e.g. drug activity recognition [25]). However, the standard MI assumption gives a relatively weak prior information for many other MIL applications such as group activity recognition. Hence, using a more solid and informative assumption can help to train more robust and powerful classifiers.

In recent works, more relaxed and general MI assumptions have been studied such as *ratio-based* assumptions [67, 92], where the proportion of positive instances in a bag determines the bag label. For example, an immediate extension to the standard MI assumption is that a bag is positive if at least a certain *ratio* of instances are positive. According to [40, 67, 60, 69], encoding the level of ambiguity of instance labels (e.g., the portion of positive instances in a bag) in the classifier by using an appropriate MI assumption can significantly influence the accuracy of classification. One main contribution of this dissertation is to propose models which can encode general and intuitive multi-instance assumptions such as the "majority of instances in a bag are positive", or "a certain proportion of instances are positive", or "a few instances of a bag are positive", or "the more positive instances, the more probable positive bag". We show that this flexible multi-instance encoding can enhance visual recognition.

On the other hand, in the bag-space paradigm, a classifier is trained directly on bags by extracting discriminative information from the whole bag. For example, each bag is mapped to a single feature vector, which summarizes the whole bag information, and next a standard single-instance learner is used to classify the bags represented in the resulting vectorial embedding space. According to some comparative studies [5], by and large, bag-space methods tend to outperform instance-space methods for bag classification[1]. In fact, by defining appropriate kernel, distance, or mapping functions, the bag-space methods can extract unified metadata information, which can improve classification accuracy. In this respect, kernel methods can make it possible to work even in infinite-dimensional

[1]Note that most bag-space methods lack the ability for instance classification. Actually, in most MIL problems, the goal is to classify the bags. But, in many applications predicting the instance labels might also matter.

spaces, which might increase discrimination power of the original MIL classifier. Another contribution of this dissertation is proposing new kernels for multi-instance classification.

Putting all above together, this research is focused on proposing novel MIL algorithms to address three main issues: (1) weakly supervised learning of models which have latent instance labels; (2) encoding the level of ambiguity of instance labels by using general MI assumptions; and (3) extracting discriminative instance-level and bag-level information from bags and integrating this information to improve multi-instance classification (i.e., combining instance-space and bag-space methods). For the first one, we propose probabilistic models which represent instance labels as latent variables. For the second issue, we introduce linguistic aggregation functions or cardinality potential models to combine instance labels and make various MI assumptions. Encoding these general MI assumptions help to train more powerful classifiers which are more robust to the amount of ambiguity (i.e. true positive instance labels) in the bags and more compatible to the classification task. For the latter, we model the bags with structured graphical models integrating instance-level and bag-level representation of bags. Further, we propose a novel kernel to map the whole information stored and consolidated in these structured models into a discriminative vectorial embedding space.

1.1 Contributions

This dissertation contributes to visual recognition by proposing novel and general frameworks for multiple instance learning. The proposed methods help to encode various multi-instance assumptions and cardinality relations in visual recognition problems with weak supervision. The detailed contributions of this work are highlighted as follows[2].

- **Showing the importance of encoding cardinality relations for visual recognition.** We show in different applications such as group activity recognition (Chapters 4, 5, 6), image categorization (Chapters 3, 4), video classification (Chapter 4), video event detection (Chapters 5, 6), and video summarization (Chapter 5) that encoding cardinality relations improves visual recognition. This is because that either the cardinality relations are intuitive and intrinsic to the problem (e.g. group activity recognition) or at least they help to enhance robustness against clutter, noise, and ambiguity. Note that although modeling spatiotemporal relations is very common in computer vision, however, cardinality relations have been usually ignored in structured visual recognition. In fact, MIL methods have been traditionally used to handle the labeling ambiguity of independent instances rather than modeling an intrinsic cardinality relation between instances.

[2]These contributions have been reported in the following publications (or submissions) [47, 46, 50, 48, 49].

- **Novel MIL frameworks to model general multi-instance assumptions or cardinality relations.** We propose frameworks which can encode the standard MI assumption as well as more general MI assumptions in order to model various cardinality-based relations or deal with different levels of ambiguity in data. A summary of the encoded multi-instance assumptions is provided as follows:

  (a) soft linguistic assumptions such as "few instances are positive" or "many instances are positive" (Chapter 3).

  (b) hard ratio-based assumptions such as "at least 30% of instances are positive" or "at most 60% of instances are positive" (Chapters 4, 5).

  (c) soft probabilistic assumptions such as "the more positive instances, the more probable positive bag" (Chapter 5).

  (d) An estimated cardinality-based relation between instance labels which is learned directly from training data (Chapter 4).

  (e) Metadata assumptions, i.e., assumptions that are not based on cardinality of instance labels but extracted from the bag as a whole entity (Chapter 5 and Appendix A).

  (f) Mixed assumptions: Our proposed structured models can also combine cardinality assumptions with metadata assumptions to integrate local and global representations of a bag (Chapters 4, 6).

  Also the proposed models have exact and efficient inference algorithms, which make prediction fast and reliable.

- **Novel learning algorithms for training latent probabilistic models.** We propose novel learning algorithms based on boosting (Chapter 3), latent max-margin training (Chapter 4), kernel learning (Chapter 5, Appendix A), and structured functional optimization (Chapter 6). All these methods (except the boosting algorithm in Chapter 3) are principled algorithms with convergence guarantees and predictable computational complexities.

Chapter 2

Previous Work

MIL has been successfully used in many applications such as drug activity prediction [25], image categorization [18], text categorization [6], content-based image retrieval [120], text-based image retrieval [67, 30], automatic image annotation [110], object detection [99], object localization [37], tracking [8], and video analysis/recognition [52, 64, 104, 60]. Dietterich et al. [25] proposed the first application of MIL. This application aims to classify molecules to "musk" or "non-musk" type for the purpose of drug activity prediction. In fact, because of twisting or bending, each molecule can have different conformations (i.e. shapes). However, it is unknown or significantly hard to figure out which conformation results in the musk label. Thus, each molecule is represented as a bag of conformations, where in a positive bag at least one of the conformations is of musk type and in a negative bag all the conformations are of non-musk type.

Because of the multi-unit structure of MIL, it is a natural fit to various computer vision problems. Chen et al. [18] treated each image as a bag of instances corresponding to blocks, regions, or patches of the image for the purpose of image categorization. Li et al. [67] and Duan et al. [30] used MIL to handle ambiguity in labels of training images incurred by coarse ranking of web images. Wu et al. [110] proposed deep MIL models for automatic image annotations. An image is formed as a bag of noisy candidate keywords extracted from web data. Next, the proposed MIL model identifies the relevant keywords which provide more rich annotations for the image. Galleguillos et al. [37] showed the application of MIL in object localization. An image is represented as a bag of segments, where by classifying the segments, the object is also localized. Viola et al. [99] used MIL to overcome the ambiguity in object annotation, by representing each image with a bag of windows centered around the ground truth. Likewise, in object tracking Babenko et al. [8] used several blocks around the estimated object location to construct a positive training bag for MIL. Hu et al. [52] applied MIL to human action detection in videos. In the proposed approach, each video was segmented at variant location/scale with different temporal length to make the instances of a bag. Leung et al. [64] used MIL to increase robustness against label noise in video

classification. To this end, positive training videos were divided into positive bags instead of being individually labeled. This approach was used for YouTube video classification, and the experiments verified increase of robustness against label noise. Wang et al. [104] applied MIL to recognize videos by treating each video as a bag of frames. Lai et al. [60] applied MIL for video event detection by representing a video as multi-granular temporal video segments.

To solve the problems discussed above, a variety of MIL algorithms have been proposed in the last two decades. The input to these algorithms are labeled bags of instances. Let $\mathbf{X}_i = \{\mathbf{x}_{i1}, \cdots, \mathbf{x}_{im}\}$ denote a bag with m instances and a binary label $Y_i \in \{-1, 1\}$. Each instance is represented by a fixed-size feature vector $\mathbf{x}_{ij} \in \mathbb{R}^d$. In most MI assumptions, the instances have also binary labels $y_{ij} \in \{-1, 1\}$, which are unknown. Finally the whole training data is represented by $\{(\mathbf{X}_1, Y_1), \cdots, (\mathbf{X}_N, Y_N)\}$. Given this, the goal is to learn a bag classification function $F(\mathbf{X})$, which scores the bag for being positive. In this respect, some methods try to firstly learn an instance-level scoring function $f(\mathbf{x})$.

Generally, MIL methods can be categorized based on different criteria such as the learning approach (e.g. maximum likelihood, max-margin, etc.) [7], the multi-instance assumption (e.g. standard assumption, ratio-based assumption, etc.) [33], or the space/level that the discriminative information lies in the method (instance-space vs. bag-space) [5]. In this chapter, we review a variety of representative MIL methods in two main sections of instance-space methods (a.k.a. instance-level methods) and bag-space methods (a.k.a. bag-level methods). However, we also try to briefly explain the learning approach and the multi-instance assumption used in each method. These algorithms provide guidelines and ideas to propose new MIL methods, which will be presented in the subsequent chapters.

2.1 Instance-Space Methods

Instance-level methods classify bags by aggregation of instance-level classification scores. To this end, an instance-level classifier is trained to classify positive and negative instances in the instance space, and based on these classifiers a bag-level classifier is obtained.

2.1.1 Methods Encoding Standard MI Assumption

Most early algorithms for MIL follow the standard MI assumption. Here, we review some of these algorithms.

Axis-Parallel Rectangle

Dietterich et al. [25] introduced one of the first methods for MIL. The main idea is to find a hyper-rectangle $\mathbf{R}$ that maximizes the number of positive bags which have at least one instance in $\mathbf{R}$, and at the same time, maximizes the number of negative bags which have

no instance in $\mathbf{R}$. This rectangle is called an axis-parallel rectangle (APR) and considered as an instance classifier:

$$f(\mathbf{x}_{ij}, \mathbf{R}) = \begin{cases} 1 & \text{if } \mathbf{x}_{ij} \in \mathbf{R} \\ 0 & o.w. \end{cases}. \tag{2.1}$$

Based on this, a bag is classified as positive if at least one of the instances falls in $\mathbf{R}$ – i.e., at least one of the instances is positive:

$$F(\mathbf{X}_i) = \max_j f(\mathbf{x}_{ij}, \mathbf{R}) \tag{2.2}$$

Hence, this algorithm works based on the standard MI assumption. To find the hyper-rectangle, three heuristic algorithms are proposed in [25]: the "standard" algorithm, the "outside-in" algorithm, and the "inside-out" algorithm. The standard algorithm finds the smallest APR which encloses all the instances in positive bags. Obviously, this all-positive APR is not a good hypothesis. Thus, it is followed by a greedy algorithm to eliminate the negative instances which require eliminating the least number of instances from positive bags. Finally, another greedy procedure is used to select a subset of features (i.e. bounds) which are sufficient to exclude all negative instances.

The outside-in algorithm follows the same procedure as the standard algorithm. But, for eliminating the negative instances it uses a better cost function to eliminate the cheapest negative instances. The new cost function computes the cost of eliminating the positive instances which should be removed after eliminating a negative instance based on the *number* and *density* of the surviving (i.e. remaining) instances inside the bags.

The inside-out algorithm starts by a greedy algorithm to grow an APR that bounds at least one instance from every positive bag. Then it analyzes the APR to select the most discriminative features. Finally, kernel density estimation is used to expand the bounds of the APR so that the probability of having new positive instances fall inside the APR increases. This procedure relaxes the tight bounds constructed in the previous steps and consequently improves generalization.

Diverse Density

Maron et al. [75] proposed diverse density (DD) approach to MIL. The main idea of DD is to find an instance prototype (a.k.a. target point or concept point), which is close to at least one instance of every positive bag but far from instances of all negative bags. This point is obtained by maximizing the diverse density function, which is the likelihood function of training bags, given a Gaussian-like probability distribution over instances. The probability distribution of any instance $\mathbf{x}_{ij}$ is computed based on a weighted distance between the

instance and the target point:

$$p_{ij} = \exp(-\|\mathbf{x}_{ij} - \mathbf{x}\|^2_{\mathbf{w}}) \tag{2.3}$$

Next, the bag probability can be obtained by a soft-max operation, e.g. Noisy-OR (NOR), and incorporated into the likelihood function:

$$DD(\mathbf{x}, \mathbf{w}) = \prod_i \left(\frac{1 + Y_i}{2} - Y_i \prod_j (1 - \exp(-\|\mathbf{x}_{ij} - \mathbf{x}\|^2_{\mathbf{w}})) \right) \tag{2.4}$$

By maximizing the DD function, we get the target point $\mathbf{x}$ and the weight vector $\mathbf{w}$. However, it is a non-convex optimization problem, and a different local maximum could be found with a different initialization. So, one possible modification is to find multiple target points, and compute the instance probabilities by taking max over the target points. To this end, optimization restarts multiple times with instances from positive bags as initial points. This is because the target points are close to positive instances according to the DD definition.

EM-DD [120] is the expectation-maximization (EM) version of above-mentioned DD algorithm. In this version, the soft-max generation of the bag probability is replaced by the exact max operation. So, the likelihood function is no more differentiable. As a result, an EM-like algorithm is proposed to maximize the likelihood. The algorithm iterates over two steps of (1) finding the most probable instance of each positive bag according to the current estimate of classifier and (2) refining the estimate of $\mathbf{x}$ and $\mathbf{w}$ by maximizing the DD function, using all negative instances and the positive instances found in the previous step.

Multi-Instance SVMs

Andrews et al. [6] adapted SVM to the MIL problem, and proposed mi-SVM and MI-SVM algorithms. Both these algorithms are max-margin algorithms, which are formulated as mixed integer optimization problems. The main difference between these two algorithms is how margin is defined in the MIL problem. mi-SVM is based on an *instance margin* formulation, while MI-SVM is based on a *bag margin* formulation of MIL.

mi-SVM aims to maximize the instance margin jointly over the latent instance labels. Thus, it attemps to recover every instance label. For the negative bags, the instance label is known to be negative, but we need to identify the instance labels for positive bags. So,

the max-margin mixed integer optimization of the mi-SVM is defined as follows:

$$\begin{aligned} \min_{y_{ij}} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{ij}\xi_{ij} \\ st. \quad & y_{ij} = -1, \qquad \forall i|Y_i = -1 \\ & \sum_j \frac{y_{ij}+1}{2} \geq 1, \qquad \forall i|Y_i = 1 \\ & y_{ij}(\mathbf{w}.\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0. \end{aligned} \tag{2.5}$$

The constraints impose that instance labels of a negative bag are always negative, but in a positive bag at least one of the instances is positive. This problem is a mixed integer program, which is hard to be solved.

To come up with this problem, Andrews et al. proposed a heuristic algorithm. This algorithm iterates over two steps of inferring integer latent instance labels and continuous optimization of the weight vector. In the first step, given the current classifier, we estimate the instance labels of positive bags. However, if all the instance labels come out as negative, the instance with the largest score (i.e. $\mathbf{w}.\mathbf{x}_{ij} + b$) is labeled positive. In the second step, using the estimated instance labels, we train a standard SVM to classify positive and negative instances. For initialization of this algorithm, all instances of the positive bags are assumed as postie instances in the fist iteration. This algorithm has no convergence guarantee, but it shows good performance in the experimental studies.

MI-SVM aims to maximize the bag margin, where the bag margin is defined by the most postive instance of the bag (a.k.a witness instance):

$$\gamma_i \equiv Y_i \max_j (\mathbf{w}.\mathbf{x}_{ij} + b). \tag{2.6}$$

In fact, this margin is derived from the standard MI assumption. Using this assumption, the bag label is given by the label of the instance with the largest score, i.e., $Y^\star = \operatorname{sign}\max_j (\mathbf{w}.\mathbf{x}_{ij} + b)$. As a result, each bag can be represented by one instance (a.k.a. witness instance), and we try to have large margins for these instances. However, since we usually have no ambiguity in the negative bags, we can unfold the negative bag instances, and define the margin for each negative instance as in regular SVM. Consequently, the maximum bag margin formulation of MIL is given by:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ st. \quad & -\mathbf{w}.\mathbf{x}_{ij} - b \geq 1 - \xi_i, \quad \forall i|Y_i = -1 \\ & \max_j (\mathbf{w}.\mathbf{x}_{ij} + b) \geq 1 - \xi_i, \quad \forall i|Y_i = 1. \end{aligned} \tag{2.7}$$

It can be observed that unlike mi-SVM, where all instances matter in the optimization problem, in MI-SVM only the most positive instance of each positive bag represents the whole bag, and all the other instances have no contribution to the resulting classification boundary. By introducing a selector variable $s(i)$, which indicates the witness instance of each positive bag, the above problem in (2.7) is converted to an equivalent mixed-integer program:

$$\begin{aligned} \min_{s(i)} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ st. \quad & -\mathbf{w}.\mathbf{x}_{ij} - b \geq 1 - \xi_i, \quad \forall i|Y_i = -1 \\ & \mathbf{w}.\mathbf{x}_{is(i)} + b \geq 1 - \xi_i, \quad \forall i|Y_i = 1. \end{aligned} \tag{2.8}$$

Similar to mi-SVM, the mixed integer program of MI-SVM is a hard optimization problem, and Andrews et al. proposed a two-step iterative heuristic algorithm to solve it approximately. At each iteration, first $s(i)$ is guessed by computing the scores of all instances of every positive bag using the current SVM classifier and selecting the instance with the largest score. In the second step, a standard SVM classifier is trained by the selected witness instances and all negative instances in order to update the weight vector. For initialization, each bag is represented by the centroid of the instances inside the bag.

Following the SVM formulations of MIL, Mangasarian and Wild [74] proposed MICA. MICA is an extension of MI-SVM, which does not explicitly identify a witness instance in a bag but finds a convex combination of the instances as a witness. Furthermore, MICA is formulated by L_1 regularization of the weights. Bunescu and Mooney [12] used the transductive SVM framework to propose a modified version of mi-SVM which can more directly enforce the standard multi-instance assumption and perform more effectively for sparse positive bags.

The very successful Latent SVM [32] can be also considered as a MIL method. In Latent SVM, a set of latent variable values is used for positive instances. One can consider the set of completed data instances (latent variable values with observed input feature values) as a "bag" in MIL, similar to the MI-SVM framework.

Deterministic Annealing for MIL (AL-SVM and AW-SVM)

Gehler and Chapelle [40] proposed deterministic annealing versions of mi-SVM and MI-SVM algorithms. In these algorithms, deterministic annealing (DA) is applied to mixed-integer programs of different MIL SVM formulations to find better solutions.

In general, DA searches for a local minimum of a non-convex optimization problem of the form $\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathbb{Z}} F(\mathbf{y})$, where $\mathbf{y}$ is a set of discrete variables. By considering the discrete variables as random variables over a space of probability distributions $\mathcal{P}$, DA tries to find a distribution $\mathbf{p} \in \mathcal{P}$, which minimizes the expected objective function $E_{\mathbf{p}}(F(\mathbf{y}))$.

However, to find a good local minimum (which is close to the global minimum) of this non-convex objective function, the Entropy of the distribution $H(\mathbf{p})$ as a convex term is added to the objective function:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}\in\mathcal{P}} E_{\mathbf{p}}\left(F(\mathbf{y})\right) - T\,H(\mathbf{p}), \tag{2.9}$$

where T indicates the temperature of annealing. The original problem $\arg\min_{\mathbf{p}\in\mathcal{P}} E_{\mathbf{p}}\left(F(\mathbf{y})\right)$ is solved in a sequence of solving the optimization problem in (2.9) by changing the parameter T. The sequence starts by a large value $T_0 \gg 0$ so that the optimization would be inclined to be convex, and gradually the value of T decreases until $T_\infty = 0$, which yields the original problem. Finally, the discrete deterministic varialbes $\mathbf{y}^*$ are identified according to $\mathbf{p}^*$.

In this respect, AL-SVM, uses DA to optimize the mi-SVM objective function in (2.5). The instance labels y_{ij} are regarded as random binary variables with the distributios $P(y_{ij} = 1) = p_{ij}$. Using the DA principles, the optimization problem is converted to optimizing the following DA objective function:

$$\begin{aligned}
\mathcal{L}_T(\mathbf{w}, b, \mathbf{p}) =& \|\mathbf{w}\|^2 + C\sum_i\sum_j p_{ij}\, l\left(\mathbf{w}.\mathbf{x}_{ij} + b\right) + \left(1 - p_{ij}\right) l\left(-\mathbf{w}.\mathbf{x}_{ij} - b\right)\xi_{ij} \\
&+ T\sum_i\sum_j p_{ij}\log p_{ij} + \left(1 - p_{ij}\right)\log\left(1 - p_{ij}\right) \\
st.\quad & 0 \le p_{ij} \le 1, \qquad \forall i, j \\
& \sum_j p_{ij} \ge 1, \qquad \forall i | Y_i = 1.
\end{aligned} \tag{2.10}$$

Note that in this formulation, the last constraint of mi-SVM in (2.5) has been replaced by the hinge loss function l. It is also interesting that by taking the expectation, the MIL constraint $\sum_j \frac{y_{ij}+1}{2} \ge 1$ is translated to $\sum_j p_{ij} \ge 1$. To solve this optimization problem, an iterative coordinate descent algorithm of alternating between estimating the SVM parameters $\{\mathbf{w}^*, b^*\} = \arg\min_{\mathbf{w},b} \mathcal{L}_T(\mathbf{w}, b, \mathbf{p}^*)$ and updating the probability distributions $\mathbf{p}^* = \arg\min_{\mathbf{p}} \mathcal{L}_T(\mathbf{w}^*, b^*, \mathbf{p})$ is proposed. The former is solved by quadratic programming, and the latter is performed by optimizing the dual function of the program.

AW-SVM is the DA version of MI-SVM and MICA. In this algorithm, a probability distribution is defined over each bag and describes the probability of each instance being the witness instance of the bag, i.e., $P\left(\mathbf{x}_{ij} = \text{witness}_i\right) = p_{ij}$. So, we have a new constraint $\sum_j p_{ij} = 1$, and the MI-SVM optimization problem in (2.7) is translated into the following

DA objective function:

$$\begin{aligned}\mathcal{L}_T(\mathbf{w}, b, \mathbf{p}) =& \|\mathbf{w}\|^2 + C\sum_i\sum_j p_{ij}\, l\left(y_i(\mathbf{w}.\mathbf{x}_{ij} + b)\right) + T\sum_i\sum_j p_{ij}\log p_{ij} \\ st.\quad & 0 \le p_{ij}, \quad \forall i, j \\ & \sum_j p_{ij} = 1, \quad \forall i | Y_i = 1.\end{aligned} \tag{2.11}$$

This optimization problem is solved similar to the coordinate descent algorithm of AL-SVM.

MI-Forests

Leistner et al. [63] proposed MI-Forests, which is a multi-instance classification algorithm based on randomized trees. The general approach is to define the labels of instances inside the bags as random variables which are obtained by optimization of a confidence-maximizing loss function over randomized trees, using a fast iterative DA-based method. More specifically, the MIL problem is formulated as an optimization procedure with the following objective function

$$\begin{aligned}(y^\star_{ij}, F^\star) =& \arg\min_{y_{ij}, F(.)} \sum_i\sum_j l(F_{y_{ij}}(\mathbf{x}_{ij})), \\ st.\quad & \text{At least one instance in each bag} \\ & \text{has the same label as the bag label,}\end{aligned} \tag{2.12}$$

where $F_k(x)$ is the classification confidence of the random forest classifier F for assigning the label k to the input x out of the K possible class labels (i.e., $F_k(x) = p(y = k|x) - 1/K$), and $l(.)$ is a loss function which negates the confidence score. Since a random forest should be trained in this optimization, and simultaneously the instance labels (which are integer variables) should be found, this problem is a non-convex combinatorial problem which is usually hard to tackle. However, Leistner et al. proposed a fast iterative algorithm based on deterministic annealing. In this approach, a convex entropy term is added to the objective function, and the iterative procedure initiates with optimization of this term over a space of probability distributions. First, the instance labels in the bags are transformed to random variables defined over a space of probability distributions $\mathcal{P}$. The goal is to optimize the probability distribution $\hat{\mathbf{p}}$ for each bag and train the random forest F at the same time:

$$\begin{aligned}(\mathbf{p}^\star, F^\star) =& \arg\min_{\hat{\mathbf{p}}, F(.)} \sum_i\sum_j\sum_k \hat{p}(k|\mathbf{x}_{ij}) l(F_k(\mathbf{x}_{ij})) + T\sum_i H(\hat{\mathbf{p}}_i), \\ st.\quad & \text{At least one instance in each bag} \\ & \text{has the same label as the bag label,}\end{aligned} \tag{2.13}$$

where H denotes the Entropy function, and T is the temperature parameter of DA. T is set to a large value at the first iteration such that the entropy term dominates the loss function, but it is gradually decreased to zero in order to reach the original optimization problem. To solve the optimization problem in (2.13), the problem is split into a two-step convex optimization problem. First, the distribution $\hat{\mathbf{p}}$ is fixed, and for every tree the labels of the training instances are randomly chosen according to this distribution. Next, a random forest is trained by these instances and the corresponding instance labels. In order to satisfy the MIL constraint, for each bag the label of the instance with the highest probability is set to the bag label. After training the randomized trees, the probability distribution $\hat{\mathbf{p}}$ is optimized with respect to the objective function in (2.13), given the random forest fixed. This is a convex optimization problem, which is solved by taking the derivatives and setting to zero.

In addition, Leistner et al. [63] extended this algorithm to an on-line procedure, where the bags arrive sequentially. To this end, bagging (in random forest training) is performed on-line by modeling the arriving samples with a Poisson distribution. Also, the decision trees are trained on-line, and the probability distribution in (2.13) is updated sequentially.

The advantages of using random forests in the proposed algorithm is that they are fast in both training and test, and also they are inherently parallel and multi-class.

MILBoost

Viola et al. [99] proposed MILBoost, a boosting algorithm adapted for MIL. Boosting is defined as a general algorithm to make an accurate prediction rule by combining some rough and week rules. The general idea is that the boosting algorithm repeatedly calls a weak or base classifier and trains it with a different distribution over training samples. Finally, the strong classifier is given as weighted sum of the weak classifiers:

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}), \tag{2.14}$$

where h_t and α_t denote a weak classifier and a positive weight at step t, respectively. One approach of finding h_t and α_t is to employ AnyBoost framework [76]. AnyBoost tries to minimize a cost function $\mathcal{L}(H)$ with respect to H in a functional space by performing gradient descent. At each step of boosting, first a weight w_i is computed for each data example $\mathbf{x}_i$: $w_i = \frac{\partial \mathcal{L}}{\partial H}|_{\mathbf{x}=\mathbf{x}_i}$. Next, h_t and α_t are obtained by solving the following two optimization problems:

$$h_t = \arg\max_{h} \sum_{i} w_i h(\mathbf{x}_i), \tag{2.15}$$

$$\alpha_t = \arg\min_{\alpha} \mathcal{L}(H_{t-1} + \alpha h_t). \tag{2.16}$$

The proposed MILBoost uses AnyBoost framework to minimize the negative log likelihood of the training bags:

$$\mathcal{L}(H) = \sum_i \mathbb{1}(Y_i = 1) \log p(\mathbf{X}_i) + \mathbb{1}(Y_i = -1) \log \left(1 - p(\mathbf{X}_i)\right), \tag{2.17}$$

where $p(\mathbf{X}_i)$ is the probability of the ith bag being positive and expressed in terms of its instance probabilities by the Noisy-OR (NOR) model:

$$p(\mathbf{X}_i) = 1 - \prod_{\mathbf{x}_{ij} \in \mathbf{X}_i} \left(1 - p(\mathbf{x}_{ij})\right), \tag{2.18}$$

where $p(\mathbf{x}_{ij}) = \frac{1}{1+\exp\left(-H(\mathbf{x}_{ij})\right)}$ denotes the probability of $\mathbf{x}_{ij}$ being positive. The rationale for the NOR model is that the probability of a bag being positive is high if at least one of the instances has high positive probability. Viola et al. [99] used MILBoost in a cascade detection procedure for object detection.

2.1.2 Methods Encoding Non-Standard MI Assumptions

In recent years, more general MIL algorithms have been developed to address non-standard multi-instance assumptions.

ALP-SVM

ALP-SVM [40] is an extension to the AL-SVM and AW-SVM methods explained before. In fact, the experiments in [40] showed that although AL-SVM and AW-SVM find better local minima compared to mi-SVM and MI-SVM, but they do not always yield better test error. This suggests that probably the objective functions do not adequately describe the problem. Especially, it was observed that AL-SVM algorithm underestimates the number of positive instances in the positve bags. To come up with this problem, Gehler and Chapelle [40] proposed a new objective function, which controls the expected number of positive instances in the bags. This function is obtained by extending the mi-SVM objective function as follows:

$$\mathcal{L}'_T\left(\mathbf{w}, b, y_{ij}, \xi_{ij}\right) = \mathcal{L}_T\left(\mathbf{w}, b, y_{ij}, \xi_{ij}\right) + C_2 \sum_i \left(\sum_j \frac{y_{ij}+1}{2} - m_i p_i^* \right)^2, \tag{2.19}$$

where p_i^* determines the ratio of positive labeled instances in a bag, and m_i denotes the total number of instances in the bag. In fact the new additional term penalizes deviation of the ratio of positive labeled instances in the bag from p_i^*. Note that for all positive bags p_i^* is set to the same prefixed value (which can be estimated by cross-validation), and for all negative bags it is set to 0. This objective function can be optimized by deterministic annealing, similar to AL-SVM. In this case, after taking the expectation, the new term

is translated into $C_2 \sum_i \left(\sum_j p_{ij} - m_i p_i^*\right)^2$. The experimental results show that ALP-SVM outperforms the other SVM formulations of MIL in most cases.

MIL-CPB

Li et al. [67] proposed MIL-CPB, an algorithm for multi-instance learning with constrained positive bags. This model extends the MI assumption of positive bags from "at least one instance is labeled positive" to "at least a portion of instances are labeled positive" (i.e., ratio-constrained MI assumption). The formulation of MIL-CPB is similar to other MIL SVM formulations but with squared bias penalty and squared hinge loss function:

$$\begin{aligned} \min_{y_{ij},\mathbf{w},b,\rho,\xi_{ij}} \quad & \frac{1}{2}\left(\|\mathbf{w}\|^2 + b^2 + C\sum_{ij}\xi_{ij}^2\right) - \rho \\ st. \quad & y_{ij} = -1, \quad \forall i | Y_i = -1 \\ & \sum_j \frac{y_{ij}+1}{2} \geq \sigma |\mathbf{X}_i|, \quad \forall i | Y_i = 1 \\ & y_{ij}(\mathbf{w}.\phi(\mathbf{x}_{ij}) + b) \geq \rho - \xi_{ij}, \xi_{ij} \geq 0, \quad \forall i, j, \end{aligned} \tag{2.20}$$

where σ specifies the least proportion of positive instances in a positive bag and $\phi(\mathbf{x})$ is a mapping function which maps $\mathbf{x}$ into another probably higher dimensional space. The dual form of this optimization is written as

$$\min_{\mathbf{y}} \max_{\alpha \in \mathcal{A}} -\frac{1}{2}\alpha' \left(\tilde{\mathbf{K}} \odot \mathbf{y}\mathbf{y}' + \frac{1}{C}\mathbf{I}\right)\alpha, \tag{2.21}$$

where $\mathbf{y} = [y_1, \cdots, y_n]$ is a vector constructed by concatenating all instance labels of all training bags, $\alpha = [\alpha_1, \cdots, \alpha_n]$ is the vector of dual variables, $\mathcal{A} = \{\alpha | \alpha_i \geq 0, \sum_i \alpha_i = 1\}$ denotes the feasible set of α, $\mathbf{I}$ is the identity matrix, and $\tilde{\mathbf{K}} = \mathbf{K} + \mathbf{1}_{n\times n}$ where $\mathbf{K}$ is the kernel matrix corresponding the mapping function $\phi(.)$. This is a hard mixed integer programming problem. However, it is proved in [30] that the lower bound of the objective value of this problem is the optimal objective value of the following problem:

$$\min_{\mathbf{d}\in\mathcal{D}} \max_{\alpha \in \mathcal{A}} -\frac{1}{2}\alpha' \left(\sum_{\mathbf{y}^t \in \mathcal{Y}} d_t \, \tilde{\mathbf{K}} \odot \mathbf{y}^t\mathbf{y}^{t'} + \frac{1}{C}\mathbf{I}\right)\alpha, \tag{2.22}$$

where $\mathbf{d}$ is a vector of new dual variables (d_ts) with the feasible space of $\mathcal{D} = \{\mathbf{d} | d_t \geq 0, \sum_t d_t = 1\}$ and $\mathbf{y}^t$ is any solution in the set of all feasible solutions $\mathcal{Y}$. This problem can be assumed as a multiple kernel learning (MKL) optimization problem, with the base kernels $\tilde{\mathbf{K}} \odot \mathbf{y}^t\mathbf{y}^{t'}, \forall \mathbf{y}^t \in \mathcal{Y}$. However, $|\mathcal{Y}|$ and consequently the number of kernels is exponential in size, and learning with standard MKL solvers such as SimpleMKL [84] is not tractable. To

come up with this problem, Li et al. proposed a cutting-plane algorithm to find a subset $\mathcal{C} \subset \mathcal{Y}$ of feasible solutions which can adequately approximate the original problem. It is an iterative algorithms that alternates between (1) optimizing α and $\mathbf{d}$ using SimpleMKL given the current approximate set $\mathcal{C}$ and (2) adding a new $\mathbf{y}^t$ to $\mathcal{C}$ by enumerating all possible labeling candidates of the instances in the positive bags. Note that this algorithm can be computationally expensive if the bags contain many instances.

$\propto$SVM [118] is a similar algorithm for learning from instance labels proportions. This method follows a SVM-based formulation, where the instance labels are modeled as latent variables with constraints on the positive label proportion. Yu et al. [118] proposed two algorithms to learn the model: (1) alternating optimization of the mixed-integer programming problem and (2) convex relaxation of the objective function.

Despite successful results of the algorithms above, almost all of them use some kind of heuristics or relaxation and consequently provide approximate solutions to the general problem of multi-instance learning based on label proportions and lack solid mathematical proof of convergence. In addition, they are limited to specific cardinality assumptions (e.g. ratio-based assumptions) and to capture new cardinality relations between the instance labels the proposed models or learning algorithms should be modified.

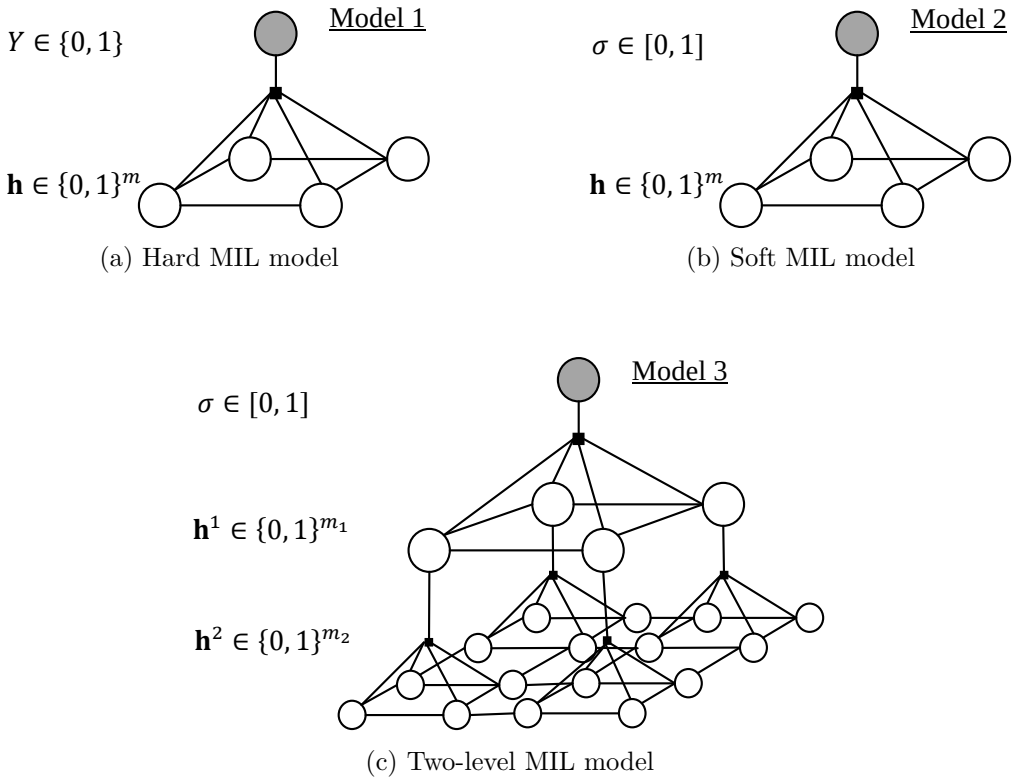
2.1.3 Methods based on Probabilistic Graphical Models

Probabilistic graphical models (PGMs) are powerful tools to capture inter-relations between random variables and learn structured models. Thus, they can be assumed a natural fit to model multi-instance problems. Note that although we have categorized PGM-based methods as instance-space methods, but these methods also fall close to the boundaries of bag-space methods. In fact, in PGMs, both instance-level local information and bag-level global information can be modeled and mixed. However, since the base of these models is built on the instances, and the first-level discrimination lies in the instance space, we think that PGM-based MIL methods are mostly (not always) closer to instance-space methods.

MIL with Structured Bag Models

Warrel and Torr [109] developed an algorithm for multi-instance learning with structured bag models. These models can capture spatial relations or interactions between instances of a bag and alleviate the assumption that instances of a bag are independent. To this end, Warrel et al. use CRFs to model the bag structures and at the same time incorprate the MIL constraints.

Three structured bag models have been studied in this work. The first model shown in Figure 2.1a considers unary and pairwise potentials on the instances and a hard MIL constraint on the instance labels. The hard MIL constraint embeds the standard MI assumption that a bag is positive if at least one of the instnaces is positive. In this model,



(a) Hard MIL model

(b) Soft MIL model

(c) Two-level MIL model

Figure 2.1: Graphical representation of the structured bag models in [109].

$\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ denotes the bag, $\mathbf{h} = \{h_1, \cdots, h_m\}$ denotes the collection of instance labels and Y is the bag label. The energy function for this bag model is defined as

$$E^1(\mathbf{h}, Y|\mathbf{X}) = E_{\text{base}}(\mathbf{h}|\mathbf{X}) + \phi^{\text{hard-MIL}}(\mathbf{h}, Y), \tag{2.23}$$

where the base energy function is given by

$$E_{\text{base}}(\mathbf{h}|\mathbf{X}) = \sum_j \phi^{\text{unary}}(\mathbf{x}_j, h_j) + \sum_{j1,j2\in\mathcal{N}} \phi^{\text{pair}}(h_{j1}, h_{j2}), \tag{2.24}$$

and the MIL constraint is described by

$$\phi^{\text{hard-MIL}}(\mathbf{h}, Y) = \begin{cases} 0 & \text{if } \max_j h_j = Y \\ \infty & o.w. \end{cases} \tag{2.25}$$

The second model shown in Figure 2.1b is similar to the first model except that a soft and more general MIL constraint is incorporated in the model. This constraint controls the ratio of positive instances by introducing continuous bag labels $\sigma \in [0, 1]$. The energy of

this model is characterized by

$$E^2(\mathbf{h}, \sigma|\mathbf{X}) = E_{\text{base}}(\mathbf{h}|\mathbf{X}) + \phi^{\text{soft-MIL}}(\mathbf{h}, \sigma), \tag{2.26}$$

where the soft MIL constraint

$$\phi^{\text{soft-MIL}}(\mathbf{h}, \sigma) = g(\sigma.m - \sum_j h_j), \tag{2.27}$$

is defined based on a distance function $g(.)$ (e.g. l_1-distance or l_2-distance), which penalizes the deviation from the expected number of positive instances.

The third structured model shown in Figure 2.1c is a combination of the two models above. It is a two-level model. At the top, there is a continuous bag label σ, which interacts with the binary instance labels in the first level $\mathbf{h}^1$ according to the soft MIL constraint in (2.27). At the same time, the first level labels $\mathbf{h}^1$ form a set of hard sub-bag labels for a subset of instances in the second level. This model can be used in applications like semi-supervised multi-level segmentation. For example, at the first level we have the patch labels and in the second finer level we have the pixel labels. The energy function for this bag model is given by

$$\begin{aligned} E^3(\mathbf{h}^1, \mathbf{h}^1, \sigma|\mathbf{X}^1, \mathbf{X}^2) =& E_{\text{base}}(\mathbf{h}^1|\mathbf{X}^1) + E_{\text{base}}(\mathbf{h}^2|\mathbf{X}^2) \\ &+ \phi^{\text{soft-MIL}}(\mathbf{h}^1, \sigma) + \sum_j \phi^{\text{hard-MIL}}(\mathbf{h}^2_j, h^1_j), \end{aligned} \tag{2.28}$$

where $\mathbf{h}^2_j$ is the subset of instance labels in the second level which is associated with the instance label h^1_j in the first level.

To use these models for MIL, we should solve the following two problems.

- **Inference**: The inference problem is to minimize the energy function of the bag models:

$$(\mathbf{h}^\star, Y^\star) = \arg\min_{\mathbf{h}, Y} E \tag{2.29}$$

  Inference of the first model is simple. If the bag label y is not observed, the instance labels are found by $\mathbf{h}^\star = \arg\min_{\mathbf{h}, y} E_{\text{base}}$ using graph-cut. Next, the bag label is inferred as $Y^\star = \max_j x^\star_j$. However, if the bag label is known, inferring the instance labels is a bit more tricky. For the case $Y = 0$ inference is still simple and $\mathbf{h}^\star = \mathbf{0}$. Also, if $Y = 1$ and the graph-cut on E_{base} returns at least one positive instance label, there is no problem. However, if $Y = 1$ and graph-cut on E_{base} returns $\mathbf{h}^\star = \mathbf{0}$, then we should perform graph-cut m times by forcing one of the instances be positive at each time and take the solution with the lowest energy. Inference of the second and third models is harder and performed approximately by dual decomposition.

- **Learning**: For learning these models, deterministic annealing (DA) is used to minimize an objective function similar to negative log-likelihood:

$$(\theta^\star, \mathbf{h}_i^\star) = \arg\min_{\theta, \mathbf{h}_i} \sum_i -\log\left(P(\mathbf{h}_i, Y_i|\mathbf{X}_i, \theta)\right), \tag{2.30}$$

where θ represents the set of model parameters. Using DA approach, the objective function is transformed to

$$(\theta^\star, \pi^\star) = \arg\min_{\theta, \pi_i} \sum_i \sum_{\mathbf{h}_i} -\pi_i \log\left(P(\mathbf{h}_i, Y_i|\mathbf{X}_i, \theta)\right) - T \sum_i H(\pi_i), \tag{2.31}$$

where $\pi_i \equiv \pi(\mathbf{h}_i)$ introduces a probability distribution over the instance labels of a bag and H is the entropy function. Following the DA approach, optimization starts with a large T and it is reduced gradually, while for each T, $\theta^\star$ and $\pi^\star$ are found using an iterative algorithm of alternating between optimizing θ and π given the other one fixed. However, these optimization problems are very hard to be solved or intractable if the actual CRF probability distribution $P(\mathbf{h}_i, Y_i|\mathbf{X}_i, \theta) = \exp(-E)/Z$ is used. The reason is that estimation of the partition function Z is NP-hard, and also each distribution π_i is growing exponentially with the bag size. To come up with a tractable solution, Warrel and Torr proposed to replace the CRF joint probability distribution by $P'(\mathbf{h}_i, Y_i|\mathbf{X}_i, \theta) = P'(Y_i|\mathbf{X}_i, \theta)P'(\mathbf{h}_i|Y_i, \mathbf{X}_i, \theta)$. It is shown that the new factorization and proper parameterization of $P'(Y_i|\mathbf{X}_i, \theta)$ and $P'(\mathbf{h}_i|Y_i, \mathbf{X}_i, \theta)$ can solve the both above-mentioned problems. Especially, using this probability distribution causes the instances become independent given the model parameters θ and the bag labels. So, the optimal distribution over instance labels factorizes as $\pi_i^\star = \prod_j \pi_{ij}^\star$, and consequently each $\pi_{ij}^\star$ can be found independently by solving a series of convex optimization problems in parallel. Note that for the soft labels σ_i in the second and third Models, a similar learning procedure is used after quantizing σ into a number of levels, which acts as hard labels for the bags.

MI-CRF

Deselaers and Ferrari [23] 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. So, the bags are jointly trained and classified in this model. The model uses instance classifiers as unary terms and dissimilarity measures between the witness instances as pairwise terms. Thus, bag classification can be improved by using other MIL-based instance classifiers and integrating information from all bags.

Set Restricted Boltzmann Machines

Louradour and Larochelle [71] proposed extensions of restricted Boltzmann machines (RBMs) for classifying sets. This method can be also applied to MIL data. A standard RBM consists of two layers. One layer of observations (visible layer) and one layer of hidden units (hidden). In the proposed method, RBMs are extended by duplicating the visible and hidden layers for each instance. The basic idea is to encode the bag label besides the instance input vectors in the visible layer and adding constraints on the hidden layer. At the test time, the predictions is performed by comparing the likelihood of all possible labels.

Generative Graphical Models for MIL

Adel et al. [1] proposed a general framework to use generative graphical models in the MIL paradigm. This framework studies and analyzes different Bayes net structures for MIL. For example, in one structure the bag label generates all instance labels and then each instance label generate the instance feature vector independently. In another structure, this generation flow is reversed. For training, an expectation-maximization algorithm is used, which alternates between estimating the model paramters and inferring the instance labels.

2.2 Bag-Space Methods

Bag-space methods treat each bag as a whole entity and train a classifier directly on the bags by making a global representation of bags or extracting discriminative bag-level information from them. In this section, we briefly explain these methods, classified in three main subcategories: “embedded-space” methods, "kernel-based" methods, and "distance-based" methods.

2.2.1 Embedded-Space Methods

The methods described in this section transform MIL problem to a standard classification problem by mapping the bags into an embedded single-instance space. First, each bag is mapped to a single feature vector by a mapping function. Next, a single-instance classifier is trained in the embedded space.

Simple MI

Simple MI [29] is a very simple and fast algorithm. In this algorithm, each bag is mapped to the average of its instances. The averaging can be performed by arithmetic mean or geometric mean. Then, any standard single-instance classifier can be trained for bag classification. Although this algorithm is very simple, but surprisingly it has shown successful

results in some MIL data set (e.g. data sets which have positive bags with many positive instances, i.e., have less instance-label ambiguity).

Histogram-Based Methods

Histogram-Based Methods [5] works similar to bag-of-words (BOW) methods by mapping each bag to a histogram vector, using a vocabulary. First, a vocabulary of concepts (or words) is obtained by hard or soft clustering of all instances in the training bags. Next, each bag $\mathbf{X}$ is mapped to a histogram vector $\mathbf{v} = (v_1, \cdots, v_K)$ with

$$v_k = \frac{1}{Z} \sum_{\mathbf{x}_j \in \mathbf{X}} f_k(\mathbf{x}_j), \tag{2.32}$$

where $f_k(\mathbf{x}_j) \in [0, 1]$ is a function which specifies membership of the instance $\mathbf{x}_j$ in the kth concept of the vocabulary according to hard or soft-assignment of the clustering algorithm. In the hard-assignment, v_k counts the number of instances of the bag, which are assigned to the kth concept. However, in the soft-assignment v_k is the sum of membership value of all the instances in that concept. Z is a factor which normalizes the histogram vector such that $\sum_k v_k = 1$.

Note that (2.32) can be simply modified to model some new MI assumptions such as *presence-based* assumption, *threshold-based*, and *count-based* assumption [33]. For presence-based assumption, we can replace the sum in (2.32) with a max function. Thus, a concept is said to be positive if at least one instance of the bag is present in that concept. For the threshold-based assumption, the sum is replaced by a threshold function. This means that a concept is positive if at least a certain number of instances belong to that concept. A similar modification takes place for the count-based assumption.

DD-SVM and MILES

DD-SVM [19] and MILES [18] are two algorithms, which combine diverse density (DD) approach and SVM classification. Both these algorithms use the notion of target points (instance prototypes) introduced in the DD paradigm (Section 2.1.1) to embed the bags into a new single-instance feature space. So, any single-instance classifier (e.g. SVM here) can be used to classify positive and negative bags.

First, given the target points $(\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^t)$, each bag $\mathbf{X}_i$ is converted to a vector as follows:

$$\phi(\mathbf{X}_i) = \begin{bmatrix} s\left(\mathbf{x}^1, \mathbf{X}_i\right) \\ s\left(\mathbf{x}^2, \mathbf{X}_i\right) \\ \vdots \\ s\left(\mathbf{x}^t, \mathbf{X}_i\right) \end{bmatrix}, \tag{2.33}$$

where $s(\mathbf{x}, \mathbf{X}_i) = \min_j \|\mathbf{x}_{ij} - \mathbf{x}\|^2_{\mathbf{w}}$ in DD-SVM, and $s(\mathbf{x}, \mathbf{X}_i) = \max_j \exp\left(-\frac{\|\mathbf{x}_{ij}-\mathbf{x}\|^2_{\mathbf{w}}}{\sigma^2}\right)$ in MILES. Using the embedded bag-level feature vectors, a standard L_2-norm SVM classifier is trained in DD-SVM. However, in MILES, an L_1-norm SVM is used. L_1 regularization leads to an SVM with a sparse weight vector. So, L_1-norm SVM can be employed for both feature selection and classification. As a result, Chen et al. [18] proposed to construct the vector in (2.33) by all the instances of the positive bags as target points, and let the L_1-norm SVM choose the most effective instances. They reported that this approach improves final classification accuracy although the computational cost is reduced.

2.2.2 Kernel-Based Methods

Kernel-Based Methods work by defining kernels on the bags. As a result, any standard kernel machine can be used for classification. Note that kernel-based methods also works by performing an implicit space transformation and mapping. Thus, it might be also possible to categorize kernel-based methods as a type of embedded-space methods. However, to have more clear and focused presentation, we dedicate a distinct section for these methods.

Multi-Instance Kernels

Gartner et al. [39] introduced a class of multi-instance (MI) kernels, which are defined directly on the bags. In this approach MIL problems can be solved by plugging the proposed MI kernels into SVM or other kernel machines. These kernels are inspired by the set kernels in [38] and [51]. Given two sets $\mathbf{X}$, $\mathbf{X}'$, a set kernel k_{set} is defined by

$$k_{set}(\mathbf{X}, \mathbf{X}') := \sum_{\mathbf{x}\in\mathbf{X}, \mathbf{x}'\in\mathbf{X}'} k(\mathbf{x}, \mathbf{x}'). \tag{2.34}$$

It is proved that k_{set} is a kernel (i.e., has all properties of a kernel) if and only if k is a kernel. Based on the set kernel, Gartner et al. proposed an MI kernel

$$k_{MI}(\mathbf{X}, \mathbf{X}') := \sum_{\mathbf{x}\in\mathbf{X}, \mathbf{x}'\in\mathbf{X}'} k_I^p(\mathbf{x}, \mathbf{x}'), \tag{2.35}$$

where $\mathbf{X}$, $\mathbf{X}'$ are two bags and k_I^p is an instance-level kernel k_I raised to the power of p. Since products of kernels are kernels, k_I^p is also a kernel, and consequently k_{MI} is a variant of the set kernel in (2.34). Gartner et al. showed that for sufficiently large p, k_{MI} can separate a standard MI concept[1] into positive and negative sets if and only if the underlying instance-level concept is separable by k_I. Note that k_{set} in (2.34) and k_{MI} in (2.35) are biased

[1] A standard MI concept ν_{MI} is defined based on the standard MI assumption on an underlying instance-level concept ν_I, i.e., a bag is positive if and only if there is at least one positive instance in the bag.

towards the bags with large cardinality. So, in practice these kernels are normalized, e.g.,

$$k_{MI}\left(\mathbf{X},\mathbf{X}'\right) := \frac{k_{MI}\left(\mathbf{X},\mathbf{X}'\right)}{\sqrt{k_{MI}\left(\mathbf{X},\mathbf{X}\right)}\sqrt{k_{MI}\left(\mathbf{X}',\mathbf{X}'\right)}}. \tag{2.36}$$

An interesting point about the MI kernel in (2.35) is that it can be rewritten as

$$k_{MI}\left(\mathbf{X},\mathbf{X}'\right) := \sum_{\mathbf{x}\in\mathbf{X},\mathbf{x}'\in\mathbf{X}'} \phi_I(\mathbf{x})\phi_I(\mathbf{x}') = \left(\sum_{\mathbf{x}\in\mathbf{X}} \phi_I(\mathbf{x})\right)\left(\sum_{\mathbf{x}'\in\mathbf{X}'} \phi_I(\mathbf{x}')\right), \tag{2.37}$$

where $\phi_I(\mathbf{x})$ is the underlying feature mapping function of the kernel k_I^p. Thus, it can be observed that SVM with k_{MI} corresponds to representing each bag by sum of its instances in the underlying feature space and applying a standard linear SVM. This shows that the proposed MI kernel assumes equal weights on all instances of a bag. However, we know that in positive bags all the instances are not equally important. To alleviate this problem, later, Kwok and Cheung [58] proposed marginalized MI kernels. These kernels specify the importance of an instance pair from two bags according to the consistency of their probabilistic instance labels.

For large bags, the computational complexity of k_{MI} in (2.35) is high. Hence, Gartner et al. also introduced a simple and more efficient class of kernel functions. The new kernel, called statistic kernel k_{stat}, uses the statistical properties of the bags to summarize and map the bags into vectors, which can be compared by standard single-instance kernels:

$$k_{stat}\left(\mathbf{X},\mathbf{X}'\right) := k_I\left(s(\mathbf{X}), s(\mathbf{X}')\right), \tag{2.38}$$

where $s(\mathbf{X})$ is a mapping which collects some statistics of the set X such as mean, median, minimum, maximum, etc. An example of this class of kernels is the *minmax kernel*, which is defined based on the following mapping

$$s(\mathbf{X}) = \left(\min_{\mathbf{x}\in\mathbf{X}} x_1, \cdots, \min_{\mathbf{x}\in\mathbf{X}} x_d, \max_{\mathbf{x}\in\mathbf{X}} x_1, \cdots, \max_{\mathbf{x}\in\mathbf{X}} x_d\right) \tag{2.39}$$

and has been shown to be very successful in some MIL problems (e.g. drug activity prediction).

MIGraph and miGraph

Zhou et al. [121] proposed two graph-based algorithms, MIGraph and miGraph, for multi-instance learning. Both algorithms work by mapping a bag into an undirected graph and designing a graph kernel. So, the classification problem can be solved by any kernel machine, e.g. SVM.

MIGraph constructs a weighted ϵ-graph for every bag. In this graph, each instance is modeled as a node, and every two nodes are connected if the Euclidean distance between

the two instances is less than a preset threshold ϵ. The weight of each edge is defined by the normalized reciprocal of non-zero distance of the the two nodes connecting, and is a notion of affinity between them. Then, given two graphs (i.e. two bags $\mathbf{X}_i$ and $\mathbf{X}_j$), a kernel function is defined as follows:

$$\begin{aligned} k_G(\mathbf{X}_i, \mathbf{X}_j) = &\sum_{a=1}^{n_i}\sum_{b=1}^{n_j} k_{node}\left(\mathbf{x}_{ia}, \mathbf{x}_{jb}\right) \\ &+ \sum_{a=1}^{m_i}\sum_{b=1}^{m_j} k_{edge}\left(\mathbf{e}_{ia}, \mathbf{e}_{jb}\right), \end{aligned} \tag{2.40}$$

and normalized to

$$k_G(\mathbf{X}_i, \mathbf{X}_j) = \frac{k_G(\mathbf{X}_i, \mathbf{X}_j)}{\sqrt{k_G(\mathbf{X}_i, \mathbf{X}_i)}\sqrt{k_G(\mathbf{X}_j, \mathbf{X}_j)}}, \tag{2.41}$$

where n_i and m_i indicates the number of nodes and edges in a graph, respectively. The node kernel k_{node} and the edge kernel k_{edge} can be defined as any positive semidefinite single-instance kernels such as Gaussian RBF kernel. However, k_{edge} is more tricky because we need to define a feature vector (i.e. $\mathbf{e}_{ia}$) for each edge of the graph. Zhou et al. define the edge feature vector between the nodes $\mathbf{x}_{iu}$ and $\mathbf{x}_{iv}$ as $[d_u, p_u, d_v, p_v]$, where d_u denotes the degree of the node $\mathbf{x}_{iu}$, i.e., the number of edges connected to that. p_u shows the importance of connection to the node $\mathbf{x}_{iv}$ for the node $\mathbf{x}_{iu}$ and defined as $p_u = w_{uv}/\sum w_{u,*}$, where w_{uv} is the weight of the edge between $\mathbf{x}_{iu}$ and $\mathbf{x}_{iv}$. Likewise, d_v and p_v are defined for the node $\mathbf{x}_{iv}$.

The complexity of computing the kernel $k_G(\mathbf{X}_i, \mathbf{X}_j)$ in MIGraph is $O\left(n_i n_j + m_i m_j\right)$, which is costly due to the large number of edges usually existing in the constructed graph. Thus, Zhou et al. propose miGraph, which is more computationally efficient. miGraph implicitly maps a bag to a graph by constructing the affinity matrix (W^i) of the graph. First a Gaussian distance is computed between every two pairs of instances (nodes). By comparing the distance of each pair (e.g. $\mathbf{x}_{ia}$ and $\mathbf{x}_{iu}$) with a threshold δ, the correspond element of the affinity matrix (e.g. w^i_{au}) is set to 1 or 0. Next the graph kernel is defined as

$$k_g(\mathbf{X}_i, \mathbf{X}_j) = \frac{\sum_{a=1}^{n_i}\sum_{b=1}^{n_j} W_{ia} W_{jb} k\left(\mathbf{x}_{ia}, \mathbf{x}_{jb}\right)}{\sum_{a=1}^{n_i} W_{ia} \sum_{b=1}^{n_j} W_{jb}}, \tag{2.42}$$

where $W_{ia} = 1/\sum_{u=1}^{n_i} w^i_{au}$, $W_{jb} = 1/\sum_{v=1}^{n_j} w^j_{bv}$, and $k\left(\mathbf{x}_{ia}, \mathbf{x}_{jb}\right)$ is a positive semidefinite instance kernel. The intuition behind this kernel is that the instances which form a clique in the graph are treated fairly equally and consequently k_g implies a notion of soft clique-based graph kernel. The computational complexity of this kernel is $O(n_i n_j)$, which is independent of the number of edges.

2.2.3 Distance-Based Methods

A class of MIL algorithms uses distance metrics to classify bags. The distance can be a bag-to-bag (B2B) distance or a class-to-bag (C2B) distance. Also, the distance metric can be fixed or learned from training data.

Citation kNN

Wang and Zucker [107] proposed Citation kNN, which modifies the popular kNN classification to MIL. First the B2B distance between each pair of bags is computed by the *minimum Hausdorff* distance:

$$D(\mathbf{X}_i, \mathbf{X}_{i'}) = \min_{\mathbf{x}_{ij} \in \mathbf{X}_i} \min_{\mathbf{x}_{i'j'} \in \mathbf{X}_{i'}} \|\mathbf{x}_{ij} - \mathbf{x}_{i'j'}\|. \tag{2.43}$$

Next, following the typical kNN appraoch, the class label of an unlabeled bag can be precidted by majority voting among bag label of its nearest neighbors. However, Wang and Zucker argued that this approach does not achieve satisfactory results in the multi-instance paradigm. The reason is the large number of negatives in the positive bags, which can mislead class prediction based on minimum Hausdorff distance. To alleviate this problem, Wang and Zucker [107] proposed to classify a bag $\mathbf{X}$ by majority voting among both "references" and "citers" of $\mathbf{X}$. In this setting, references of $\mathbf{X}$ are the nearest neighbors of $\mathbf{X}$ (the same as previous), but citers of $\mathbf{X}$ are those bags which count $\mathbf{X}$ as one of their neighbors. The experimental results have shown that this approach improves the robustness against outlier instances and increases classification accuracy.

Metrics Enhanced Class-to-Bag Distance (M-C2B)

Wang et al. [104] proposed an algorithm to learn a robust and discriminative class-to-bag (C2B) distance for MIL. Unlike the MI distances defined in the similar previous works (e.g., [102, 42, 103]), the proposed distance is based on not-squared l_2-norm distance. It is well-known that not-squared l_2-norm distance is robust againts outliers [77], which makes it suitable for MI data, where the outlier instances abound because of label ambiguity in positive bags.

This algorithm can be used for multi-class multi-label MIL. So, each bag label is denoted by a vector $Y_i = [Y_{i1}, \cdots, Y_{iK}]$, where $Y_{ik} = 1$ if bag $\mathbf{X}_i$ belongs to the kth class. According to standard multi-class MI assumption, a bag is assigned to the kth class if at least one of the instances is from the kth class. Given the training bags $\mathcal{X} = \{\mathbf{X}_i, Y_i\}_{i=1}^N$, the M-C2B distance is proposed as follows. First each class is represented as a super-bag, collecting all instances of the training bags: $C_k = \{\mathbf{x}_{ij} | i \in \pi_k\}$, where $\pi_k = \{i | Y_{ik} = 1\}$. Then, given an elementary instance-to-bag distance

$$d_k(\mathbf{x}_{ij}, \mathbf{X}_{i'}) = \|\mathbf{x}_{ij} - \mathcal{N}_{i'}(\mathbf{x}_{ij})\|, \tag{2.44}$$

which is described as the distance from the instance $\mathbf{x}_{ij}$ to its nearest neighbor in the bag $\mathbf{X}_{i'}$ (i.e., $\mathcal{N}_{i'}(\mathbf{x}_{ij})$), the class-to-bag distance is defined as

$$D(C_k, \mathbf{X}_{i'}) = \sum_{\mathbf{x}_{ij} \in C_k} d_k(\mathbf{x}_{ij}, \mathbf{X}_{i'}). \tag{2.45}$$

However, instead of using Euclidean distance in (2.44), Wang et al. proposed to use Mahalanobis distance with a class-specific distance metric M_k in order to model the second-order interactions between input feature vectors. Thus, the proposed M-C2B distance is rewritten as

$$D(C_k, \mathbf{X}_{i'}) = \sum_{\mathbf{x}_{ij} \in C_k} \sqrt{[\mathbf{x}_{ij} - \mathcal{N}_{i'}(\mathbf{x}_{ij})]^\top M_k [\mathbf{x}_{ij} - \mathcal{N}_{i'}(\mathbf{x}_{ij})]}. \tag{2.46}$$

For learning the distance metrics M_k, the following optimization problem should be solved for each class:

$$\min_{M_k} \frac{\sum_{i' \in \pi_k} D(C_k, X_{i'})}{\sum_{i' \notin \pi_k} D(C_k, X_{i'})} \tag{2.47}$$

This problem tries to minimize the M-C2B distance from a class to all its bags, while maximizing the distance to all bags of the other classes. If the size of feature vector is large (which is the case in most MI data sets), the cardinality of $M_k \in \mathbb{R}^{d \times d}$ is very high and the search space is huge. A popular trick to compress the problem is to take advantage of the semi-definite property of M_k and decompose it as $M_k = U_k U_k^\top$, where $U_k \in \mathbb{R}^{d \times r}$ and $r \ll d$. Finally, the optimization problem takes the form

$$\min_{U_k} \frac{\sum_{i' \in \pi_k} \|[\mathbf{x}_{ij} - \mathcal{N}_{i'}(\mathbf{x}_{ij})]^\top U_k\|_2}{\sum_{i' \notin \pi_k} \|[\mathbf{x}_{ij} - \mathcal{N}_{i'}(\mathbf{x}_{ij})]^\top U_k\|_2}. \tag{2.48}$$

It can be shown that the optimization in (2.48) can be rewritten in a vectroized compact general form as

$$\min_{U_k} \frac{\|\mathbf{A}_k U_k\|_{2,1}}{\|\mathbf{B}_k U_k\|_{2,1}} = \frac{\sum_p \|\mathbf{a}_k^p U_k\|_2}{\sum_p \|\mathbf{b}_k^p U_k\|_2}, \tag{2.49}$$

where $\|\mathbf{M}\|_{2,1}$ indicates the $l_{2,1}$-norm of the matrix $\mathbf{M}$, i.e., $\|\mathbf{M}\|_{2,1} = \sum_p \|\mathbf{m}^p\|_2$, where $\mathbf{m}^p$ is the p-th row of $\mathbf{M}$. In fact, this optimization problem is a general $l_{2,1}$-norm minmax problem. Wang et al. [104] proposed an efficient iterative algorithm to solve this problem.

2.3 Summary and Conclusions

In this chapter we reviewed a varity of MIL applications and algorithms. MIL has been successfully applied to visual recognition tasks such as image classification, image retrieval, object detection, video classification, and unconstrained video event detection. We also categorized the MIL algorithms in two main paradigms: instance-space methods and bag-space methods. In the instance-level paradigm, an instance-level classifier is trained to

classify positive and negative instances, and based on these classifiers a bag-level classifier is obtained. However, in the bag-level paradigm, a classifier is trained directly on the bags by extracting discriminative bag-level information from them.

A summery of the algorithms reviewed in this chapter is presented in Table 2.1. This table also provides the MI assumption used in each method. According to the standard MI assumption a bag is labeled positive if at least one of the instances in the bag is positive. Ratio-based assumptions refer to any assumption which is based on the proportion of positive/negative instances in a bag. The ratio-constrained assumption is a special ratio-based assumption, which states a bag is positive if at least a certain ratio of the instance are positive. In fact, the ratio-constrained assumption is a generalized version of the standard MI assumption. Metadata assumption is used by convention to refer to the assumption used in embedded-space and kernel-based bag-level methods [33]. This assumption originates from the fact that in these methods classification is performed in a metadata embedding space[2].

in Table 2.1, we also summarize our proposed models. MIRealBoost is a boosting framework for MIL, which can softly explore different levels of ambiguity using linguistic aggregation functions with different degrees of orness. Hence, the notion of positive bag is extended to a wider and more intuitive range of assumptions. The proposed algorithm is inspired by the ideas in the MILBoost [99] and RealBoost [34] algorithms. In summary, this algorithm maximizes the expected likelihood of training bags, where the bag likelihood is estimated by aggregating the likelihood of instances. In our second method, we propose multiple instance cardinality models. These are latent probabilistic graphical models which can integrate instance-level and bag-level interrelations in a bag and at the same time encode any cardinality relations on instance labels. Hence, they provide a unified framework for MIL, which addresses all major issues discussed in Chapter 1. To train these models, we propose novel algorithms based on max-margin classification, kernel learning, and gradient boosting.

Table 2.1: A list of some well-known MIL methods

Method	**Summery of the algorithm**	**Base Discrimination Level**	**Multi-instance assumption**
Axis-Parallel Rectangles [25]	Finding a hyper-rectangle that maximizes the number of positive bags which have at least one instance in this region, but excludes instances of negative bags as much as possible.	Instance space	Standard assumption
Diverse Density [75]	Estimating the probability of instances based on distance from an instance prototype, which is close to at least one instance of every positive training bag but far from instnaces of all negative training bags.	Instance space	Standard assumption

Continued on next page

[2]Note that the mapping is explicit in embedded-space methods and implicit in kernel-based methods

Table 2.1 – *Continued from previous page*

Method	Summery of the algorithm	Base Discrimination Level	Multi-instance assumption
EM-DD [120]	Using expectation-maximization to maximize the diverse density function.	Instance space	Standard assumption
mi-SVM [6]	Maximizing the instance margin jointly over the latent instance labels, using an iterative algorithm.	Instance space	Standard assumption
MI-SVM [6]	Maximizing the bag margin in an iterative procedure, where at each iteration every positive bag is represented by the most postive instance of the bag.	Instance space	Standard assumption
sMIL, stMIL [12]	sMIL modifies miSVM constraints to be more effective for sparse positive bags. stMIL is the transductive SVM version of sMIL.	Instance space	Standard assumption
AL-SVM, AW-SVM, ALP-SVM [40]	Optimizing mi-SVM and MI-SVM objective functions with deterministic annealing.	Instance space	Standard assumption for AL-SVM & AW-SVM. Ratio-based for ALP-SVM.
MI-Forests [63]	Optimizing a confidence maximizing loss function over randomized trees, using an iterative DA-based method.	Instance space	Standard assumption
MILBoost [99]	Maximizing the log likelihood of training bags using AnyBoost framework.	Instance space	Standard assumption
MIL-CPB [67]	Optimizing SVM-like objective functions with ratio-based MIL constraints for the positive bags, using an iterative cutting plane algorithm.	Instance space	Ratio-constrained assumption
$\propto$SVM [118]	Solving a max-margin mixed-integer optimization problem, given predetermined instance label proportions, following alternating optimization or convex relaxation.	Instance space	Ratio-based assumption
MI-CRF [23]	Using a CRF where each node represents a bag which can take one of its instance as the value. In this model, all the bags are jointly classified based on unary instance classifiers and pairwise dissimilarity measurements	Instance space	Standard assumption
Structured Bag Models [109]	Using CRFs to model the bag structures and at the same time incorporating different MIL constraints. Learning is performed by minimizing an objective function with deterministic annealing approach	Instance space	Standard and Ratio-based assumptions
SetMaxRBM [71]	Extending restricted Bltzman machines (RBMs) to classify general sets of data	Bag space	No assumption
Generative Models for MIL [1]	Using Bayesian networks with different structures to learn generative models for MIL	Instance space	Standard assumption

Continued on next page

Table 2.1 – *Continued from previous page*

Method	Summery of the algorithm	Base Discrimination Level	Multi-instance assumption
Simple MI [29]	Mapping each bag to average of its instances and training a standard single-instance classifier.	Bag space	Metadata assumption
Histogram-Based Methods [5]	Finding a vocabulary of concepts by clustering the instances. Then, mapping each bag to a histogram vector of the concepts and finally train a single-instance classifier.	Bag space	Metadata assumption
DD-SVM [19] & MILES [18]	Mapping each bag to a vector built by the distances between the bag and instance prototypes of the DD algorithm. Next, classifying the vectors by the regular SVM (in DD-SV) or 1-norm SVM (in MILES).	Bag space	Metadata assumption
MI kernels [39]	Defining a number of MI kernels on bags and plug them into kernel methods.	Bag space	Metadata assumption
miGraph & MIGraph [121]	Mapping a bag into an undirected graph and designing a graph kernel. Next, classifying the bags by a kernel machine.	Bag space	Metadata assumption
Citation kNN [107]	Using a bag-to-bag distance in a modified nearest neighbor appraoch, where each bag is classified by majority voting among both citers and references.	Bag space	Nearest neighbor assumption (with B2B distance)
M-C2B [104]	Learning a robust and discriminative class-to-bag (C2B) distance for MIL by solving an $l_{2,1}$-norm minmax problem.	Bag space	Nearest neighbor assumption (with C2B distance)
Ours: MIRealBoost [47]	Maximizing the expected log likelihood of training bags, using standard RealBoost algorithm and linguistic aggregation functions.	Instance space	Soft linguistic cardinality codes (e.g. some, many)
Ours: Cardinality Models	Modeling bags using Markov networks with parameterized cardinality potentials so that different cardinality-based MI assumptions can be plugged into the models or even learned from data. Three different algorithms are proposed to train these models. First, learning is formulated as a latent max-margin classification problem and solved with a non-convex cutting plane method. Second, a multi-instance kernel is defined and tuned to classify these models. Third, a gradient boosting algorithm is introduced to maximize the data likelihood function.	Instance space + bag space	Any cardinality-based assumption + metadata assumption of bag-level features.

Chapter 3

Multiple Instance Real Boosting with Aggregation Functions

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 in data and encode a variety of soft multi-instance assumptions. We evaluate the performance of the proposed algorithm on popular MIL datasets. The experimental results show that this algorithm outperforms the standard MILBoost algorithm.

3.1 Overview

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 [99], image categorization [18], image retrieval [67, 30], and object tracking [8] from weakly annotated data. For example, Viola et al. [99] 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. [18] 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 L1-norm SVM and used for image categorization. Gehler and Chapelle [40] 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 [12] 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. [30] and Li et al. [67] 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 multi-instance assumption, where a positive bag contains 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 tackle the ambiguities in the instances.

In this work, we propose a novel algorithm called MIRealBoost to train a multi-instance classifier. The main advantage of our framework is that a diverse set of aggregation functions are introduced to encode various multi-instance assumptions and deal with different levels of ambiguity in the bags. 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 [40, 30, 67], which need prior knowledge about exact fraction of positives inside bags. Instead, our proposed framework can roughly extract this knowledge by exploring different aggregation functions and directly optimizes the expected data likelihood to train the bag classifier. In addition, this algorithm has the general advantages of boosting algorithms like simple programming, few parameters for tuning, and ability of feature selection.

The rest of this chapter is organized as follows. Section 3.2 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 3.3 the experiments are presented, and MIRealBoost is compared with the state-of-the-art algoithms. Finally, the conclusions are drawn in Section 3.4.

3.2 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 $\{(X_1, Y_1), \cdots, (X_N, Y_N)\}$. According to the standard MI assumption, 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. [99] introduced the MILBoost algorithm for standard MIL, based on the AnyBoost framework [76]. This algorithm trains a boosting classifier which maximizes the log likelihood of the training bags:

$$L = \sum_i \mathbb{1}(Y_i = 1) \log p(X_i) + \mathbb{1}(Y_i = -1) \log \left(1 - p(X_i)\right),$$

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:

$$p(X_i) = 1 - \prod_{x_{ij} \in X_i} \left(1 - p(x_{ij})\right). \tag{3.1}$$

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 section, we propose an algorithm which maximizes the expected log likelihood of training examples based on RealBoost framework [34] by training 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 assumptions. 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.

3.2.1 Ordered Weighted Averaging

Ordered Weighted Averaging (OWA) as an aggregation operator was proposed by Yager [113]. OWA is a mapping $\mathbf{owa} : [0,1]^n \to [0,1]$, 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 (3.2).

$$\mathbf{owa}\left(a_1, a_2, \cdots, a_n\right) = \sum_{i=1}^{n} b_i w_i. \tag{3.2}$$

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:

$$\theta\left(w_1, w_2, \cdots, w_n\right) = \left(\frac{1}{n-1}\right) \sum_{i=1}^{n} \left((n-i) w_i\right). \tag{3.3}$$

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 $Q : [0,1] \to [0,1]$ such that $Q(0) = 0$ and $Q(1) = 1$. Consequently, the OWA weight vector is computed based on Q using (3.4).

$$w_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n}). \tag{3.4}$$

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 [72, 113]: *At least one* ($\alpha \to 0$, i.e. *Max* function), *Few* ($\alpha = 0.1$), *Some* ($\alpha = 0.5$), *Half* ($\alpha = 1$), *Many* ($\alpha = 2$), *Most* ($\alpha = 10$), *All* ($\alpha \to \infty$), which we also summarize in Table 3.1.

Table 3.1: Family of RIM qunatifiers and their relevant values of α and θ

Linquistic quantifier	α	Orness (θ)
At least one of them	$\alpha \to 0$	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	$\alpha \to \infty$	0.001

3.2.2 Multiple Instance RealBoost

In our proposed 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

$$p(X) = \frac{e^{F^b(X)}}{e^{F^b(X)} + e^{-F^b(X)}}. \tag{3.5}$$

Under this model, the binomial log-likelihood will be

$$\begin{aligned} l\left(Y, p(X)\right) &= \mathbb{1}(Y=1)\log p(X) + \mathbb{1}(Y=-1)\log\left(1-p(X)\right) \\ &= -\log\left(1+e^{-2YF^b(X)}\right) \end{aligned} \tag{3.6}$$

Our goal is to maximize the expected log-likelihood $E\, l(Y, p(X))$. It is proved in [34] that the maximizer of this function is $p(X) = P(Y=1|X)$ or equivalently:

$$F^b(X) = \frac{1}{2}\log\frac{P(Y=1|X)}{1-P(Y=1|X)}. \tag{3.7}$$

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

$$P(Y=1|X) = \mathbf{agg}_{x\in X}(P(y=1|x)). \tag{3.8}$$

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 $w_{ij}^p = 1/\sum_i(|X_i|)$, the pseudo-labels $y_{ij}^p = Y_i$, and the confidence of each instance $F(x_{ij}) = 0$.
for $m = 1 \to M$ **do**
  for each available feature $h_k(.), k = 1 \to K$ **do**
    Compute weak classifier of each instance w.r.t. each feature $f_m^k(x_{ij}) = \frac{1}{2}\log\frac{\hat{P}_{\{h_k(x_{ij}),y_{ij}^p,w_{ij}^p\}}(h_k(x_{ij})|y=1)}{\hat{P}_{\{h_k(x_{ij}),y_{ij}^p,w_{ij}^p\}}(h_k(x_{ij})|y=-1)}$
    Compute the probability of each instance $p^k(x_{ij}) = \frac{e^{\left(F(x_{ij})+f_m^k(x_{ij})\right)}}{e^{\left(F(x_{ij})+f_m^k(x_{ij})\right)}+e^{-\left(F(x_{ij})+f_m^k(x_{ij})\right)}}$.
    Compute the probability of each bag $p^k(X_i) = \mathbf{agg}_{x_{ij}}\left(p^k(x_{ij})\right)$.
    Compute the empirical log-likelihood $L^k = \sum_i \mathbb{1}(Y_i = 1)\log p^k(X_i) + \mathbb{1}(Y_i = -1)\log\left(1 - p^k(X_i)\right)$
  end for
  Set $k^\star = \arg\max_k L^k$.
  Set $F(x_{ij}) \leftarrow F(x_{ij}) + f_m^{k^\star}(x_{ij})$
  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 $w_{ij}^p \leftarrow e^{-Y_i F^b(X_i)}$, $i = 1, \cdots, N$, and normalize the weights such that $\sum_{ij} w_{ij}^p = 1$.
end for
Output: The bag-classifier $\mathbf{sign}\left(F^b(X)\right)$.

The aggregation function **agg** can be the NOR model in (3.1) or the OWA operators in (3.2). On the other hand, if an instance classifier $H(x) = \mathbf{sign}(F(x))$ is trained by the original instance-level RealBoost algorithm [34], the probability of each instance is given by

$$P(y = 1|x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}. \tag{3.9}$$

Therefore, if we know the confidence score 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:

$$E\, e^{-y(F_{m-1}(x)+f_m(x))}. \tag{3.10}$$

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

$$f_m(x) = \frac{1}{2} \log \frac{P_w\left(y=1|x\right)}{P_w\left(y=-1|x\right)}, \tag{3.11}$$

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

$$f_m(x) = \frac{1}{2} \log \frac{P_w\left(x|y=1\right)}{P_w\left(x|y=-1\right)}. \tag{3.12}$$

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 $w^p_{ij} = e^{-Y_i F^b_{m-1}(X_i)}$. 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

$$f_m(x) = \frac{1}{2} \log \frac{\hat{P}_{\{x^p_{ij}, y^p_{ij}, w^p_{ij}\}}\left(x|y=1\right)}{\hat{P}_{\{x^p_{ij}, y^p_{ij}, w^p_{ij}\}}\left(x|y=-1\right)}. \tag{3.13}$$

$$F_m(x_{ij}) = F_{m-1}(x_{ij}) + f_m(x_{ij}). \tag{3.14}$$

Now that we have the confidence of all the instances, we can find the confidence of each bag by (3.9), (3.8), (3.7) 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 (3.1), 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.

3.3 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 [6] and *Musk1* and *Musk2* drug activity prediction datasets [25]. 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 3.2. 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 [40] 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 3.2: 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 3.3 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 [63], we also report the MILBoost results from this paper.

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

Table 3.3: 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 the state-of-the-art MIL methods is provided in Table 3.4. 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).

Table 3.4: 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 [63]	84	**64**	82	85	82
MI-Kernel [6]	84	60	84	88	89
MI-SVM [6]	81	59	84	78	84
mi-SVM [6]	82	58	79	87	84
MILES [18]	81	62	80	88	83
SIL-SVM [12]	85	53	77	88	87
AW-SVM [40]	82	**64**	83	86	84
AL-SVM [40]	79	63	78	86	83
EM-DD [120]	78	56	72	85	85
MIGraph [121]	85	61	82	90	**90**
miGraph [121]	**87**	62	**86**	90	**90**

3.4 Conclusion

We proposed a novel framework for MIL based on boosting that can model a wide range of soft multi-instance assumptions and deal with 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 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. The proposed MIRealBoost algorithm achieves state-of-the-art results and outperforms the MILBoost algorithm on these datasets.

Chapter 4

Multiple Instance Classification by Max-Margin Training of Cardinality-Based Conditional Random Fields

We propose a probabilistic graphical framework for multiple instance learning (MIL) based on conditional random fields (CRFs). This framework can deal with different levels of labeling ambiguity in weakly supervised data (i.e., the portion of positive instances in a bag) by parameterizing cardinality potential functions. Consequently, it can be used to encode different cardinality-based multi-instance assumptions, ranging from the standard MIL assumption to more general assumptions. In addition, this framework can be efficiently used for both binary and multiclass classification. To this end, an efficient inference algorithm and a discriminative latent max-margin learning algorithm are introduced to train and evaluate the proposed multi-instance CRFs. We study the performance of the proposed framework on binary and multi-class MIL benchmark datasets as well as two challenging computer vision tasks: cyclist helmet recognition and human group activity recognition. Experimental results verify that encoding the degree of ambiguity in data can improve classification performance.

4.1 Overview

Multiple instance learning (MIL) aims to recognize patterns from weakly supervised data. Contrary to standard supervised learning, where each training instance is labeled, in the MIL paradigm a *bag of instances* share a label. For example in the binary MIL, each bag of instances is labeled positive or negative. The training data is given as labeled bags, and the goal is to predict the label of test bags. In the standard binary multi-instance

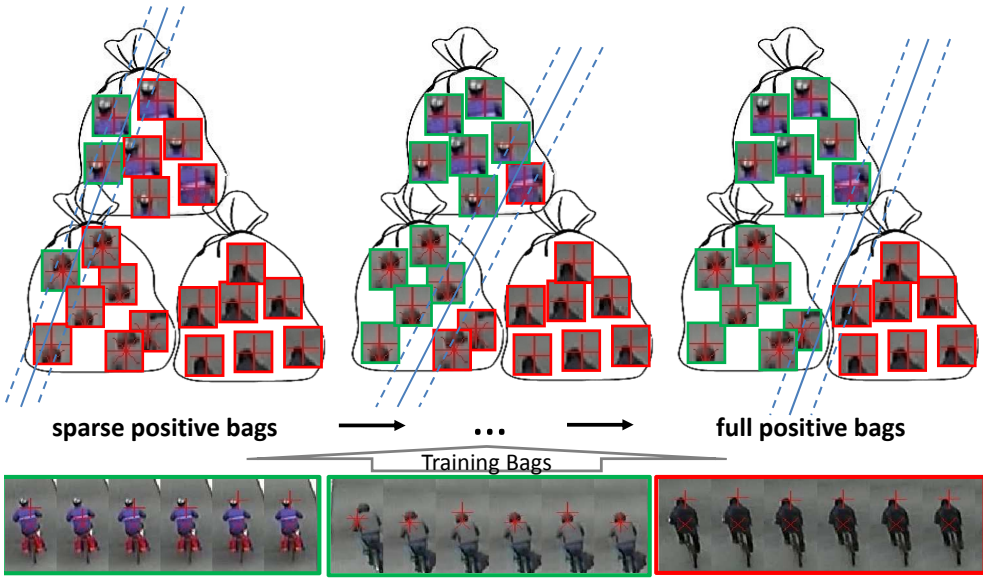


Figure 4.1: Cyclist helmet recognition using the proposed max-margin cardinality-based method. The goal is to recognize if the cyclist is wearing helmet or not, given the input video. Each video is treated as a bag of instances, where each instance is represented by an automatically detected window around the cyclist's head. The proposed cardinality-based models help to control the label proportions in the positive bag and encode a wide range of multi-instance assumptions.

(MI) assumption, a bag is positive if it contains *at least one* positive instance, while in a negative bag all the instances are negative. This ambiguity in the instance labels is passed on to the learning algorithm, which should incorporate the information to classify unseen bags. In this chapter we develop a novel framework for MIL, which can model more general multi-instance assumptions and deal with different levels of labeling ambiguity in the bags.

The standard multi-instance assumption (i.e., at least one of the instances in a positive bag is positive) is a too weak assumption in many MIL applications. For example, in the cyclist helmet recognition problem shown in Figure 4.1, the goal is to detect if the cyclist is wearing helmet given the automatically extracted track of the cyclist's head position. This can be modeled as a MIL problem, where the cyclist track is represented as a bag of image patches extracted around the estimated cyclist's head position in each frame. Because of the imperfect tracking, not all instances in a positive bag are positive. However, the positive instances are not sparse in positive bags, either. In fact, many instances are true positives and not just additional irrelevant elements in a bag. Using this prior information can help to train stronger classifiers. Further, because of noisy, occluded, or low-quality feature representations, negative bags can also contain instances that are effectively indistinguishable from positive instances. In these situations more robust multi-instance assumptions are needed.

On the other hand, considering and analyzing cardinality-based relations is intrinsic to some visual recognition problems. For example, in group activity recognition (e.g. [21]) the prominent approach to analyze the activity of a group of people is to look at the actions of individuals in a scene. A number of impressive methods have been developed for modeling the *structure* of a group activity [61, 20, 4], capturing spatio-temporal relations between people in a scene. However, these methods do not directly consider cardinality relations about the *number* of people that should be involved in an activity. These cardinality relations vary per activity. An activity such as a fall in a nursing home [61] is different in composition from an activity such as queuing [20], involving different numbers of people (one person falls, many people queue). Further, clutter, in the form of people in a scene performing unrelated actions, confounds recognition algorithms.

To address these issues, we develop a general MIL framework to encode various types of cardinality relations and make a flexible notion of labeled bags. This framework is built on a latent structured model based on conditional random fields (CRFs) to incorporate cardinality-based measurements over instances, which can extend from the notion of "at least one positive" to "at least some positives" to "nearly all positives." Thus, it can (1) deal with different levels of ambiguity or clutter in the data and (2) encode various kinds of cardianlity-based relations/constraints on instances, either predefined by the user or learned directly from the data. In fact, this framework can be even adapted to estimate the appropriate MIL notion from training data without prior assumption on the proportion of positives in the bags.

As explained in Chapter 2, there are some other works [40, 30, 67, 47, 109, 118] which try to model nonstandard and more general multi-instance assumptions. However, comparing to the previous works, our proposed method presents the following contributions. First, it can encode any cardinality-based multi-instance assumption in the bag[1]. It can even work without prior assumption on the cardinality of positive instances inside the bags and be trained to discover this knowledge directly from data. Second, it can be used for both binary and multi-class classification without converting the multi-class problem to multiple binary classification problems (e.g., by employing exhustive one-vs-all or one-vs-one appraoches, commonly used in MIL methods). Third, the inference and learning of the proposed models is exact and no approximation or heuristics are required. Finally, the proposed model allows flexible integration of bag-level and instance-level information in a bag, leveraging benefits from both global and local representations of the bag in both bag and instance spaces. For example, an image can be jointly represented by local feature vectors extracted from several regions of interest in the image as well as a global feature vector extracted from the whole image.

[1]Although we focus on ratio-based cardinality assumptions in this work, but the proposed model is not limited to ratio-based assumptions and can be used to encode any cardinality-based assumptions on the instance labels

This chapter is organized as follows. Section 4.2 describes our framework of multi-instance learning with CRFs. In particular, the models for different multi-instance assumptions, including the standard MI assumption and more general ratio-based MI assumptions are described in this section. In Section 4.3 the inference and learning algorithms are explained. Section 4.4 provides experimental studies on MIL benchmark datasets as well as cyclist helmet classification and human group activity recognition. We conclude in Section 4.5.

4.2 MIL Using Cardinality-Based CRFs

In MIL, training examples are presented in bags where the instances in a bag share a label. In this section, we use cardinality-based CRFs to model MIL problems and develop a generalized notion of labeled bags. The proposed CRFs are used to define a scoring function for bag classification.

4.2.1 The Proposed CRF for MIL

In this section, we firstly introduce the model for binary multi-instance classification and next extend it for multiclass classification.

Binary Classification

Let $\mathcal{B} = \{\mathcal{I}_1, \cdots, \mathcal{I}_m\}$ denote a bag with m instances and a binary bag label $Y \in \{-1, 1\}$. Each instance $\mathcal{I}_i$ is represented by a fixed-length feature vector $\mathbf{x}_i = [x_{i1}, \cdots, x_{iD}] \in \mathbb{R}^D$. Likewise, each bag might be globally described by another feature vector $\mathbf{X}$. For example, if the bag is an image, $\mathbf{X}$ can be a global bag-of-words feature vector extracted from the whole image. Another approach to construct $\mathbf{X}$ is using the prediction scores of other MIL methods[2] as a bag-level feature descriptor. Each instance $\mathcal{I}_i$ has also a hidden label y_i, and the collective binary instance labels of a bag are denoted by $\mathbf{y} = \{y_1, \cdots, y_m\}$. Given this notation, we propose a CRF to define a scoring function over tuples $(\mathbf{X}, \mathbf{x} = \{\mathbf{x}_i\}_{i=1}^m, Y, \mathbf{y} = \{y_i\}_{i=1}^m)$. This function is used to predict the label of a test bag by inferring the bag and instance labels which maximize the scoring function, given the feature vectors.

A graphical representation of the proposed cardinality-based CRF is shown in Figure 4.2. In our framework, we call this CRF the *"cardinality model"*. The components of the proposed cardinality model are described as follows. Each instance and its label are represented by two nodes in a clique. The potential function of this clique specifies a classifier for an individual instance. A second clique contains all instance labels and the bag label. This clique is used to define what makes a bag positive or negative. Varying this clique potential will lead to different MI assumptions, and is the focus of our work. Finally, there is an

[2]In our experiments, we use MI-Kernel [39].

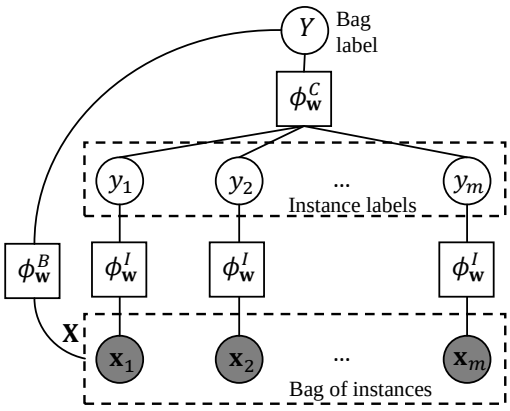


Figure 4.2: Graphical illustration of the proposed cardinality model for binary multi-instance learning. Instance potential functions $\phi^I_{\mathbf{w}}(\mathbf{x}_i, y_i)$ relate instances $\mathbf{x}_i$ to labels y_i. A second clique potential $\phi^C_{\mathbf{w}}(\mathbf{y}, Y)$ relates all instance labels y_i to the bag label Y. There is also an optional potential function $\phi^B_{\mathbf{w}}(\mathbf{X}, Y)$, which relates the global representation of the bag to the bag label.

optional clique potential between the global representation of the whole bag and the bag label.

We define the scoring function on these cliques as:

$$f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \sum_i \phi^I_{\mathbf{w}}(\mathbf{x}_i, y_i) + \phi^C_{\mathbf{w}}(\mathbf{y}, Y) + \phi^B_{\mathbf{w}}(\mathbf{X}, Y), \tag{4.1}$$

where $\phi^I_{\mathbf{w}}(\mathbf{x}_i, y_i)$ represents the potential between each instance and its label, $\phi^C_{\mathbf{w}}(\mathbf{y}, Y)$ is the clique potential over all the instance labels and the bag label, and finally $\phi^B_{\mathbf{w}}(\mathbf{X}, Y)$ expresses the potential between the bag-level feature vector and the bag label. Note that the potential functions are parametrized by the learning weights $\mathbf{w}$. We explain the details of these potential functions as follows.

Instance-Label Potential $\phi^I_{\mathbf{w}}(\mathbf{x}_i, y_i)$: This potential function models the compatibility between the ith instance feature vector $\mathbf{x}_i$ and its label y_i. It is parametrized as:

$$\begin{aligned} \phi^I_{\mathbf{w}}(\mathbf{x}_i, y_i) &= \mathbf{w}_I^\top \, \mathbf{x}_i \, \mathbb{1}(y_i = 1) \\ &= \mathbf{w}_I^\top \, \Psi_I(\mathbf{x}_i, y_i). \end{aligned} \tag{4.2}$$

Labels Clique Potential $\phi^C_{\mathbf{w}}(\mathbf{y}, Y)$: This potential function models the relations between the instance labels and the bag label. Since the MIL problems are defined based on the number of positive and negative instances, we need to formulate this as a *cardinality-based* clique potential. Cardinality-based potentials are only a function of label counts – in this case, the counts of the positive and negative instances in the bag.

By modifying the form of the cardinality-based potential, we can encode different MI assumptions, which will be shown in Section 4.2.2. Note that while for arbitrary clique potentials inference could be NP-complete, for cardinality potentials with binary variables exact and efficient inference algorithms exist. This will lead to efficient algorithms for learning and prediction, which will be described in Section 4.3.

In order to define the cardinality-based potentials, we will use the notation m^+ and m^- for the counts of instance labels in $\mathbf{y}$ which are positive and negative, respectively. The complete clique potential depends on these counts, and the bag label Y. Thus, we describe this clique potential by parameterizing two different cardinality potential functions, one for positive bags ($C^+_{\mathbf{w}}$) and one for negative bags ($C^-_{\mathbf{w}}$).

$$\begin{aligned}\phi^C_{\mathbf{w}}(\mathbf{y}, Y) &= C_{\mathbf{w}}\left(m^+, m^-, Y\right) \\ &= C^+_{\mathbf{w}}\left(m^+, m^-\right)\mathbb{1}(Y=1) \\ &+ C^-_{\mathbf{w}}\left(m^+, m^-\right)\mathbb{1}(Y=-1).\end{aligned} \tag{4.3}$$

Bag-Label Potential $\phi^B_{\mathbf{w}}(\mathbf{X}, Y)$: This potential function gives a global model of a bag, which describes how the bag as a whole entity is classified. It is parametrized as:

$$\begin{aligned}\phi^B_{\mathbf{w}}(\mathbf{X}, Y) &= \mathbf{w}_B^\top \mathbf{X}\,\mathbb{1}(Y=1) \\ &= \mathbf{w}_B^\top \Psi_B(\mathbf{X}, Y).\end{aligned} \tag{4.4}$$

Multiclass Classification

We can extend the binary model in Figure 4.2 for multiclass classification. The proposed multiclass model is illustrated in Figure 4.3. It can be observed that this CRF is formed by concatenation of the binary graphical model of each class. The main reason for this replication is that inference of cardinality clique potentials is exact and efficient only for binary labels. To this end, first we represent the multiclass bag label $Y \in \{1, 2, \cdots, L\}$ by a binary vector $(Y_1, Y_2, \cdots, Y_L)$, where $Y_l = 1$ if $Y = l$, and $Y_l = -1$ if $Y \neq l$. In addition, for each class l, we have binary instance labels $\mathbf{y}_l = \{y_{l1}, \cdots, y_{lm}\}$ ($y_{li} \in \{+1, -1\}$, $i = 1, \cdots, m$), indicating which instances are from (or relevant to) the lth class and which instances are not. We also denote the collection of all instance labels of all classes by $\mathbf{y}$. Putting all this together, the scoring function of the tuple $(\mathbf{X}, \mathbf{x}, Y, \mathbf{y})$ for the proposed multiclass graphical model is defined by:

$$f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \sum_{l=1}^{L}\left(\sum_i \phi^I_{\mathbf{w}l}(\mathbf{x}_i, y_{li}) + \phi^C_{\mathbf{w}l}(\mathbf{y}_l, Y_l) + \phi^B_{\mathbf{w}l}(\mathbf{X}, Y_l)\right), \tag{4.5}$$

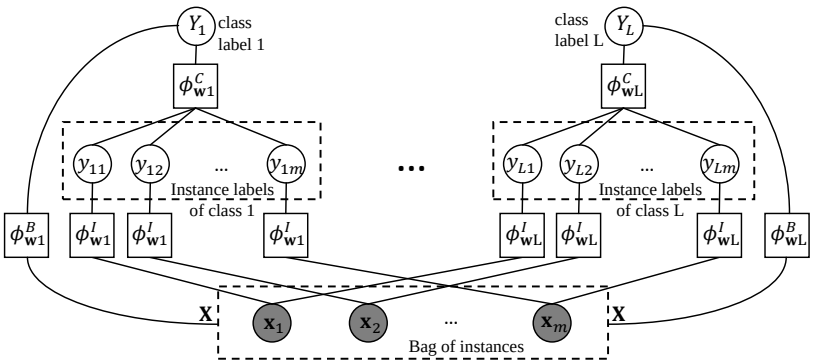


Figure 4.3: Graphical illustration of the proposed cardinality model for multiclass multi-instance learning.

where, similar to the binary model, the instance-label potentials $\phi^I_{\mathbf{w}l}(\mathbf{x}_i, y_{li})$, the labels clique potential $\phi^C_{\mathbf{w}l}(\mathbf{y}_l, Y_l)$, and the bag-label potential $\phi^B_{\mathbf{w}l}(\mathbf{X}, Y_l)$ are defined as follows.

$$\begin{aligned}\phi^I_{\mathbf{w}l}(\mathbf{x}_i, y_{li}) &= \mathbf{w}_{Il}^\top \mathbf{x}_i \, \mathbb{1}(y_{li} = 1) \\ &= \mathbf{w}_{Il}^\top \Psi_I(\mathbf{x}_i, y_{li}).\end{aligned} \tag{4.6}$$

$$\begin{aligned}\phi^C_{\mathbf{w}l}(\mathbf{y}_l, Y_l) &= C_{\mathbf{w}l}\left(m_l^+, m_l^-, Y_l\right) \\ &= C^+_{\mathbf{w}l}\left(m_l^+, m_l^-\right) \mathbb{1}(Y_l = 1) \\ &+ C^-_{\mathbf{w}l}\left(m_l^+, m_l^-\right) \mathbb{1}(Y_l = -1).\end{aligned} \tag{4.7}$$

$$\begin{aligned}\phi^B_{\mathbf{w}}(\mathbf{X}, Y_l) &= \mathbf{w}_{Bl}^\top \mathbf{X} \, \mathbb{1}(Y_l = 1) \\ &= \mathbf{w}_{Bl}^\top \Psi_B(\mathbf{X}, Y_l).\end{aligned} \tag{4.8}$$

The following section defines functions $C^+_{\mathbf{w}l}$ and $C^-_{\mathbf{w}l}$ that lead to a variety of MIL models.

4.2.2 The Proposed Cardinality Models of Multi-Instance Classification

In this section, we use our proposed cardinality model to encode different multi-instance assumptions.

Standard Cardinality Model (SCM)

This CRF models the multi-class multi-instance classification with the standard MI assumption, i.e., a bag of class l has at least one instance from the lth class. Thus, in this model,

the labels clique potential for each possible class $l \in \{1, \cdots, L\}$ is given by

$$C^{+}_{\mathbf{w}l}(0, m) = -\infty \tag{4.9}$$
$$C^{+}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = w^{+}_{Cl} \quad m^{+}_{l} = 1, \cdots, m \tag{4.10}$$
$$C^{-}_{\mathbf{w}l}(0, m) = w^{-}_{Cl} \tag{4.11}$$
$$C^{-}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = -\infty \quad m^{+}_{l} = 1, \cdots, m. \tag{4.12}$$

This clique potential states that in a bag of class l it is impossible to have no instance from the lth class (4.9), and there is the same potential of having one or more than one instance from the target class (4.10). However, if the bag label is not equal to l, none of the instances should be from this class (4.11) & (4.12). One could set w^{+}_{Cl} and w^{-}_{Cl} to a constant value (e.g. 0)[3], but we generally treat them as the model parameters and show how to learn them in Section 4.3.2.

Ratio-constrained Cardinality Model (RCM)

Ratio-constrained MI assumption extends the notion of labeled bags in MIL based on instance labels proportions. In the ratio-constrained cardinality model, each bag of class l contains at least a certain portion of instances from the lth class. For example, at least 30% of the instances should be from the lth class in a bag with label l. To encode this MI assumption with our proposed cardinality model, we only need to refine the functions $C^{+}_{\mathbf{w}l}$ and $C^{-}_{\mathbf{w}l}$:

$$\begin{aligned}
&C^{+}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = -\infty \quad 0 \leq \frac{m^{+}_{l}}{m} < \rho \\
&C^{+}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = w^{+}_{cl} \quad \rho \leq \frac{m^{+}_{l}}{m} \leq 1 \\
&C^{-}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = w^{-}_{cl} \quad 0 \leq \frac{m^{+}_{l}}{m} < \rho \\
&C^{-}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = -\infty \quad \rho \leq \frac{m^{+}_{l}}{m} \leq 1,
\end{aligned} \tag{4.13}$$

where ρ indicates the threshold proportion of relevant instances in a bag. The interesting case is $\rho = 0.5$, where we can learn models with majority assumption.

Generalized Cardinality Model (GCM)

The generalized cardinality model allows a very flexible notion of labeled bags. We allow the proportion of relevant and irrelevant instances in bags to be a learned parameter, discovered from the data. The MIL model will learn which fractions of instances tend to be of the

[3]Our experimental explorations show that setting these parameters to zero usually leads to satisfactory results

target class in a bag of that class. This cardinality model provides a very general model for multiple instance learning and is parametrized by:

$$
\begin{aligned}
&C^{+}_{\mathbf{w}l}(0, m) = -\infty \\
&C^{+}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = \sum_{k=1}^{K} w^{+}_{kl} \mathbb{1}(\frac{k-1}{K} < \frac{m^{+}_{l}}{m} \leq \frac{k}{K}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad m^{+}_{l} = 1, \cdots, m \\
&C^{-}_{\mathbf{w}l}(m^{+}_{l}, m - m^{+}_{l}) = \sum_{k=1}^{K} w^{-}_{kl} \mathbb{1}(\frac{k-1}{K} \leq \frac{m^{+}_{l}}{m} < \frac{k}{K}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad m^{+}_{l} = 0, \cdots, m-1 \\
&C^{-}_{\mathbf{w}l}(m, 0) = -\infty.
\end{aligned}
\tag{4.14}
$$

where K determines the number of weighted segments of a bag. This model divides the bag size into K equal parts, and the weight of each segment w_{kl} determines how important it is that the number of relevant instances (i.e., the instances from class l) be placed inside that interval. In other words, these learning weights specify the importance or impact of different witness ratios for labeling a bag. Large values of K provide more detailed and specific models of bag labeling by learning cardinality-based measures with finer resolution, while low values of K define a coarser model of bag. So, by controlling the granularity, this parameter is set in a trade-off between training accuracy and generalization ability[4].

The constraints $C^{+}_{\mathbf{w}l}(0, m) = -\infty$ and $C^{-}_{\mathbf{w}l}(m, 0) = -\infty$ are the only required prior information in this model, which break the symmetry between positive and negative bags and enforce at least one instance of a positive bag is positive and one instance of a negative bag is negative. Note that since this model is very general and unconstrained, it is vulnerable to overfitting (especially for multi-class classification) and requires careful training practices[5] to achieve successful results.

Linearity of the Models

In Section 4.2.1, we showed that the *instance-label* potentials and the *bag-label* potential are linear functions of the learning weights $\mathbf{w}$ (See equations (4.6) and (4.8)). Here, we demonstrate the linearity of the cardinality-based *labels clique* potential with respect to $\mathbf{w}$. Consequently, the whole model score would be a linear function of the learning parameters.

Given $C^{+}_{\mathbf{w}l}$ and $C^{-}_{\mathbf{w}l}$ defined for any of the standard, ratio-constrained, or generalized cardinality models, the labels clique potential for each class label (i.e., $\phi^{C}_{\mathbf{w}l}$) can be written

[4]In the experiments of Section 4.4, we validate on the values $K = 3$, $K = 5$, and $K = 10$ to roughly estimate this parameter.

[5]Some examples of good practices are smart initialization of the learning weights (e.g. start with the weights learned by the standard cardinality model) and early stopping on the training iterations by monitoring the validation error.

as:

$$\phi^{C}_{\mathbf{w}l}(\mathbf{y}_l, Y_l) = \mathbf{w}^{\top}_{Cl}\Psi_C(\mathbf{y}_l, Y_l) + g_C(\mathbf{y}_l, Y_l), \tag{4.15}$$

where $\mathbf{w}_{Cl}$ represents the concatenation of the learning parameters in $C^{+}_{\mathbf{w}l}$ and $C^{-}_{\mathbf{w}l}$, while $\Psi_C(\mathbf{y}_l, Y_l)$ and $g_C(\mathbf{y}_l, Y_l)$ are functions independent of $\mathbf{w}l$, which are specified by aggregation of the indicator functions.

Now, by integrating all the potential functions of the cardinality model, the scoring function introduced in (4.5) is reduced to the following linear expression:

$$f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \mathbf{w}^{\top}\Psi(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) + \sum_{l} g_C(\mathbf{y}_l, Y_l), \tag{4.16}$$

where

$$\begin{aligned}\Psi(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = [&\sum_{i} \Psi_I(\mathbf{x}_i, y_{1i})^{\top}, \cdots, \sum_{i} \Psi_I(\mathbf{x}_i, y_{Li})^{\top}, \\ &\Psi_C(\mathbf{y}_1, Y_1)^{\top}, \cdots, \Psi_C(\mathbf{y}_L, Y_L)^{\top}, \\ &\Psi_B(\mathbf{X}, Y_1)^{\top}, \cdots, \Psi_B(\mathbf{X}, Y_L)^{\top}]^{\top}.\end{aligned} \tag{4.17}$$

This linearity property facilitates parameter learning with gradient-based methods, which will be explained in Section 4.3.2.

4.3 Inference and Learning

The MIL models above define scoring functions $f_{\mathbf{w}}$ which consider counts of instance labels in a bag (see Eq. (4.5)). Using this, we can define a scoring function for assigning the bag label Y to a bag with bag-feature $\mathbf{X}$ and instance features $\mathbf{x}$ by MAP inference of the cardinality model over the hidden instance labels:

$$F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y) = \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}). \tag{4.18}$$

Below, we describe how to efficiently solve this inference problem for the cardinality-based cliques we defined above. Using this inference technique, learning can be performed using a max-margin criterion, as in the Latent SVM approach [32].

Classification of a new test bag can be done in a similar manner. We can predict the bag label by simply running inference, enumerating all possible Y and taking the maximum scoring bag label:

$$Y^{\star} = \arg\max_{Y} F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y). \tag{4.19}$$

4.3.1 Inference

The inference problem is to find the best set of instance labels of all class labels $\mathbf{y}^\star = \{\mathbf{y}_1^\star, \mathbf{y}_2^\star, \cdots, \mathbf{y}_L^\star\}$ given the observed feature vectors for the data $\{\mathbf{X}, \mathbf{x}\}$ and the bag label Y. Using (4.5) and (4.7), the inference problem in (4.18) can be written as

$$\mathbf{y}^\star = \max_{\mathbf{y}} \sum_{l=1}^{L} (\sum_i \phi_{\mathbf{w}l}^I(\mathbf{x}_i, y_{li}) + C_{\mathbf{w}l}(m_l^+, m_l^-, Y_l)). \tag{4.20}$$

However, the instance labels of each class are conditionally independent from the instance labels of other classes, given the input feature vectors and the bag label fixed. Thus, the original inference problem of all instance labels is decomposed and reduced to inference of the instance labels for each class label, separately:

$$\mathbf{y}_l^\star = \max_{\mathbf{y}_l} \sum_i \phi_{\mathbf{w}l}^I(\mathbf{x}_i, y_{li}) + C_{\mathbf{w}l}(m_l^+, m_l^-, Y_l). \tag{4.21}$$

This problem is the standard problem of inferring a probabilistic graphical model with cardinality clique potentials [44]. This class of PGMs is specified by two parts: the sum of individual node potentials and a potential function over all the nodes which only depends on the counts of the nodes which get specific labels. In our models, we only work with binary node labels (i.e., $y_{li} \in \{+1, -1\}$), for which there exists an exact inference algorithm with $O(m \log m)$ time complexity[6]. The inference algorithm is as follows. First, sort the instances in decreasing order of $\phi_{\mathbf{w}l}^I(\mathbf{x}_i, +1) - \phi_{\mathbf{w}l}^I(\mathbf{x}_i, -1)$. Then, for $k = 0, \cdots, m$, compute s_k^l, the sum of the top-k instance potentials $\phi_{\mathbf{w}l}^I(\mathbf{x}_i, +1) - \phi_{\mathbf{w}l}^I(\mathbf{x}_i, -1)$ plus the clique potential $C_{\mathbf{w}l}(k, m-k, Y_l)$. Finally, find $k_l^\star$ which gets the largest s_k^l, and inference is accomplished by assigning the top $k_l^\star$ instances to positive labels and the rest to negative labels. Repeating this algorithm for each class label, the full inference in (4.20) takes $O(L\, m \log m)$ time.

4.3.2 Learning

Let the training set is given by $\{(\mathbf{X}^1, \mathbf{x}^1, Y^1), \cdots, \left(\mathbf{X}^N, \mathbf{x}^N, Y^N\right)\}$, and the goal is to train the cardinality model by learning the parameters $\mathbf{w}$. Inspired by the relations to latent SVM [32], we formulate the learning problem as minimizing the regularized hinge loss function:

[6] For non-binary node labels, there exist only approximate inference algorithms. See [44] for more details.

$$
\begin{aligned}
&\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathcal{L}^n - \mathcal{R}^n) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \\
&\text{where } \mathcal{L}^n = \max_{Y} \max_{\mathbf{y}} (\Delta(Y, Y^n) + f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, \mathbf{y})), \\
&\mathcal{R}^n = \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y}), \\
&\Delta(Y, Y^n) = \begin{cases} 1 & \text{if } Y \neq Y^n \\ 0 & \text{if } Y = Y^n. \end{cases}
\end{aligned} \tag{4.22}
$$

One approach to solve this problem approximately is the iterative algorithm of alternating between inference of the latent variables and optimization of the model parameters. So, the first step estimates the instance labels and the second step learns a standard SVM classifier given the estimated instance labels. It can be shown that using this approach for the standard cardinality model and with binary class labels leads to the mi-SVM algorithm [6].

However, we use the non-convex regularized bundle method (NRBM) [26] to directly solve the optimization problem in (4.22). It has been shown that NRBM has a fast convergence rate compared to the state-of-the-art nonconvex optimiztion methods [28]. This method iteratively makes an increasingly accurate piecewise quadratic approximation of the objective function. At each iteration, a new linear cutting plane is obtained via the subgradient of the objective function and added to the piecewise quadratic approximation. To use this algorithm, the principal issue is to compute the subgradients $\partial_{\mathbf{w}}\mathcal{L}^n(\mathbf{w})$ and $\partial_{\mathbf{w}}\mathcal{R}^n(\mathbf{w})$. To this end, we need to know the subgradient of the cardinality model scoring function, i.e., $\partial_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y})$.

Following the linear model derived in (4.16), it is simple to show that

$$
\partial_{\mathbf{w}} f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \Psi(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}), \tag{4.23}
$$

Using equations (4.22) and (4.23), it can be shown that $\partial_{\mathbf{w}}\mathcal{L}^n(\mathbf{w}) = \Psi(\mathbf{X}^n, \mathbf{x}^n, Y^\star, \mathbf{y}^\star)$, where $(\mathbf{y}^\star, Y^\star)$ is the solution to the inference problem:

$$
\max_{Y} \max_{\mathbf{y}} (\Delta(Y, Y^n) + f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, \mathbf{y})). \tag{4.24}
$$

This inference problem can be solved using the algorithm in 4.3.1. In summary, we enumerate all possible Y, and for each fixed Y we find $\mathbf{y}$ by doing inference on the resulting graphical model (which has cardinality-based clique potentials and can be inferred efficiently). Then, the Y with the highest value gives the predicted bag label $Y^\star$.

In the same way, it can be shown that $\partial_{\mathbf{w}}\mathcal{R}^n(\mathbf{w}) = \Psi(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y}^\star)$, where $\mathbf{y}^\star$ is the solution to the inference problem:

$$\max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y}). \tag{4.25}$$

4.4 Experiments

In this section, we show the performance of the proposed framework in different classification tasks. First, the standard cardinality model is evaluated on binary and multiclass MIL benchmark datasets. Next, the extended models are applied to the two challenging computer vision tasks of cyclist helmet recognition and human activity recognition to show that flexibility in the portion of positives in a bag can lead to improved classification accuracy.

4.4.1 Benchmark Datasets

In this section, we evaluate our proposed standard cardinality model on MIL benchmark datasets to demonstrate it can achieve the state of the art performance on standard datasets.

Binary Benchmarks

We evaluate the standard cardinality model on five popular binary MIL datasets[7]. These benchmark datasets are the *Elephant*, *Fox*, *Tiger* image data sets [6] and *Musk1* and *Musk2* drug activity prediction data sets [25]. In the image data sets, each bag represents an image, and the instances inside the bag represent 230-D feature vectors of different segmented blobs of the image. These 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).

In all experiments of this section, the instance features have been extended by approximate explicit *intersection* kernel mapping [97], and the bag features have been constructed by the prediction scores of the MI-Kernel method [39] with RBF kernel. In addition, the features have been preprocessed by scaling the original features to the range $[0, 1]$. At each experimental trial, we run the non-convex cutting plane algorithm with all the learning weights initialized to 0 (except bag features[8]) and at most 100 iterations. The regularization parameter λ was roughly optimized on the 10-fold cross-validation accuracy by grid

[7]The original data sets are available online at `http://www.cs.columbia.edu/~andrews/mil/datasets.html`.

[8]Since the bag features are the MI-Kernel prediction scores, we initialize them with small positive values, e.g. 0.1, so that the first iteration of the algorithm will be the same as MI-Kernel

search in a set of predetermined values (0.1, 1, 10, and 100). The averaged classification accuracies for the standard cardinality model on different datasets are shown in Table 4.1. This table also includes the classification results of MI-Kernel method to show the performance of the bag features alone. It can be observed that combining the cardinality model with the bag features achieves the best results.

Table 4.1: Evaluating the classification performance of MIMN model on binary benchmark datasets.

Method	Elephant	Fox	Tiger	Musk1	Musk2
MI-Kernel	84.2	60.3	84.3	88.0	89.3
Cardinality model (without bag features)	87.0	60.5	84.5	87.1	87.2
Cardinality model (with bag features)	**89.0**	**63.5**	**85.5**	**88.2**	**92.3**

Now, we compare the standard cardinality model with the state-of-the-art MIL methods in Table 4.2. The performance of the methods varies depending on the data set. However, the standard cardinality model is always among the best methods. More specifically, it achieves the best accuracy on the Elephant, Fox, Tiger, and Musk2 data sets.

Table 4.2: Comparison between state-of-the-art MIL methods on the binary MIL benchmark datasets. The best and second best results are highlighted in bold and italic face respectively.

Method	Elephant	Fox	Tiger	Musk1	Musk2
Cardinality model	**89**	**64**	**86**	88	**92**
mi-SVM [6]	82	58	79	87	84
MI-SVM [6]	81	59	84	78	84
MI-Kernel [39]	84	60	84	88	*89*
γ-rule SVM [70]	84	63	81	88	85
SetMaxRBM$^{\text{XOR}}$ [71]	*88*	60	83	84	84
MIRealBoost [47]	83	63	73	**91**	77
MIForest [63]	84	**64**	82	85	82
SVR-SVM [66]	85	63	80	88	85
MIGraph [121]	85	61	82	*90*	*90*
miGraph [121]	87	62	**86**	*90*	*90*
MILES [18]	81	62	80	88	83
AW-SVM [40]	82	**64**	83	86	84
AL-SVM [40]	79	63	78	86	83
EM-DD [120]	78	56	72	85	85

Multiclass Benchmarks

In this section, we evaluate the multiclass extension of the standard cardinality model for image categorization on the COREL dataset. We work on the 1000-image and 2000-image

datasets[9] [18], which contain ten and twenty categories with 100 image per category. Each image is represented as a bag of instances, where the instances are the ROIs (Region of Interests) described by nine features (representing color, shape, and energy).

We perform the experiments with the same setup as in Section 4.4.1, i.e. extending and scaling the instance features and extracting MI-Kernel bag features. Also, the same experimental routine as described in [18] was used: the images of each category are split into half for training and test, and the experiment on each dataset is repeated five times. The results are provided in Table 4.3 and compared with other MIL methods. Note that the accuracy of MI-Kernel is based on our implementation, and for the other methods the numbers are reported from [65]. As seen in the table, the standard cardinality model is competitive with the state-of-the-art methods.

Table 4.3: Comparison between state-of-the-art MIL methods on the COREL image datasets. The numbers show the average accuracy over 5 trials and the corresponding 95% intervals.

Method	1000-Image	2000-Image
Cardinality model	85.6 ± 0.5	71.6 ± 1.0
MI-Kernel [39]	84.1 ± 0.6	69.1 ± 0.7
MKSVM-MIL [65]	85.2 ± 1.1	71.3 ± 1.2
MILES [18]	81.5 ± 3.0	68.7 ± 1.4
DD-SVM [19]	74.7 ± 1.6	67.5 ± 0.8
MissSVM [122]	78.0 ± 2.2	65.2 ± 3.1
MI-SVM [6]	74.7 ± 1.6	54.6 ± 1.5

4.4.2 Cyclist Helmet Recognition

In this section, we use our proposed models to address a binary video classification task. This problem is illustrated in Figure 4.4. Given an automatically-obtained cyclist trajectory, we must determine whether the cyclist is wearing a helmet or not. One can treat this as a MIL problem – each frame is an instance, and the trajectory forms a bag. The bag (trajectory) should be classified as containing a helmet-wearing cyclist or not. However, the standard MIL or traditional supervised learning approaches (e.g. classify each instance and majority vote) cannot easily handle this problem. Because of imperfection in tracking, it is unlikely that all the instances in a positive bag are truly positive – some will not be well centered on the cyclist's head due to jitter, regardless of the tracker used. Traditional supervised learning would have many corrupted positive instances of helmet-wearing cyclists. Standard MI assumption would not make full use of the training data, since each track would very likely have more than one positive instance.

[9]The original data sets are available online at `http://www.miproblems.org/datasets/corel`.

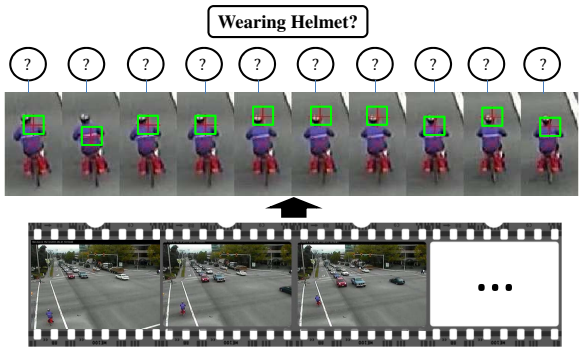


Figure 4.4: Cyclist helmet classification – is she wearing helmet? how many positives are in this bag? An automatic cyclist detector/tracker is run, with head position estimate in green rectangle. Data instances are features defined on the head position estimates, bags aggregate these over a track.

Experimental Setup

We work with cyclist trajectories automatically extracted from video data. The data are collected for a busy 4-legged intersection with vehicles, pedestrians, and cyclists, over a two-day period. Kanade-Lucas-Tomasi feature tracking and trajectory clustering are used to extract moving objects. These clusters are then automatically classified (vehicle, pedestrian, cyclist) by analyzing speed profiles (e.g. the pedalling cadence).

We chose a dataset of 24 cyclist tracks for our experiments – 12 wearing helmets and 12 not. The head location is estimated using background subtraction upon the tracks. We describe each frame of a track using texton histograms [73] in a region of size 20×20 around the head position (chosen after empirically examining other features). We report the results of helmet classification using leave-one-out cross-validation on this dataset.

We introduce a MIL approach to classify sequences. Each video is treated as a bag of frames represented by instances, and we use the proposed models in Section 4.2 to classify the bags. We also compare this approach with non-MIL methods. In the non-MIL approach, all frames from positive and negative training videos are put together and labelled according to their video labels. Next, a standard SVM classifier [13] is trained and used to predict each frame label of the test videos. Finally, the bag label is predicted by one of the following criteria:

- SVM-AtLeastOne: The bag label is positive if at least one of the instance labels is positive.
- SVM-Majority: The bag label is specified by the majority voting of the instance labels.

Experimental Results

For our proposed algorithms, we run the non-convex cutting plane algorithm with all the learning weights initialized to 0 and at most 100 iterations. For all the algorithms the

Table 4.4: Results of the experiments on cyclist helmet classification problem.

Method	Accuracy %
SVM-AtLeastOne	58.33
SVM-Majority	79.17
mi-SVM	62.50
Standard cardinality model	58.33
Ratio-constrained cardinality model ($\rho = 0.5$)	91.67
Generalized cardinality model ($K = 5$)	87.50

regularization parameter was estimated by grid search on the cross-validation accuracy. The average classification accuracy of each method is shown in Table 4.4. We include mi-SVM as an additional baseline. In addition, the results of the ratio-constrained cardinality model with different ρ values are demonstrated in Figure 4.5.

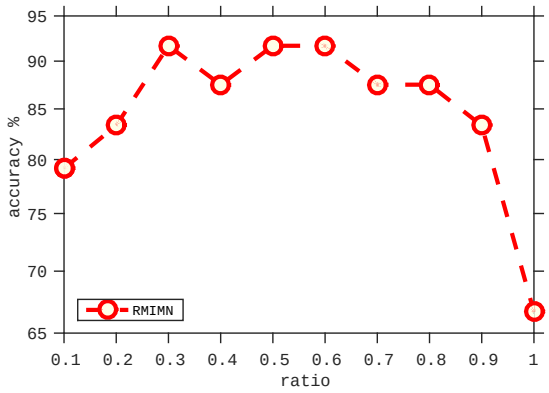
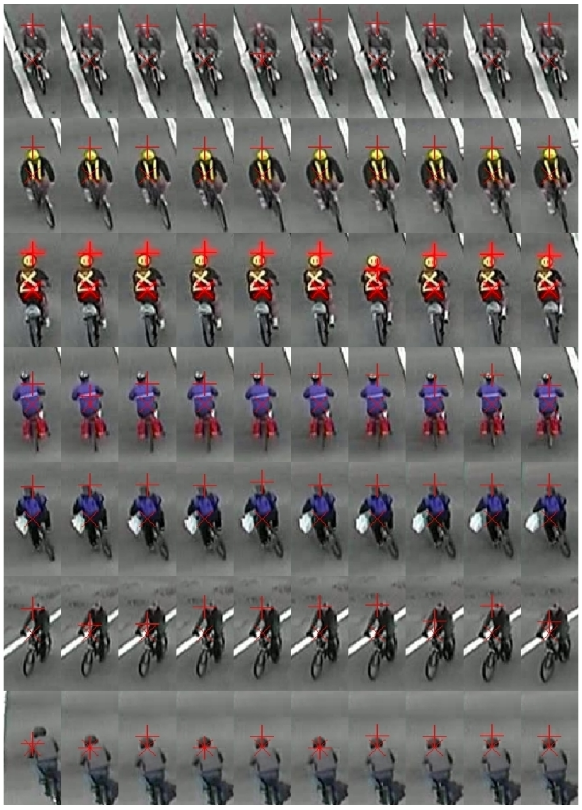
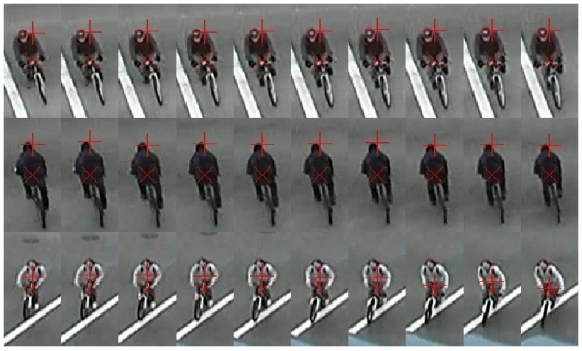


Figure 4.5: Cyclist helmet recognition accuracy with the ratio-constrained cardinality model and different values of the parameter ρ.

It can be observed that the classification accuracy of SVM-AtLeastOne, mi-SVM, and the standard cardinality model are quite low. This shows that the traditional classification approach (used in SVM-AtLeastOne) and the standard multi-instance assumption (used in mi-SVM and the standard cardinality model) are very inefficient in this problem. The standard MI assumption fails because it is very likely that at least one of the instances in a negative bag is classified as positive, and consequently most of the negative bags are assigned positive labels. This problem is due to the imperfection in the classifier and low-quality visual representation of the cyclist's head in the video. However, it is clearly evident that SVM-Majority, the ratio-constrained cardinality model (with most ρ values), and the generalized cardinality model are more robust to these defects. The results show that the ratio-constrained cardinality model (with $\rho = 0.5$) outperforms all the other methods. Also, it is shown that the generalized cardinality model has competitive performance. It learns the multi-instance assumption properly without any prior knowledge of the ambiguity level (e.g., parameter ρ) and classifies the videos successfully.



(a) correctly classified samples



(b) incorrectly classified samples

Figure 4.6: Samples of correctly and incorrectly classified videos by the generalized cardinality model. Red + shows automatic head position estimate.

Finally, we illustrate some videos correctly and incorrectly classified by our method in Figure 4.6.

4.4.3 Group Activity Recognition

In this section, we show the application of the proposed cardinality-based multi-instance models for group activity recognition. We run experiments on two datasets: nursing home dataset [61] and collective activity dataset [21].

Nursing Home Dataset

In this section, our method is evaluated for activity recognition in a nursing home. The dataset we use [61] provides scenes in which the individuals might be performing different actions such as walking, standing, sitting, bending, or falling. However, the goal is to detect the "fall" event, i.e., if any person is falling or not in a scene. Thus, we use the proposed binary standard cardinlity model to encode that at least one of the individuals is falling in a positive scene. Figure 4.7 illustrates the problem of fall scene detection in the nursing home dataset.

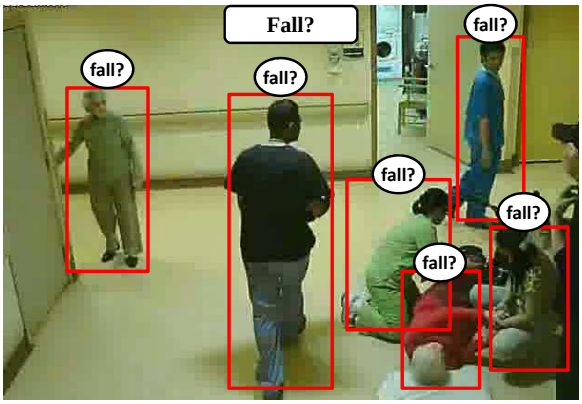


Figure 4.7: An example of "fall" scene from the nursing home dataset. We model this problem as a multi-instance learning problem, where each individual is represented as an instance and the goal is to recognize if any person is falling in the scene. To this end, we use our proposed standard cardinality model.

The dataset has 22 video clips (12 clips for training and 8 clips for test) with 2990 annotated frames, where about one third of them are assigned the “fall” activity label. We use the same features and experimental settings as used in [61]. The results in terms of classification accuracy are shown in Table 4.5. We compare our method with global bag-of-words method and the spatial structured models. It can be observed that our proposed cardinality model outperforms the others.

Collective Activity Dataset

In this section, we study the application of the proposed models in the multiclasss classification task of collective activity recognition. The collective activity dataset [21] comprises

Table 4.5: Comparison of different methods on the nursing home dataset in terms of classification accuracy (CA) and mean per-class accuracy (MPCA). We used the same features and experimental settings as in [61].

Method	CA	MPCA
Global bag-of-words with SVM [61]	52.6	53.9
Latent SVM with unconnected graph [61]	58.6	56.0
Latent SVM with tree-structured graph [61]	64.1	60.6
Latent SVM with complete graph [61]	70.0	63.1
Latent SVM with optimized graph structure [61]	71.2	65.0
Standard cardinality model (ours)	**76.1**	**66.2**

44 videos (about 2500 video frames) of *crossing*, *waiting*, *queuing*, *walking*, and *talking*. The goal is to classify the collective activity in each video frame, where the collective activity commonly tends to be the action that the *majority* of people in the scene are doing. For this purpose, each frame scene is modeled as a bag of people described by the *action context* feature descriptors[10] proposed in [61]. The MIL representation of this problem is shown in Figure 4.8. In our experiments, the same experimental setup is followed as explained in [61], i.e., the same 1/3 of the video clips were selected for test and the rest for training. We use our proposed ratio-constrained cardinality model with $\rho = 0.5$ to encode *majority* multi-instance assumption on the action labels. The results are shown in Table 4.6 and compared with the following methods: (1) SVM with global bag-of-words features and (2) spatial latent structured models in [61].

(a) Waiting (b) Crossing

Figure 4.8: Two examples from the collective human activity recognition dataset. The left figure shows a scene where the collective activity is waiting while the right figure shows a similar scene but the collective activity is crossing. The intuition is that the collective activity tends to be the action that majority of people are doing. We model this problem as a multi-instance learning problem, where the goal is to recognize the collective activity in the scene by inferring the hidden action each person is doing. We use our proposed ratio-constrained cardinality model to encode the majority multi-instance assumption.

[10]Note that this feature descriptor is built on a spatio-temporal context region around any individual. So it encodes the spatio-temporal information in the action and its context. By using our multi-instance model, the spatio-temporal and cardinality information are combined.

Table 4.6: Comparison of different methods on collective activity dataset in terms of multi-class accuracy (MCA) and mean per-class accuracy (MPCA). We used the same features and experimental settings as in [61].

Method	MCA	MPCA
Global bag-of-words with SVM [61]	70.9	68.6
Latent SVM with optimized graph [61]	79.7	78.4
Standard cardinality model	78.9	76.3
Ratio-constrained cardinality model ($\rho = 0.5$)	**80.6**	**79.7**
Generalized cardinality model ($K = 3$)	75.0	71.3

Our proposed ratio-constrained cardinality model can achieve the best results, even compared to the structure-optimized spatial model in [61] by simply replacing spatial relations with cardinality relations. We also illustrate the confusion matrix for this experiment in Figure 4.9. Finally, visualization of some example recognition results are provided in Figure 4.10.

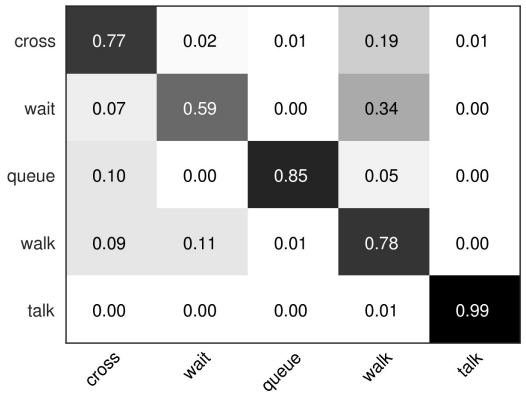


Figure 4.9: Confusion matrix for collective activity recognition using the ratio-constrained cardinality model (rows are the true labels, and columns are predicated labels).

4.5 Summary and Conclusion

We proposed a novel probabilistic graphical framework for both binary and multiclass multi-instance learning based on cardinality-based CRFs and max-margin discriminative training. This framework is flexible and can model the standard multi-instance assumption as well as more general MI assumptions. Thus, it is more robust to the amount of labeling ambiguity (i.e. true positive instances) in the bags. Specifically, it can be helpful in vision applications which exhibit imperfect annotation or ambiguous feature representations.

The experiments showed that learning and encoding the degree of ambiguity in the classifier can influence the accuracy of classification. We used the proposed framework for binary classification of cyclists with and without helmet. We also evaluated the performance

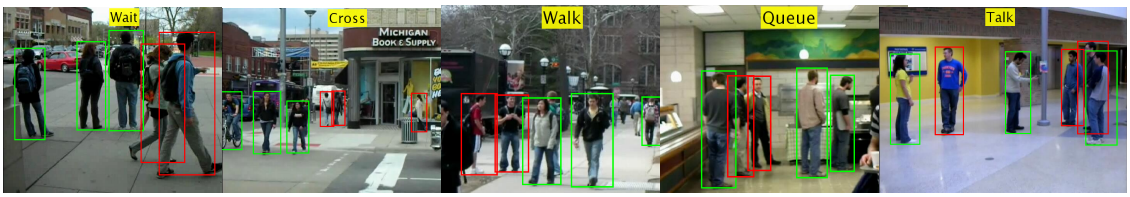


Figure 4.10: Visualization of some recognition results of the proposed method. Each figure is annotated by the predicted collective activity. Also each individual is represented by a colored bounding box. If an individual is involved in the predicted collective activity, the bounding box is green, otherwise red (In fact, these colors are used to illustrate predicted instance labels – green for positive label and red for negative label). For example, in the first figure from left, three people are waiting and two people are walking (passing by in the street). In the second figure, three people are crossing and the others are walking. In the third figure, all the people are walking except two people who are talking. Note that the instance labels are not always correctly predicted. For example in the fourth figure although all the people are involved in the queuing activity, however, two of them are incorrectly labeled by red. Also, in the last figure three people are incorrectly labeled. It seems that because of our weakly supervised learning framework (where we only incorporate the whole scene collective activity label in the max-margin learning formulation and model the individual action labels with hidden variables), the resulting model is sometimes conservative in predicting the instance labels and try to detect just enough positive instances to predict the whole scene collective activity correctly.

of the multiclass models on the collective activity recognition dataset. These are challenging problems, where the traditional supervised learning and standard MIL assumptions fail. However, the extended ratio-based models enhance classification performance by encoding more general and robust multi-instance assumptions and mining the degree of ambiguity.

The proposed graphical framework is flexible and can be easily extended or modified. For example, it can be modified for multi-label multi-instance learning, where a bag can take more than one label. Also, the model can be extended by defining more potential functions between the graph nodes. For example, new potential functions might be defined over neighbouring instance labels to model spatial or temporal relations between the instances. Finally, this framework could be adapted for individual classification from group statistics, and be applied to tasks such as privacy-preserving data mining, election results analysis, spam and fraud detection [82, 85, 118].

Chapter 5

A Multi-Instance Cardinality Potential Kernel for Visual Recognition

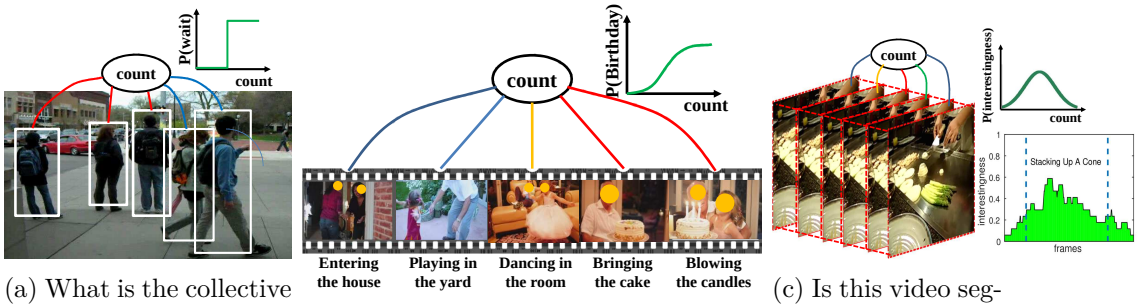
Many visual recognition problems can be approached by counting instances. To determine whether an event is present in a long internet video, one could count how many frames seem to contain the activity. Classifying the activity of a group of people can be done by counting the actions of individual people. Encoding these cardinality relationships can reduce sensitivity to clutter, in the form of irrelevant frames or individuals not involved in a group activity. This chapter develops a powerful and flexible kernel framework for multiple instance learning, which is built on probabilistic graphical models capturing cardinality relation between latent instance labels. Experiments on tasks such as human activity recognition, video event detection, and video summarization demonstrate the effectiveness of using cardinality relations for improving recognition results.

5.1 Overview

A number of visual recognition problems involve examining a set of instances, such as the people in an image or frames in a video. For example, in group activity recognition (e.g. [21]) the prominent approach to analyzing the activity of a group of people is to look at the actions of individuals in a scene. A number of impressive methods have been developed for modeling the *structure* of a group activity [61, 20, 4], capturing spatio-temporal relations between people in a scene. However, these methods do not directly consider cardinality relations about the *number* of people that should be involved in an activity. These cardinality relations vary per activity. An activity such as a fall in a nursing home [61] is different in composition from an activity such as queuing [20], involving different numbers of people (one person falls, many people queue). Further, clutter, in the form of people in a scene

performing unrelated actions, confounds recognition algorithms. In this chapter we present a framework built on a latent structured model to encode these cardinality relations and deal with the ambiguity or clutter in the data.

Another example is unconstrained internet video analysis. Detecting events in internet videos [79] or determining whether part of of a video is *interesting* [45] are challenging for many reasons, including temporal clutter – videos often contain frames unrelated to the event of interest or that are difficult to classify. Two broad approaches exist for video analysis, either relying on holistic bag-of-words models or building temporal models of events. Again, successful methods for modeling temporal structure exist (e.g. [43, 93, 90, 95]). Our method builds on these successes, but directly considers cardinality relations, counting how many frames of a video appear to contain a class of interest, and using soft and intuitive constraints such as "the more, the better" to enhance recognition.



(a) What is the collective activity?

(b) What is this video about?

(c) Is this video segment interesting?

Figure 5.1: Encoding cardinality relations can improve visual recognition. (a) An example of collective activity recognition. Three people are waiting, and two people are walking (passing by in the street). Using only spatial relations, it is hard to infer what the dominant activity is, but encoding the cardinality constraint that the collective activity tends to be the majority action helps to break the tie and favor "waiting" over "walking". (b) A "birthday party" video from the TRECVID MED11 dataset [79]. Some parts of the video are irrelevant to birthdays and some parts share similarity with other events such as "wedding". However, encoding soft cardinality constraints such as "the more relevant parts, the more confident decision", can enhance event detection. (c) A video from the SumMe summarization dataset [45]. The left image shows an important segment, where the chef is stacking up a cone. The right image shows the human-judged interesting-ness score of each frame. Even based on human judgment, not all parts of an important segment are equally interesting. Due to uncertainty in labeling the start and end of a segment, the cardinality potential might be non-monotonic.

Fig. 5.1 shows an overview of our method. We encode our intuition about these counting relations in a multiple instance learning framework. In multiple instance learning, the input to the algorithm is a set of labeled *bags* containing *instances*, where the instance labels are not given. We approach this problem by modeling the bag with a probabilistic latent structured model. Here, we highlight the major contributions of this chapter.

Showing the importance of cardinality relations for visual recognition. We show in different applications that encoding cardinality relations, either hard (e.g. *majority*) or soft (e.g. *the more, the better*), can help to enhance recognition performance and increase robustness against labeling ambiguity.

A kernelized framework for classification with cardinality relations. We use a latent structured model, which can easily encode any type of cardinality constraint on instance labels. A novel kernel is defined on these probabilistic models. We show that our proposed kernel method is effective, principled, and has exact and tractable inference and learning methods.

5.2 Related Work

In this chapter, we present a novel model for cardinality relations in visual recognition, in particular for the analysis of video sequences. Existing video analysis methods generally focus on structured spatio-temporal models, complementary to our proposed approach. For instance, pioneering work was done by Gupta et al. [43] in analyzing structured videos by creating "storyline" models populated from AND-OR graph representations. Related models have proven effective at analyzing scenes of human activity more broadly in work by Amer et al. [4]. A series of recent papers has focused on the problem of group activity recognition, inferring an activity that is performed by a set of people in a scene. Choi et al. [21, 20], Lan et al. [61], and Khamis et al. [54] devised models for spatial and temporal relations between the individuals involved in a putative interaction. Zhu et al. [123] consider contextual relations between humans and objects in a scene to detect interactions of interest. The structural relations exploited by these methods are a key component of activity understanding, but present different information from the cardinality relations we study.

Analogous approaches have been studied for "unconstrained" internet video analysis. Methods to capture the temporal structure of high-level events need to be robust to the presence of irrelevant frames. Successful models include Tian et al. [93] and Niebles et al. [78], who extend latent variable models in the temporal domain. Tang et al. [90] develop hidden Markov models with variable duration states to account for the temporal length of action segments. Vahdat et al. [95] compose a test video with a set of kernel matches to training videos. Tang et al. [91] effectively combine informative subsets of features extracted from videos to improve event detection. Bojanowski et al. [10] label videos with sequences of low-level actions. Pirsiavash and Ramanan [80] develop stochastic grammars for understanding structured events. Xu et al. [112] propose a feature fusion method based on utilizing related exemplars for event detection. Lai et al. [60] apply multiple instance learning to video event detection by representing a video as multi-granular temporal video

segments. Our work is similar in spirit, but contributes richer cardinality relations and more powerful kernel representations; empirically we show these can deliver superior performance.

The continued increase in the amount of video content available has rendered the summarization of unconstrained internet videos an important task. Kim et al. [56] build structured storyline-type representations for the events in a day. Khosla et al. [55] use web images as a prior for selecting good summaries of internet videos. Popatov et al. [81] learn the important components of videos of high-level events. Gygli et al. [45] propose a benchmark dataset for measuring interesting-ness of video clips and explore a set of high-level semantic features along with superframe segmentation for detecting interesting video clips. We demonstrate that our cardinality-based methods can be effective for this task as well, scoring a clip by the number of interesting frames it contains.

5.2.1 Multi-Instance Learning

We develop an algorithm based on multiple instance learning, where an input example consists of a bag of instances, such as a video represented as a bag of frames. The traditional assumption is that a bag is positive if it contains at least one positive instance, while in a negative bag all the instances are negative. However, this is a very weak assumption, and recent work has developed advanced algorithms with different assumptions [67, 47, 46, 60].

For example, Li et al. [67] formulated a prior on the number of positive instances in a bag, and used an iterative cutting plane algorithm with heuristics to approximate the resultant learning problem. Yu et al. [118] proposed $\propto$SVM for learning from instance proportions, and showed promising results on video event recognition [60]. Our work improves on this approach by permitting more general cardinality relations with an efficient and exact training scheme.

Similar to Chapter 4, our approach starts by modeling a bag of instances with a probabilistic model which has a cardinality-based clique potential between the instance labels. This cardinality potential facilitates defining any cardinality relations between the instance labels and efficient and exact solutions for both maximum a posteriori (MAP) and sum-product inference [44, 92]. Next, we extend our previous line of work in Chapter 4 by developing a novel kernel-based learning algorithm that enhances classification performance.

Kernel methods for multiple instance learning include Gärtner et al.'s [39] MI-Kernel, which is obtained by summing up the instance kernels between all instance pairs of two bags. Hence, 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 [58] proposed marginalized MI-Kernel. This kernel specifies 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 a probabilistic graph-structured kernel.

5.3 Proposed Method: Cardinality Kernel

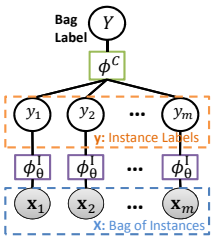
We propose a novel kernel for modeling cardinality relations, counting instance labels in a bag – for example the number of people in a scene who are performing an action. We start with a high-level overview of the method, following the depiction in Fig. 5.2.

The method operates in a multiple instance setting, where the input is bags of instances, and the task is to label each bag. For concreteness, Fig. 5.2(a) shows video event detection. Each video is a bag comprised of individual frames. The goal is to label a video according to whether a high-level event of interest is occurring in the video or not. Temporal clutter, in the form of irrelevant frames, is a challenge. Some frames may be directly related to the event of interest, while others are not.

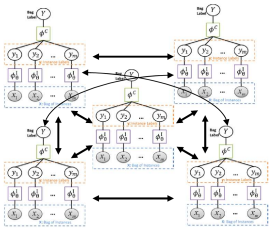
Fig. 5.2(b) shows a probabilistic model defined over each video. Each frame of a video can be labeled as containing the event of interest, or not. Ambiguity in this labeling is pervasive, since the low-level features defined on a frame are generally insufficient to make a clear decision about a high-level event label. The probabilistic model handles this ambiguity and a counting of frames – parameters encode the appearance of low-level features and the intuition that more frames relevant to the event of interest makes it more likely that the video as a whole should be given the event label.

A kernel is defined over these bags, shown in Fig. 5.2(c). Kernels compute a similarity between any two videos. In our case, this similarity is based on having similar cardinality relations, such as two videos having similar counts of frames containing an event of interest. Finally, this kernel can be used in any kernel method, such as an SVM for classification, Fig. 5.2(d).

(a) Preparing Training Data as Positive and Negative Bags of Instances



(b) Learning the cardinality model



(c) Computing Cardinality Kernels for all Bag Pairs

(d) Using a Kernel Method to Train a Classifier

Figure 5.2: The high-level scheme of the proposed kernel method for bag classification.

5.3.1 Cardinality Model

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 positively and negatively labeled variables. Given a set of binary random variables

$\mathbf{y} = \{y_1, y_2, \cdots, y_m\}$ ($y_i \in \{0, 1\}$), the cardinality potential model is described by the joint probability

$$P(\mathbf{y}) = \frac{C(\sum_i y_i) \prod_i \exp(\varphi_i y_i)}{\sum_{\mathbf{y}} C(\sum_i y_i) \prod_i \exp(\varphi_i y_i)}, \tag{5.1}$$

which consists of one cardinality potential $C(\cdot)$ over all the variables and unary potentials $\exp(\varphi_i y_i)$ on features φ_i on each single variable. Maximum a posteriori (MAP) inference of this model is straight-forward and takes $O(m \log m)$ time [44]. Sum-product inference is more involved, but efficient algorithms exist [92], computing all marginal probabilities of this model in $O\left(m \log^2 m\right)$ time.

In problems with multiple instances, there are assumptions or constraints which are defined on the counts of instance labels. For example, the standard multi-instance assumption states that at least one instance in a positive bag is positive. So, it is intuitive that these constraints can be modeled by a cardinality potential over the instance labels. This modeling helps to have exact and efficient solutions for MIL problems, using existing state-of-the-art inference and learning algorithms.

Using this cardinality potential model as the core, a probabilistic model of the likelihood of a bag of instances $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$ with the bag label $Y \in \{-1, +1\}$ and the instance labels $\mathbf{y}$ with model parameters $\boldsymbol{\theta}$, is built (c.f. [92]):

$$P(Y, \mathbf{y}|\mathbf{X}; \boldsymbol{\theta}) \propto \phi^C(Y, \mathbf{y}) \prod_i \phi^I_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i). \tag{5.2}$$

A graphical representation of the model is shown in Fig. 5.2(b). In our framework, we call this the "cardinality model", and the details of its components are described as follows:
Cardinality clique potential $\phi^C(Y, \mathbf{y})$: a clique potential over all the instance labels and the bag label. This is used to model multi-instance or label proportion assumptions and is 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. In this chapter we work with the "normal" model in (5.3) and the "ratio-constrained" model in (5.4).

$$\begin{aligned} C^{(+1)}(c) &= \exp\left(-(\frac{c}{m} - \mu)^2 / 2\sigma^2\right) \\ C^{(-1)}(c) &= \exp\left(-(\frac{c}{m})^2 / 2\sigma^2\right). \end{aligned} \tag{5.3}$$

$$\begin{aligned} C^{(+1)}(c) &= \mathbb{1}(\frac{c}{m} >= \rho) \\ C^{(-1)}(c) &= \mathbb{1}(\frac{c}{m} < \rho). \end{aligned} \tag{5.4}$$

The parameter μ in the normal model or ρ in the ratio-constrained model controls the proportion of positive labeled instances in a bag. The Normal model does not impose hard

constraints on the number of positive instances, and consequently a positive bag can have any proportion of positive instances but it is more likely to be around μ. On the other hand, the ratio-constrained model makes a hard constraint, assuming a bag must have at least a certain ratio (ρ) of positive instances.

Instance-label potential $\phi^I_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i)$: represents the potential between each instance and its label. Essentially, this potential describes how likely it is for an instance (e.g. video frame) to receive a certain label (e.g. relevant or not to an event). It is parameterized as:

$$\phi^I_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i) = \exp\left(\boldsymbol{\theta}^t \mathbf{x}_i \, y_i\right) \tag{5.5}$$

With these potential functions, the joint probability in (5.2) can be rewritten as

$$P(Y, \mathbf{y}|\mathbf{X}; \boldsymbol{\theta}) \propto C^{(Y)}(\sum_i y_i) \prod_i \exp\left(\boldsymbol{\theta}^t \mathbf{x}_i \, y_i\right). \tag{5.6}$$

And finally, the bag label likelihood, is obtained by

$$P(Y|\mathbf{X}; \boldsymbol{\theta}) = \sum_{\mathbf{y}} P(Y, \mathbf{y}|\mathbf{X}; \boldsymbol{\theta}) = \frac{Z^{(Y)}}{\sum_{Y'} Z^{(Y')}}, \tag{5.7}$$

$$\text{where } Z^{(Y)} = \sum_{\mathbf{y}} \left(C^{(Y)}(\sum_i y_i) \prod_i \exp\left(\boldsymbol{\theta}^t \mathbf{x}_i \, y_i\right) \right) \tag{5.8}$$

is the partition function of a standard cardinality potential model, which can be computed efficiently.

In summary, we have a unified probabilistic model which states the probability that a bag (e.g. video) receives a label based on classifying individual instances (e.g. frames) and a cardinality potential which prefers certain counts of positively labeled instances.

Parameter Learning

Since only the bag labels, and not the instance labels, are provided in training, the cardinality model is a hidden conditional random field (HCRF). A commonly used algorithm for learning HCRFs is maximum a posteriori estimation of the parameters given the parameter prior distributions by maximizing the following log likelihood function:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_i \log P(Y_i|\mathbf{X}_i; \boldsymbol{\theta}) - \lambda \, r(\boldsymbol{\theta}). \tag{5.9}$$

This is the standard maximum likelihood optimization of an HCRF 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 [83].

5.3.2 Cardinality Kernel

This section presents the proposed probabilistic kernel for multi-instance classification. Kernels operate over a pair of inputs, in this case two bags. This kernel is defined using the cardinality models defined above. Each bag has its own set of instances, and a probabilistic model is defined over each bag. A kernel over bags is formed by marginalizing over latent instance labels.

Given two bags $\mathbf{X}_p$ and $\mathbf{X}_q$, a joint kernel is defined between the combined instance features and instance labels for these bags $\mathbf{z}_p = (\mathbf{X}_p, \mathbf{y}_p)$ and $\mathbf{z}_q = (\mathbf{X}_q, \mathbf{y}_q)$:

$$k_z(\mathbf{z}_p, \mathbf{z}_q) = \sum_{i=1}^{m_p} \sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) k_y(y_{pi}, y_{qj}), \tag{5.10}$$

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[1]. By marginalizing the joint kernel w.r.t. the hidden instance labels and with independence assumed between the bags, a kernel is defined on the bags as:

$$\tilde{k}(\mathbf{X}_p, \mathbf{X}_q) = \sum_{\mathbf{y}_p, \mathbf{y}_q} P(\mathbf{y}_p|\mathbf{X}_p) P(\mathbf{y}_q|\mathbf{X}_q) k_z(\mathbf{z}_p, \mathbf{z}_q). \tag{5.11}$$

Combining the fully observed label instance kernel (5.10) with the probabilistic version (5.11), it can be shown that the marginalized joint kernel is reduced to

$$\sum_{i=1}^{m_p} \sum_{j=1}^{m_q} \sum_{\mathbf{y}_p, \mathbf{y}_q} \Big(k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) k_y(y_{pi}, y_{qj})\, P(y_{pi}|\mathbf{X}_p) P(y_{qj}|\mathbf{X}_q) \Big). \tag{5.12}$$

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

$$P(y_i|\mathbf{X}) = \sum_{Y} P(y_i|Y, \mathbf{X})\, P(Y|\mathbf{X}), \tag{5.13}$$

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 (5.7).

In general, any kernel for discrete spaces can be used as k_y. The most commonly used discrete kernel is $k_y(y_{pi}, y_{qj}) = \mathbb{1}(y_{pi} = y_{qj})$. However, since throughout this chapter we are dealing with binary instance labels and we are interested in performing recognition with the most salient and positively relevant instances of a bag, k_y is assumed to be

$$k_y(y_{pi}, y_{qj}) = \mathbb{1}(y_{pi} = 1) \cdot \mathbb{1}(y_{qj} = 1). \tag{5.14}$$

[1] If $k_y(\cdot, \cdot)$ is set to 1, the resulting kernel will be equivalent to MI-Kernel [39]. Also, note that since the joint kernel is obtained by summing and multiplying the base kernels, it is proved to be a kernel, has all kernel properties, and can be safely plugged into kernel methods.

Using this, the kernel in (5.12) is simplified as:

$$\tilde{k}(\mathbf{X}_p, \mathbf{X}_q) = \sum_{i=1}^{m_p} \sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) P(y_{pi} = 1|\mathbf{X}_p) P(y_{qj} = 1|\mathbf{X}_q). \tag{5.15}$$

It is interesting to note that this kernel in (5.15) can be rewritten as

$$\tilde{k}(\mathbf{X}_p, \mathbf{X}_q) = \Big(\sum_{i=1}^{m_p} P(y_{pi} = 1|\mathbf{X}_p) \Psi(\mathbf{x}_{pi}) \Big) \Big(\sum_{j=1}^{m_q} P(y_{qj} = 1|\mathbf{X}_q) \Psi(\mathbf{x}_{qj}) \Big), \tag{5.16}$$

where $\Psi(\mathbf{x})$ is the mapping function that maps the instances to the underlying feature space of the instance kernel k_x. This proves that the unnormalized cardinality kernel in the original feature space corresponds to weighted sum of the instances in the induced feature space of k_x, where the weights are the marginal probabilities inferred from the cardinality model in the original space. It can be also shown that in the more general case of $k_y(y_{pi}, y_{qj}) = \mathbb{1}(y_{pi} = y_{qj})$, the resulting cardinality kernel would correspond to weighted sum of all the instances which take the same instance label in the mapped feature space and concatenating them altogether.

Finally, to avoid bias towards the bags with large numbers of instances, the kernel is normalized as [39]:

$$k(\mathbf{X}_p, \mathbf{X}_q) = \frac{\tilde{k}(\mathbf{X}_p, \mathbf{X}_q)}{\sqrt{\tilde{k}(\mathbf{X}_p, \mathbf{X}_p)} \sqrt{\tilde{k}(\mathbf{X}_q, \mathbf{X}_q)}}. \tag{5.17}$$

We call the resulting kernel the *"Cardinality Kernel"*. By using this kernel in the standard kernel SVM, we propose a method for multi-instance classification with cardinality relations.

5.3.3 Algorithm Summary

The proposed algorithm is summarized as follows. First the parameters $\boldsymbol{\theta}$ of the cardinality model are learned (Sec. 5.3.1). These parameters control the classification of individual instances and the cardinality relations for bag classification. Next, the marginal probabilities of instance labels under this model are inferred and used in the kernel function in (5.15). Finally, the kernel is normalized and plugged into an SVM classifier.

5.3.4 Computational Complexity

First, we analyze the time complexity of computing the cardinality kernel. Assume that evaluation of the primitive kernel k_x takes $O(d)$ time, where d is the size of the instance feature vectors. Consequently, $k_x(\cdot, \cdot)$ 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 5.3.2, the time complexity of computing the marginal probabilities $P(y_i|Y, \mathbf{X})$ is $O\left(m \log^2 m\right)$. Thus, the kernel in (5.15) 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 with this kernel in a standard SVM for a single bag $\mathbf{X}_p$ 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 the training bags.

Now, we analyze the computational complexity of training the Cardinality Kernel. First, the parameters of the cardinality model should be learned. Learning this HCRF with regularized likelihood maximization takes $O(N_{iter}\,N\,\bar{m}\log^2\bar{m} + N_{iter}\,N\,\bar{m}\,d)$ time, where N is the number of training bags and 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[2], the computational complexity of the entire algorithm is $O(N_{iter}\,N\,\bar{m}\log^2\bar{m} + N_{iter}\,N\,\bar{m}\,d + N^2\,\bar{m}^2\,d + N^3)$.

We performed our experiments on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. As a numerical example, in the the collective activity recognition problem of Section 5.4.1 (which consists of 1908 training bags and 639 test bags with average 5 instances per bag and the instances are 240 dimensional), the total training time was around 10 minutes versus 7 minutes for the test time.

5.3.5 Parameter setting guidelines for the proposed Cardianlity Kernel method

To train the cardinality model, two hyper-parameters should be set: the regularization weight for likelihood optimization of the cardinality model (i.e., λ), and the regularization weight for training SVM (i.e., C). In our experiments we used the standard grid search over a set of predetermined values of λ and C (e.g., powers of 10) to set these parameters. However, this requires a quadratic number of runs of the algorithm, which might be inefficient. As a faster alternative, we found that just running MI-Kernel is effectively enough to estimate the value of parameter C. Next, we fix this parameter and run the Cardinality Kernel method to find the best estimate for λ.

Another setting is to initialize the learning parameters of the cardinality model (i.e., $\boldsymbol{\theta}$) for regularized likelihood maximization. Actually, this is a non-convex optimization and sensitive to initialization. In all our experiments we initialized the parameters $\boldsymbol{\theta}$ to zero. In fact, since the resulting kernel is finally plugged into SVM, and SVM relies on a convex optimization, the whole algorithm is fairly robust to initialization. For example, even if we freeze $\boldsymbol{\theta} = \mathbf{0}$ and do not train the cardinality model, it can be shown the resulting kernel is equivalent to MI-Kernel, which is an effective multi-instance algorithm [39].

[2]In our experiments, we used the LIBSVM [13] solver, which can be much more efficient than $O(N^3)$ in practice.

5.4 Experiments

We provide empirical results on three tasks: group activity recognition, video event detection, and video interesting-ness analysis.

5.4.1 Collective Activity Recognition

The Collective Activity Dataset [21] comprises 44 videos (about 2500 video frames) of *crossing*, *waiting*, *queuing*, *walking*, and *talking*. Our goal is to classify the collective activity in each frame. To this end, we model the scene as a bag of people represented by the *action context* feature descriptors[3] developed in [61]. We use our proposed algorithms with the ratio-constrained cardinality model in (5.4) with $\rho = 0.5$, to encode a majority cardinality relation. We follow the same experimental settings as used in [61], i.e., the same 1/3 of the video clips were selected for test and the rest for training. The one-versus-all technique was employed for multi-class classification. We applied l_2-norm regularization in likelihood maximization of the cardinality model and simply used linear kernels as the instance kernels in our method. The results of our Cardinality Kernel are shown in Table 5.1 and compared with the following methods[4]: (1) SVM on global bag-of-words, (2) Graph-structured latent SVM method in [61], (3) MI-Kernel [39], (4) Cardinality model of Section 5.3.1 (our own baseline).

Table 5.1: Comparison of classification accuracies of different algorithms on collective activity dataset. Both multi-class accuracy (MCA) and mean per-class (MPC) accuracy are shown because of class size imbalance.

Method	MCA	MPCA
Global bag-of-words with SVM [61]	70.9	68.6
Latent SVM with optimized graph [61]	79.7	78.4
Cardinality Model	79.5	78.7
MI-Kernel	80.3	78.4
Cardinality Kernel (our proposed method)	**83.4**	**81.9**

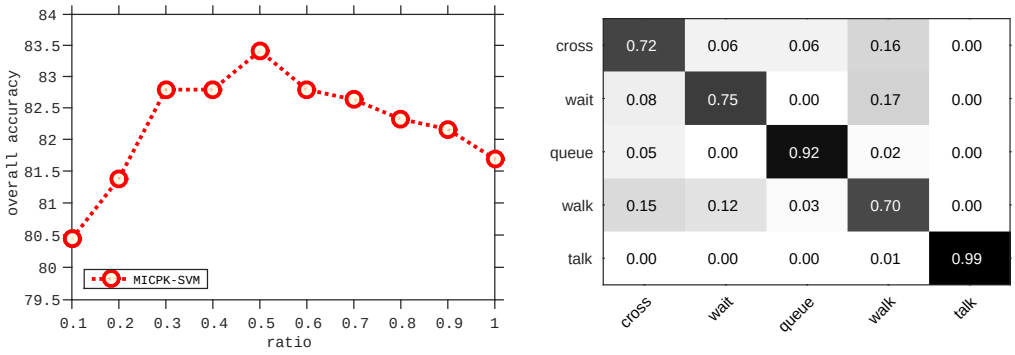
Our simple cardinality model can achieve results comparable to the structure-optimized models by replacing spatial relations with cardinality relations. Further, the proposed Cardinality Kernel can significantly improve classification performance of the cardinality model. Finally, our Cardinality Kernel is considerably better than MI-Kernel, showing the advantage of using importance weights (i.e. probability of being positive) of each instance for non-uniform aggregation of instance kernels.

Fig. 5.3a illustrates the effect of ρ in the ratio-constrained cardinality model on classification accuracy of the Cardinality Kernel. It can be seen that as expected, the best result

[3]These features are based on a spatio-temporal context region around a person. So by using our cardinality-based model, the spatio-temporal and cardinality information are combined.

[4]All these methods follow the standard evaluation protocol introduced in [20].

is achieved with $\rho = 0.5$. We also provide the confusion matrix for the Cardinality Kernel method in Fig. 5.3b. Finally, two examples of recognition with the cardinality model for crossing and waiting activities are visualized in Fig. 5.4.



(a) (b)

Figure 5.3: Performance of the Cardinality Kernel on collective activity dataset. (a) Classification accuracy with different values of ρ in the ratio-constrained cardinality model. (b) Confusion matrix with $\rho = 0.5$ (rows are the true labels, and columns are predicated labels)

Figure 5.4: Examples of recognition with the proposed model. The annotation of each person shows the true activity label of the scene with a tuple, indicating the MAP-inferred action label and the corresponding marginal probability w.r.t. the the scene activity label. -1 values denote "not" of the corresponding category; people performing other actions (left: two people not waiting, right: people not crossing the street) are correctly given -1 labels.

5.4.2 Event Detection

We evaluate our proposed method for event detection on the TRECVID MED11 dataset [79]. Because of temporal clutter in the videos, not all parts of a video are relevant to the underlying event, and the video segments might have unequal contributions to event detection. Our framework can deal with this temporal ambiguity, i.e., when the evidence of an event is occurring in a video and what the degree of discrimination or importance of each temporal segment is. We represent each video as a bag of ten temporal video segments, where each

segment is represented by pooling the features inside it. As the cardinality potential, we use the Normal model in (5.3) with $\mu = 1$ and $\sigma = 0.1$ to embed a soft and intuitive constraint on the number of positive instances: "the more relevant segments in a video, the higher the probability of occurring the event".

We follow the evaluation protocol used in [95, 90]. The DEV-T split of MED11 dataset is used for validation and finding the hyper-parameters such as the regularization weights in learning the cardinality model and SVM. Then, we evaluate the methods on the DEV-O test collection (32061 videos), containing the events 6 to 15 and a large number of null (or background events). For training, an Event-Kit collection of roughly 150 videos per event is used, and as in [95, 90], the classifiers are trained for each event versus all the others.

We compare our methods with the kernelized latent SVM methods in [95], applied to a structured model where the temporal location and scene type of the salient video segments are modeled as latent variables. To have a fair comparison, we use the same set of features: HOG3D, sparse SIFT, dense SIFT, HOG2x2, self-similarity descriptors (ssim), and color histograms, which are simply concatenated to a single feature vector[5]. For training the cardinality model, regularized maximum likelihood is used with l_1-norm regularization. For the Cardinality Kernel, histogram intersection kernel is plugged as the instance kernel. The results in terms of average precision (AP) are shown in Fig. 5.5. It can be observed that based on mean AP, our proposed Cardinality Kernel clearly outperforms the baselines:

- The cardinality model of Sec. 5.3.1.
- Kernelized SVM (KSVM) and multiple kernel learning SVM (MKL-SVM), which are kernel methods with global bag-of-words models.
- MI-Kernel [39], which is a multi-instance kernel method with uniform aggregation of the instance kernels.

On the other hand, our method is comparable to the kernelized latent SVM (KLSVM) methods in [95]. However, our model is considerably less complicated, and unlike these methods, our proposed framework has exact and efficient inference and learning algorithms. For example the training time for our method is about 35 minutes per event, but those methods takes about 30 hours per event[6]. In addition, based on comparison on individual events, our proposed method achieves the best AP in 6 out of 10 events.

Recently, Lai et al. [60] proposed a multi-instance framework for video event detection, by treating a video as a bag of temporal video segments of different granularity. Since this is the closest work to ours, we run another experiment on TRECVID MED11 to evaluate

[5]In the experiments of this section we compare our method with the most relevant methods, which use the same features. By using, combining, or fusing other sets of features, better results can be achieved (e.g. [91, 112])

[6]We performed our experiments on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, and compared to our previous work [95].

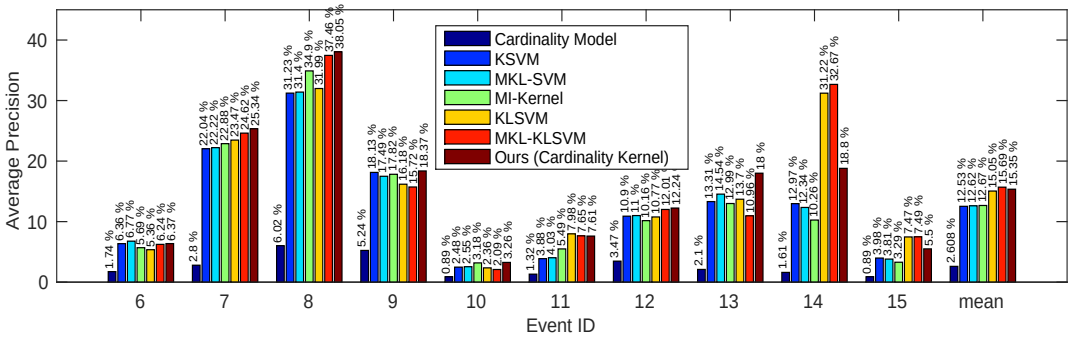


Figure 5.5: The APs for events 6 to 15 in TRECVID MED 2011. The results for KSVM, MKL-SVM, KLSVM, and MKL-KLSVM are reported from [95]. MI-Kernel is based on our own implementation of the algorithm in [39].

performance of our algorithm compared to [60]. We use exactly the same settings as before, but since Lai et al. [60] used dense SIFT features, we also extract dense SIFT features quantized into a 1500-dimensional bag-of-words vector for each video segment[7], where the video segments are given by dividing each video into 10 equal parts. This is slightly different from the multi-granular approach in [60], where both the single frames and temporal video segments are used as the instances (single–g $\propto$SVM uses only single frames and multi–g $\propto$SVM uses both the single frames and video segments). The results are shown in Table (5.2). Our method outperforms multi–g $\propto$SVM (which is the best in [60]) by around 20%. In addition, our algorithm is more efficient, and training takes only about half an hour per event.

Table 5.2: Comparing our proposed Cardinality Kernel method with $\propto$SVM algorithms in [60] on TRECVID MED11. The best AP for each event is highlighted in bold

Event	single–g $\propto$SVM [60]	multi–g $\propto$SVM [60]	Cardinality Kernel
6	1.9 %	**3.8** %	2.8 %
7	2.6 %	**5.8** %	**5.8** %
8	11.5 %	11.7 %	**17.0** %
9	4.9 %	5.0 %	**8.8** %
10	0.8 %	0.9 %	**1.3** %
11	1.8 %	2.4 %	**3.4** %
12	4.8 %	5.0 %	**10.7** %
13	1.7 %	2.0 %	**4.7** %
14	10.5 %	**11.0** %	4.9 %
15	2.5 %	**2.5** %	1.4 %
mAP	4.3 %	5.0 %	**6.1** %

[7]We use VLFeat, as in [60], though with fewer codewords (5000 in [60]).

5.4.3 Video Summarization by Detecting Interesting Video Segments

Recently, Gygli et al. [45] proposed a novel method for creating summaries from user videos by selecting a subset of video segments, which are interesting and informative. For this purpose, they created a benchmark dataset (SumMe[8]) of 25 raw user videos, summarized and annotated by 15 to 18 human subjects. In their proposed method, each video segment is scored by summing the *interestingness* score of its frames, estimated by a regression model learned from human annotations. At the end, a subset of video segments is selected such that the summary length is 15% of the input video.

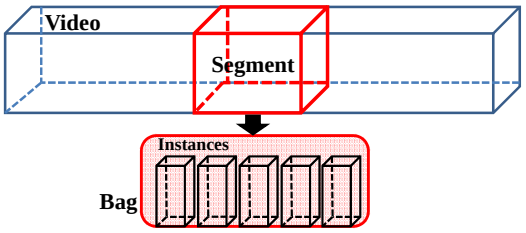


Figure 5.6: Detecting interesting video segments. A video is modeled as a bag of sub-segments.

In this section, we propose a new approach for creating segment-level summaries. Instead of predicting the per-frame scores and using a heuristic aggregation operation such as "sum", we use our multi-instance model to directly estimate the interestingness of a video segment. The proposed approach is illustrated in Fig. 5.6. Each segment is modeled as a bag of sub-segments, where a positive bag is a segment which has large overlap with human annotated summaries. To represent each sub-segment, we extract HSV color histogram (with 8×8 bins) and bag-of-words dense trajectory features [101] (with 4000 words) for each frame and max-pool the features over the sub-segment. Here, we summarize our method and the baselines:

- Ours: A segment is divided into 5 sub-segments, and the proposed Cardinality Kernel with Normal cardinality potential ($\mu = 1, \sigma = 0.1$) is used to score the segments.
- Global Model: A global representation of each segment is constructed by max-pooling the features inside it, and an SVM is trained on the segments.
- Single-Frame SVM: An SVM is trained on the frames, and the score of each segment is estimated by summing the frame scores.
- Single-Frame SVR: This is our simulation of the algorithm in [45] but with our own features, fixed length segments, and using support vector regression.

[8]The dataset and evaluation code for computing the f-measure are available at `http://www.vision.ee.ethz.ch/~gyglim/vsum/`

The top scoring 15% of segments are selected in each.

For all methods a video is segmented into temporal segments of length $P_l = 1.85$ seconds (the segment length given in [45]), and histogram intersection kernel is used for training the SVMs. To evaluate the methods, the procedure in [45] is used: leave-one-out validation and comparison based on per segment f-measure. The results are shown in Fig. 5.7. It can be observed that our method outperforms the baselines and is competitive with the state-of-the-art results in [45]. In fact, although we are using general features (color histogram and dense trajectory) we achieve a performance which is comparable to the performance in [45], which uses specialized features to represent *attention*, *aesthetics*, *landmarks*, etc. Note that the best f-measure in [45] is obtained by over-segmenting a video into cuttable segments called *superframe*, using guidelines from editing theory.

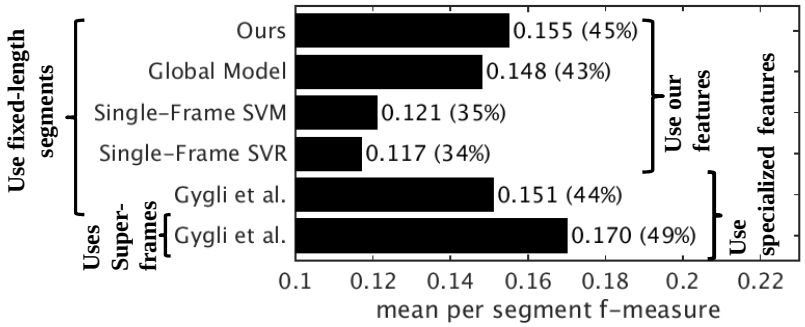


Figure 5.7: Comparison of different algorithms for segment-level summarization of the SumMe benchmark videos. The percent scores are relative to the average human.

5.5 Summary and Conclusion

We demonstrated the importance of cardinality relations in visual recognition. To this end, a probabilistic structured kernel method was introduced. This method is constructed based on a multi-instance cardinality model, which can explore different levels of ambiguity in instance labels and model different cardinality-based assumptions. We evaluated the performance of the proposed method on three challenging tasks: collective activity recognition, video event detection, and video summarization. The results showed that encoding cardinality relations and using a kernel approach with non-uniform (or probabilistic) aggregation of instances leads to significant improvement of classification performance. Further, the proposed method is powerful, straightforward to implement, with exact inference and learning, and can be simply integrated with off-the-shelf structured learning or kernel learning methods. As an extension to this work, in Appendix A, we show how to jointly learn the cardianlity kernel and SVM parameters in a general multiple kernel learning framework.

Chapter 6

Learning Ensemble Latent Structured Models in Functional Space

Many visual recognition tasks involve modeling variables which are structurally related. Hidden conditional random fields (HCRFs) are a powerful class of models for encoding structure in weakly supervised training examples. This chapter presents HCRF-Boost, a novel and general framework for learning HCRFs in functional space. An algorithm is proposed to learn the potential functions of an HCRF as a combination of abstract nonlinear feature functions, expressed by regression models. Consequently, the resulting latent structured model is not restricted to traditional log-linear potential functions or any explicit parameterization. Further, functional optimization helps to avoid direct interactions with the possibly large parameter space of nonlinear models and improves efficiency. As a result, a complex and flexible ensemble method is achieved for structured prediction which can be efficiently used in a variety of applications. We validate the effectiveness of this method on tasks such as group activity recognition, human action recognition, and multi-instance learning of video events.

6.1 Overview

Challenging structured vision problems necessitate the use of high-capacity models. Examples include problems such as modeling group activities or temporal dynamics in human action recognition and internet video analysis. Recently, visual recognition has made great strides using deep models. Deep learning has been successfully applied to image classification [57, 89] and object detection [41]. This success arises from large-scale training of highly non-linear functions which can induce complex models and learn powerful abstract feature representations. However, learning non-linear functions for structured vision problems re-

mains an open challenge. In this chapter, we present a general method to learn non-linear representations for structured models.

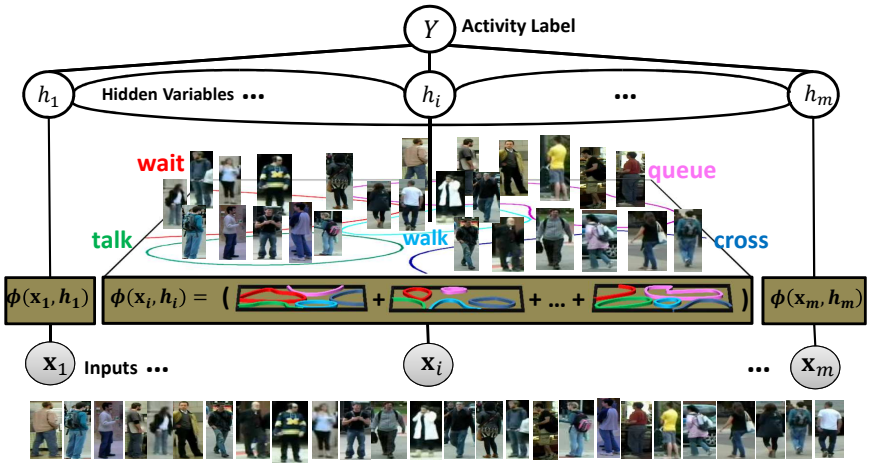


Figure 6.1: The proposed method (HCRF-Boost) learns non-linear potential functions in a latent structured model. An example model for group activity is shown. Potential functions relate input image regions to variables such as body pose or action/activity. Each potential function is learned as a combination of non-linear models leading to a high-capacity model. The colored ribbon-like lines show the decision boundaries obtained by nonlinear potential functions.

Our method works within a graphical model framework, building an HCRF to model structure, as depicted in Fig. 6.1. Recent efforts in this vein [94, 16, 88] have attempted to design unified deep structured models by equipping Markov random fields (MRFs) with the representational power of convolutional neural networks (CNNs). These methods jointly train an MRF and a CNN by maximizing likelihood via back-propagation and stochastic gradient descent. However, all these methods are defined for fully observed output variables and cannot incorporate or infer dependencies on unlabeled variables in the case of weak supervision. Full annotation of all output variables in MRFs is very costly for many visual recognition tasks, and hence many variables remain latent, unobserved, in training.

The standard learning algorithms for latent structured models (e.g. latent SVM [31] or HCRF [83]) are restricted to simple log-linear models, where the potential functions are parameterized by linear combination of the input features. Thus, they lack the non-linearity and feature abstraction power of deep models. In this work, we alleviate this problem by proposing a general framework to learn latent structured models with arbitrary potential functions in functional space.

We propose an algorithm based on functional gradient ascent (i.e., gradient boosting). By using this functional approach, training a latent structured model is decoupled from explicit representation of feature interactions in the potentially large parameter space of the potential functions. This provides scalability and improves efficiency [24]. This decoupling

helps to define potential functions as a combination of new abstract features encoded by nonlinear regression models such as regression trees, kernel support vector machines, or deep neural networks. As a result, a highly complex model can be achieved with an efficient learning algorithm. In addition, because of the *ensemble effect* of combining numerous base models, the proposed method is less prone to overfitting.

6.2 Previous Work

In this section, we review related work within learning algorithms for structured prediction and their use in computer vision.

Learning algorithms for structured prediction: Conditional random fields (CRFs) are among the primary tools for structured visual recognition. Nonlinear variants of CRFs include kernel conditional random fields [59], and CRFs with deep neural network features [27]. Dietterich et al. [24] and Chen et al. [17] proposed a boosting framework to train CRFs with abstract features represented by regression trees. Jancsary et al. [53] introduced regression tree fields, a Gaussian CRF model parameterized by regression trees. Tompson et al. [94], Chen et al. [16], Schwing and Urtasun [88] proposed methods to combine convolutional neural networks with CRF-based graphical models for deep structured prediction.

Hidden conditional random fields [83] learn CRFs with latent variables by maximizing the likelihood function marginalized over the hidden variables via gradient ascent. Max-margin variants of HCRF (a.k.a. latent SVM) [31, 117, 108] use alternating minimization strategies. Schwing et al. [87] introduced a general structured loss minimization framework for structured prediction with latent variables. All these algorithms are used for learning log-linear models, which limits their ability to model complex prediction tasks.

Nonlinear extensions of these algorithms have been proposed based on predefined kernels, e.g. kernelized latent SVM [114], kernels on CRFs [50], or non-linear feature encoding techniques [97]. However, the kernelized latent SVM methods have high computational complexity and lack efficient inference algorithms, resorting to enumeration over (single) latent variables. The CRF kernel method uses log-linear models trained similar to the standard HCRF [83].

In contrast, our work presents a general framework for learning latent structured models, which trains HCRFs with arbitrary potential functions represented by an ensemble of nonlinear base models. Thus, it can represent richer dependencies between the variables, be integrated with a variety of base models, and provide efficient learning and inference algorithms; empirically we show these can deliver superior recognition performance.

Structured prediction for group activity: Structured prediction has been extensively used in a variety of computer vision applications. A series of recent papers has focused on the problem of group activity recognition, inferring an activity that is performed by a

set of people in a scene. Choi et al. [21], Lan et al. [61], and Khamis et al. [54] devised models for spatial and temporal relations between the individuals involved in a putative interaction. Lan et al. [61] proposed latent CRF models with optimized graph structures for joint action-activity recognition. Amer et al. [2] proposed a hierarchical random field to jointly model temporal and frame-wise relations of video features describing an activity in a hierarchy of mid-level video representations.

Individual human action recognition: A variety of feature descriptors has been designed to extract discriminative spatio-temporal information from depth sequences. For example, Yang et al. [116] proposed new HOG descriptors built on depth motion maps. Wang et al. [106] trained an actionlet ensemble model based on novel local skeleton features to represent and recognize human actions. Xia and Aggarwal [111] introduced depth cuboid similarity features to make codewords for depth video recognition. Yang and Tian [115] proposed super normal vector (SNV) to describe a depth sequence with a codebook of polynormals obtained by clustering surface normals in the sequence. We perform empirical evaluation on action recognition from depth data, showing the efficacy of our learning approach.

Unconstrained internet video analysis: Structural models have been also successfully used for unconstrained internet video analysis. Methods to capture the temporal structure of high-level events need to be robust to the presence of irrelevant frames. Successful models include Tian et al. [93] and Niebles et al. [78], who extended latent variable models in the temporal domain. Vahdat et al. [95] composed a test video with a set of kernel matches to training videos. Tang et al. [91] effectively combined informative subsets of features extracted from videos to improve event detection. Bojanowski et al. [10] labeled videos with sequences of low-level actions. Pirsiavash and Ramanan [80] developed stochastic grammars for understanding structured events. Xu et al. [112] proposed a feature fusion method based on utilizing related exemplars for event detection. Lai et al. [60] applied multi-instance learning to video event detection by representing a video as multi-granular temporal video segments.

6.3 Proposed Method: HCRF-Boost

We propose a general framework for learning non-linear latent structured models. A high-level overview of our proposed method is as follows. We need to learn potential functions for a structured model over inputs, latent variables, and outputs. These potential functions control compatibilities between various settings of the variables – e.g. the relationships between image observations and their class labels. In order to model challenging problems, complex non-linear relationships between these variables are needed.

Figure 6.2 shows our proposed HCRF-Boost model. The potential functions are defined as a combination of multiple nonlinear functions, obtained stage by stage. To find these

functions we use functional gradient ascent (i.e. gradient boosting). Gradient boosting is the functional analog of the standard gradient ascent. At each step, a functional gradient is found by taking the derivatives of the objective function (likelihood function in our case) directly w.r.t. the potential functions (instead of the parameters). So, at each step a new function g_t is derived, where the potential function should move in that functional direction. In this section, we show how to take these derivatives efficiently and approximate the functional gradients with nonlinear fitting functions. In the following sections the preliminaries and details of the proposed method are explained. A summary of the resulting algorithm is given in Alg. 2.

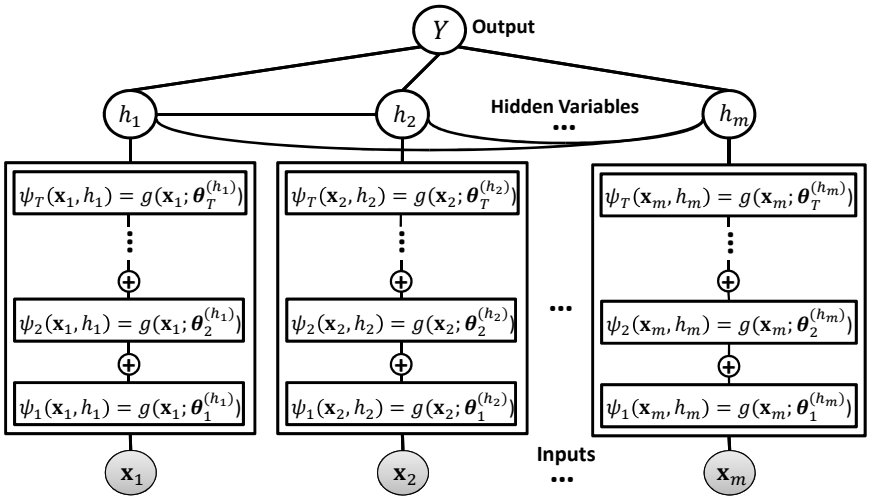


Figure 6.2: Latent structured prediction with our proposed HCRF-Boost model.

6.3.1 Preliminaries

Due to space limitations, we provide very brief summaries of gradient boosting [35] and HCRFs [83] below. Please see the corresponding references for more details.

Gradient Boosting: Gradient boosting learns a classifier $F(\mathbf{x}) = \sum_t \beta_t f_t(\mathbf{x})$ by optimizing an objective function $\mathcal{L}(y, F(\mathbf{x}))$ in a functional space by performing gradient ascent. The optimization is approximated by a greedy stage-wise optimization of the form

$$(\beta_t, f_t) = \arg\min_{\beta, f} \sum_{n=1}^{N} \mathcal{L}(y^n, F_{t-1}(\mathbf{x}^n) + \beta\, f(\mathbf{x}^n)). \tag{6.1}$$

using a training set $\{(\mathbf{x}^1, y^1), \cdots, (\mathbf{x}^N, y^N)\}$. To solve this problem, first the so-called pseudo-residuals are computed for each training instance as

$$\widehat{f}(\mathbf{x}^n) = \frac{\partial \mathcal{L}(y^n, F(\mathbf{x}^n))}{\partial F(\mathbf{x}^n)}|_{F(\mathbf{x})=F_{t-1}(\mathbf{x})} \tag{6.2}$$

After computing the pseudo-residuals, a new base classifier $f_t(\mathbf{x})$ is trained by fitting a regression model to the training set $\{(\mathbf{x}^n, \widehat{f}(\mathbf{x}^n))\}_n$, i.e., $f_t : \mathbf{x}^n \rightarrow \widehat{f}(\mathbf{x}^n)$. Given this function fixed, the multiplier β_t is found simply by doing a line search. It has been shown that since a whole model is added at each iteration of gradient boosting, a big step can be taken to maximize the objective function [24].

Hidden Conditional Random Fields: A hidden conditional random field (HCRF) is defined on a 3-tuple ($\mathbf{X} \in \mathcal{X}, \mathbf{h} \in \mathcal{H}, Y \in \mathcal{Y}$), where $\mathbf{h}$ is the set of latent variables, which are not observed in the training data. Given this, the posterior probability distribution is obtained by

$$P(Y|\mathbf{X}) = \sum_{\mathbf{h}} P(Y, \mathbf{h}|\mathbf{X}) = \frac{\sum_{\mathbf{h}} \exp\left(F(\mathbf{X}, Y, \mathbf{h})\right)}{\sum_{Y', \mathbf{h}} \exp\left(F(\mathbf{X}, Y', \mathbf{h})\right)}, \tag{6.3}$$

where the whole graph potential function factorizes as

$$F(\mathbf{X}, Y, \mathbf{h}) = \sum_i \phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i). \tag{6.4}$$

In the standard HCRF model proposed by [83], the potential functions are linearly parameterized as

$$\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \boldsymbol{\gamma_i}(\mathbf{X}_i, Y_i, \mathbf{h}_i)\, \boldsymbol{\theta}_i. \tag{6.5}$$

and parameters are learned using maximum a posteriori estimation.

In this work, we alleviate the limitation of parameterizing the HCRFs and learn the potential functions in a functional space, using a boosting approach. As a result, highly non-linear and powerful models can be achieved.

6.3.2 HCRF-Boost: Gradient Boosting of HCRFs

Gradient boosting [35] is a non-parametric functional analog of gradient ascent. In this approach, the derivatives of an objective function are taken with respect to the function, and each step of the gradient boosting is regarded as training a new base learner.

In this work, we use gradient boosting for training HCRF models. For this purpose, we maximize the likelihood function in (6.3) directly with respect to the clique potential functions. Consequently, each potential function is written as the combination of a number of "base potential functions":

$$\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \sum_t \beta_t\, \psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i), \tag{6.6}$$

where each base potential function is estimated in a stagewise manner by taking the derivatives of the log likelihood function w.r.t. the potential functions (given the current model estimation):

$$\widehat{\psi}_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \frac{\partial \log P(Y|\mathbf{X})}{\partial \phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i)}|_{f=f_{t-1}}. \tag{6.7}$$

We call this the pseudo-residual potential function. By plugging into the likelihood function of (6.3) and using the relations in [24], we get the following functional gradients at a given point $(\mathbf{X}^n, Y^n)$:

$$\begin{aligned}\widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i) = &\frac{\partial \log \sum_{\mathbf{h}} \exp\left(f(\mathbf{X}^n, Y^n, \mathbf{h})\right)}{\partial \phi_i(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)} \\ &- \frac{\partial \log \sum_{Y', \mathbf{h}} \exp\left(f(\mathbf{X}^n, Y', \mathbf{h})\right)}{\partial \phi_i(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)} \\ =&P(\mathbf{h}_i|\mathbf{X}^n, Y^n)\mathbb{1}(Y_i = Y_i^n) - P(\mathbf{h}_i, Y_i|\mathbf{X}^n) \\ \forall\, i, Y_i, \mathbf{h}_i.&\end{aligned} \tag{6.8}$$

Given the finite training set $\mathcal{D}^{tr} = \{(\mathbf{X}^n, Y^n)\}_{n=1}^N$ these are point-wise functional gradients, which are only defined at the training data points [35]. However, they provide a set of functional gradient training examples $\mathcal{D}_{i,t}^{(Y_i,\mathbf{h}_i)} = \{(\mathbf{X}^n, Y^n), \widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)\}_n$, which can be fitted by a regression model in order to make smooth approximate pseudo-residual potential functions:

$$\begin{aligned}&\psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \arg\min_{\psi_i} \sum_n \Big(\psi_i(\mathbf{X}_i^n, Y_i, \mathbf{h}_i) - \widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)\Big)^2 \\ &\forall\, i, Y_i, \mathbf{h}_i.\end{aligned} \tag{6.9}$$

This fitting is done by learning the parameters of a regression model for each possible value of the output and hidden variables, i.e.,

$$\begin{aligned}&\psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i) = g(\mathbf{X}_i; \boldsymbol{\theta}_{i,t}^{(Y_i,\mathbf{h}_i)}), \\ &\boldsymbol{\theta}_{i,t}^{(Y_i,\mathbf{h}_i)} = \arg\min_{\boldsymbol{\theta}} \sum_n \Big(g(\mathbf{X}_i^n; \boldsymbol{\theta}) - \widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)\Big)^2 \\ &\forall\, i, Y_i, \mathbf{h}_i.\end{aligned} \tag{6.10}$$

Hence, in the most general case, the number of trained models can grow exponentially with the number of variables in the largest clique. However, in practice, where common HCRF models are used, this procedure is reduced to training a few models (see next section). Finally, given the resulting functions, the potential function at the current iteration is updated as

$$\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) \leftarrow \phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) + \beta_t\, \psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i), \tag{6.11}$$

where the the step-length parameter β_t can be found by optimizing the likelihood function with a simple line search[1].

[1]However, there is both theoretical and empirical evidence that this parameter can be safely set to a small constant value (e.g., 0.1) [11]. In all our experiments, we follow this rule.

Algorithm 2 HCRF-Boost Algorithm

Input: Training data $\{(\mathbf{X}^n, Y^n)\}_{n=1}^N$.
Initialize the potential functions $\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \mathbf{0}$.
repeat
  for each potential function ϕ_i **do**
    Compute the pseudo-residual potentials $\widehat{\psi}_{i,t}$ according to (6.8) for all training examples.
    Train new base potential functions $\psi_{i,t}$ according to (6.10) by fitting the input training examples to the pseudo-residual potentials.
    Update the potential function: $\phi_i \leftarrow \phi_i + \beta_t \psi_{i,t}$.
  end for
until converged or maximum number of iterations

6.3.3 HCRF-Boost for Unary and Pairwise Potentials

In the previous section, we described the HCRF-Boost algorithm for general HCRF models. In this section, a more detailed explanation of the algorithm is provided for HCRF models with unary and pairwise potentials, which are commonly used in visual recognition [83].

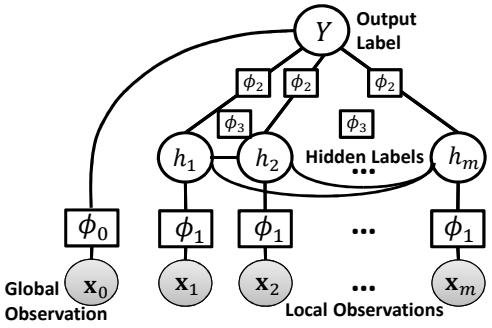


Figure 6.3: A hidden conditional random field with unary and pair-wise potential functions.

A graphical representation of this model is shown in Figure 6.3. This graph is composed of the input observations $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_m\}$, the output label Y, and the hidden labels $\mathbf{h} = \{h_1, \cdots, h_m\}$. The input observations are feature descriptors extracted from an image or video, where $\mathbf{x}_0$ is a global feature descriptor which represents the whole input, while $\mathbf{x}_i\,(i \neq 0)$ are local observations. Each local observation $\mathbf{x}_i$ is connected to its hidden label h_i. The connections between the hidden labels is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edges $(i, j) \in \mathcal{E}$ denote the links between the hidden labels h_i and h_j. Finally, all hidden labels are linked to the output label Y. The goal is to predict the output label Y, given the input observations $\mathbf{X}$ and the structural constraints of the induced graph, by modeling the posterior probability $P(Y|\mathbf{X})$.

Given this model, the whole graph potential function takes the following form:

$$f(\mathbf{X}, Y, \mathbf{h}) = \phi_0(\mathbf{x}_0, Y) + \sum_{i=1}^{m} \phi_1(\mathbf{x}_i, h_i) + \sum_{i=1}^{m} \phi_2(Y, h_i) + \sum_{(i,j)\in\mathcal{E}} \phi_3(Y, h_i, h_j). \tag{6.12}$$

The learning process is to find the potential functions $\phi_0, \phi_1, \phi_2, \phi_3$ which maximize the likelihood function, by taking the functional gradients. Following the formula derived in (6.6), the pseudo-residuals of each potential function for a given data point $(\mathbf{X}^n, Y^n)$ at iteration t are obtained by[2]:

$$\widehat{\psi}_{0,t}(\mathbf{x}_0^n, Y) = \mathbb{1}(Y = Y^n) - P(Y|\mathbf{X}^n) \tag{6.13}$$

$$\widehat{\psi}_{1,t}(\mathbf{x}_i^n, h_i) = P(h_i|\mathbf{X}^n, Y^n) - P(h_i|\mathbf{X}^n) \tag{6.14}$$

$$\widehat{\psi}_{2,t}(Y, h_i) = P(h_i|\mathbf{X}^n, Y^n)\mathbb{1}(Y = Y^n) - P(h_i, Y|\mathbf{X}^n) \tag{6.15}$$

$$\widehat{\psi}_{3,t}(Y, h_i, h_j) = P(h_i, h_j|\mathbf{X}^n, Y^n)\mathbb{1}(Y = Y^n) - P(h_i, h_j, Y|\mathbf{X}^n) \tag{6.16}$$

$$\forall\, i \in \mathcal{V}, (i,j) \in \mathcal{E}, Y \in \mathcal{Y}, h_i \in \mathcal{H}.$$

Note that all these probabilities are the marginal probabilities which can be found by sum-product inference of the CRFs. For the popular CRF models that we use in our experiments, such as tree-structured graphs or cardinality models, these marginals can be inferred exactly in linear or linearithmic time.

[2]Although the behind-the-scenes steps to derive the functional gradients are non-trivial, the results are intuitive. For example, (6.13) says that if Y is observed in the training data (i.e., $Y = Y^n$), $P(Y|\mathbf{X}^n)$ should be equal to 1 to make the subgradients zero and maximize the likelihood. Likewise, (6.14) says that the probability of the latent variables, with and without Y being observed, should become equal. In fact, these functional gradients are representing the errors but on a probability scale.

Next, by solving the fitting problem of (6.10), it can be shown that the smooth approximate functions are found as

$$\psi_{0,t}(\mathbf{x}_0, Y = a) = g(\mathbf{x}_0, \boldsymbol{\theta}_{0,t}^{(a)}) : \{\mathbf{x}_0^n \rightarrow \widehat{\psi}_{0,t}(\mathbf{x}_0^n, a)\}_{\mathcal{D}^{tr}}$$
$$\forall\, a \in \mathcal{Y} \tag{6.17}$$
$$\psi_{1,t}(\mathbf{x}_i, h_i = b) = g(\mathbf{x}_i, \boldsymbol{\theta}_{1,t}^{(b)}) : \{\mathbf{x}_i^n \rightarrow \widehat{\psi}_{1,t}(\mathbf{x}_i^n, b)\}_{\mathcal{D}^{tr},\mathcal{V}}$$
$$\forall\, b \in \mathcal{H} \tag{6.18}$$
$$\psi_{2,t}(Y = a, h_i = b) = \texttt{mean}\,\{\widehat{\psi}_{2,t}(a, b)\}_{\mathcal{D}^{tr},\mathcal{V}}$$
$$\forall\, a \in \mathcal{Y}, b \in \mathcal{H} \tag{6.19}$$
$$\psi_{3,t}(Y = a, h_i = b, h_j = c) = \texttt{mean}\,\{\widehat{\psi}_{3,t}(a, b, c)\}_{\mathcal{D}^{tr},\mathcal{E}}$$
$$\forall\, a \in \mathcal{Y}, b \in \mathcal{H}, c \in \mathcal{H}. \tag{6.20}$$

The first set of functions in (6.17) and (6.18) are trained by a regression model. So, only $|\mathcal{Y}| + |\mathcal{H}|$ functions should be trained. However, the next functions in (6.19) and (6.20) are simply obtained by taking the mean over all training examples.

6.3.4 Discussion

The fitting in (6.17) and (6.18) can be performed by training any regression model such as regression trees, kernel support vector machines, or even deep neural networks. In practice training a support vector regression (SVR) model is faster than trees (especially for large feature vectors). Thus, in all our experiments we used SVR models. However, note that tree models can help for feature selection as well.

Further, for all the visual recognition tasks in Section 6.4, we use task-specific handcrafted features. But, by using convolutional neural networks (CNNs) for model fitting, deep features can be also learned. In fact, employing CNNs with our method leads to an extension of the recent algorithms for learning deep structured models [88, 16]. These algorithms maximize the likelihood function $P(Y|\mathbf{X}, \mathbf{w}) = \frac{\exp\left(f(\mathbf{X},Y,\mathbf{w})\right)}{Z(\mathbf{X},\mathbf{w})}$ w.r.t. the parameters $\mathbf{w}$ via gradient ascent and backpropagation, where $f(\mathbf{X}, Y, \mathbf{w})$ is a CNN parameterized by $\mathbf{w}$. However, HCRF-Boost with CNNs extends these algorithms by (1) incorporating the structured hidden variables and (2) learning via functional gradient ascent (i.e. gradient boosting).

6.3.5 Some Implementation details

In our implementation, we used *stochastic gradient boosting* [36]. In this variation of gradient boosting, at each step, a random subset of training data is selected for computing the pseudo-residuals and fitting the base models. As a result, gradient boosting is combined with bagging (similar to random forest). The incorporation of this randomization is advantageous

for both improving the accuracy and speeding up the algorithm [36]. In all the experiments we subsampled 90% of data (without replacement) at each iteration.

In the proposed HCRF-Boost algorithm, the potential functions may be initialized to zero at the first iteration. However, because of the nonconvexity of the likelihood optimization problem, a more smart initialization can improve the results (Note that the stochastic gradient ascent algorithm already helps to avoid some local optima). In our empirical studies we found that initializing the potentials with a model poorly trained by the standard HCRF algorithm [83] or with a global model trained by SVM can yield decent results in a few iterations (even 10 iterations). In fact, since each iteration of gradient boosting adds an entire model, a big step can be taken at each iteration [24]. In all experiments of Section 6.4.1 we used 50 iterations with $\beta = 0.1$.

6.3.6 Computational Complexity

The computational complexity of each iteration of gradient boosting comprises two parts: (1) computing point-wise pseudo-residuals and (2) training the base models. As discussed in Section 6.3.3, the former is obtained by inferring the CRF model for each data point and finding the marginal probabilities. We indicate the computational time of inferring the marginals of a CRF by T_{infer}. For example, for the tree/chain-structured CRF models of Section 6.4.1 and 6.4.2, $T_{infer} = O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ using belief propagation. For the cardinality model used in Section 6.4.3, $T_{infer} = O(m \log(m))$, where m is the number of instances in a bag. Consequently, the total computational time of this part is obtained by summing over the whole data: $\sum_n T^n_{infer}$.

Next, the base models should be fitted to the point-wise pseudo-residuals. We assume a regression model can be trained to fit a set of training examples of size $|S|$ in $O(|S|\, d)$ time, where d is the size of the input feature vector. Given this assumption, each function approximation in (6.17) and (6.18) takes $O(N\, d)$ and $O(\sum_n |\mathcal{E}^n|\, d)$ time, respectively. Finally, the computational time of fitting all functions would be $|\mathcal{Y}|\, O(N\, d) + |\mathcal{H}|\, O(\sum_n |\mathcal{E}^n|\, d)$.

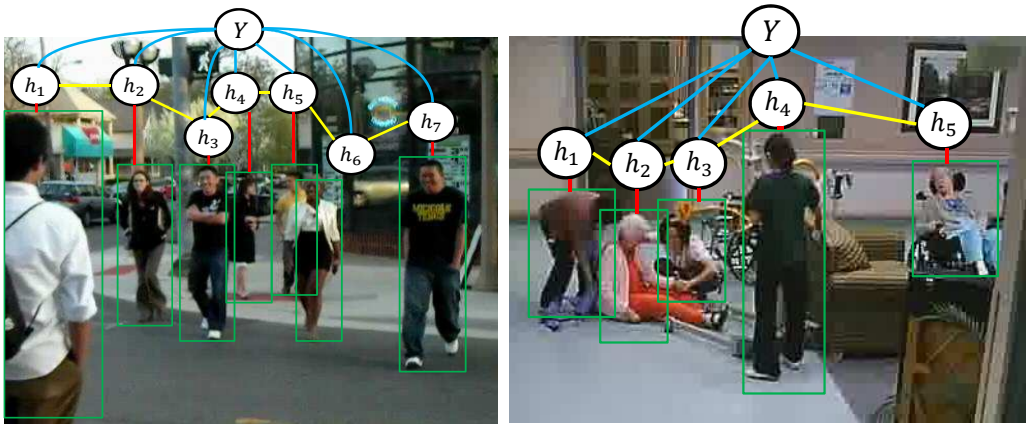
We performed our experiments on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. As a numerical example, for the collective activity dataset, used in the experiments of Section 6.4.1 (which consists of 1908 training examples with average 5 local observations per example and the feature vectors are 240 dimensional), the training time was around 10 seconds per iteration. For the nursing home dataset of Section 6.4.1 with 1910 training examples and average 2 local obsevations per example and 5D feature vectors, the trainig time was around 2 seconds per iteration.

6.4 Experiments

We provide empirical results on three different tasks: group activity recognition, human action recognition, and video event detection.

6.4.1 Spatial Structured Models: Group Activity Recognition

In this section, our proposed HCRF-Boost algorithm is used to train HCRFs which model spatial relations between individuals doing actions in a scene to recognize high-level group activities. Hence, the individual actions provide the context to infer the whole group activity. We run experiments on two datasets: collective activity dataset [21] and nursing home dataset [61]. Example HCRF models for this task are shown in Figure 6.4. This model is composed of nodes representing the people, actions, and the group activity. The hidden nodes are the individual actions which are linked to each other with a tree-structured graph, obtained by running maximum spanning tree.



(a) Collective activity recognition (b) Nursing home fall detection

Figure 6.4: Group activity recognition with spatial structured models. (a) An example HCRF model from collective activity dataset. (b) An example HCRF model from nursing home dataset.

Collective Activity Dataset

The Collective Activity Dataset [21] comprises 44 videos (about 2500 video frames) of *crossing*, *waiting*, *queuing*, *walking*, and *talking*. Our goal is to classify the collective activity in each frame. Each person is represented by the *action context* feature descriptor proposed in [61]. We follow the same experimental settings as used in [61], i.e., the same 1/3 of the video clips were selected for test and the rest for training. As the latent models, we use the HCRF shown in Figure 6.4a with 5 hidden labels. The result of our method is shown in Table 6.1 and compared with the following methods[3]: (1) SVM on global bag-of-words, (2) latent SVM method in [61], and (3) HCRF (our own baseline). We also visualize some examples of recognition with our method in Figure 6.5.

[3]These methods follow the standard multiclass classification evaluation protocol in [21, 61].

Table 6.1: Comparison of classification accuracies of different algorithms on collective activity dataset. Both multi-class accuracy (MCA) and mean per-class accuracy (MPCA) are shown because of class size imbalance.

Method	MCA	MPCA
Global bag-of-words with SVM [61]	70.9	68.6
Latent SVM with optimized graph [61]	79.7	78.4
HCRF	76.2	75.2
HCRF-Boost (our proposed method)	**82.5**	**79.4**

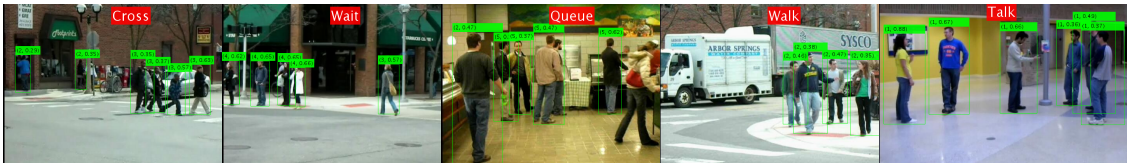


Figure 6.5: Examples of recognition with the proposed HCRF-Boost method. Each figure is annotated by the predicted collective activity. Also each individual is annotated by a tuple, indicating the inferred hidden label and its probability. Since the hidden labels are not observed during training, they have been represented symbolically by 1, 2, 3, 4, 5. However, interestingly, they have been learned to semantically categorize the individual actions (i.e., 1: talk; 2: walk; 3: cross; 4: wait; 5:queue). For example, in the first figure from left, four people are crossing the street while the two others are walking in the sidewalk. In the second figure, four people are waiting and one is crossing. In the third figure, four people are queuing in the line and one person is walking to join the lineup. In the fourth and fifth figures, all the individuals are walking and talking, respectively.

Nursing Home Dataset

In this section, we evaluate our method for activity recognition in a nursing home. The dataset we use [61] contains scenes in which the individuals might be performing any of five actions: walking, standing, sitting, bending, or falling. However, the goal is to detect the whole scene activity, i.e., if any person is falling or not.

The dataset has 22 video clips (12 clips for training and 8 clips for test) with 2990 annotated frames, where about one third of them are assigned the "fall" activity label. We use the same feature descriptor as used in [61]. In short, this feature vector is obtained by concatenating the score of SVM classifiers trained for recognizing each of the five actions on the training dataset. Similar to the previous section, we use the HCRF model shown in Figure 6.4b with five hidden labels. The results in terms of classification accuracy and average precision are shown in Table 6.2. Again, we compare our method with a global bag-of-words model, latent SVM, and standard HCRF algorithm.

Table 6.2: Comparison of different algorithms on the nursing home dataset in terms of average precision (AP), mean per-class accuracy (MPCA), and multi-class accuracy (MCA). Note that because of the significant class size imbalance between the two classes, MCA is not an informative metric in this task

Method	AP	MPCA	MCA
Global bag-of-words [61]	43.3	52.4	48.0
Latent SVM [61]	48.8	67.4	71.5
HCRF	44.4	66.3	75.2
HCRF-Boost (ours)	**49.6**	**73.0**	**75.4**

6.4.2 Temporal Structured Models: Human Action Recognition

In this section, we apply our method for human action recognition with chain-structured HCRFs, capturing the temporal dynamics of the action. A graphical model of this task is illustrated in Figure 6.6. This HCRF consists of the input nodes, representing temporal segments of a depth sequence, connected to the hidden-state nodes. There is also a root potential function to globally model the interaction between the whole action sequence and the action label.

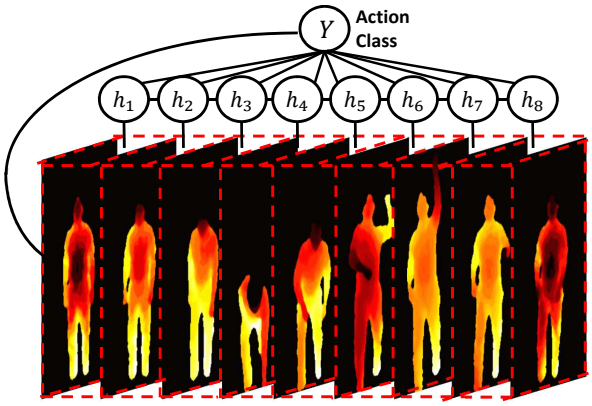


Figure 6.6: The HCRF model for human action recognition from a depth sequence.

We evaluate the proposed model on the MSRAction3D dataset [68]. This dataset has 567 depth map sequences of 20 different actions performed by 10 subjects. The actions are movements common in gaming such as “hand catch", “forward punch", “draw tick", “tennis swing". As the features, we use the super normal vector (SNV) descriptors [115]. But, instead of the raw SNV features, we convert them into SVM scores and make a discriminative feature descriptor, as in Section 6.4.1.

The experiments were conducted by dividing each depth sequence into eight equal temporal segments and using the HCRF model of Figure 6.6 with 5 hidden states for each segment. To have a fair comparison we followed the same experimental protocol as [115, 106].

The results are shown in Table 6.3 and compared with the state-of-the art methods for depth-based action recognition. Note that the global model[4] and HCRF algorithm are our own baselines.

Table 6.3: Comparison of classification accuracies of different algorithms on MSRAction3D dataset.

Method	Accuracy
Bag of 3D Points [68]	74.70%
Random Occupancy Pattern [105]	86.50%
Actionlet Ensemble [106]	88.20%
Depth Motion Maps [116]	88.73%
DSTIPv [111]	89.30%
Skeletal [98]	89.48%
Pose Set [100]	90.00%
Moving Pose [119]	91.70%
DMM-LBP-DF [15]	93.0%
SNV [115]	93.09%
Our global model (using SNV)	92.73%
HCRF (using SNV)	91.64%
HCRF-Boost (using SNV)	**94.18%**

6.4.3 Cardinality Models for Multi-Instance Learning: Multimedia Event Detection

Multiple instance learning (MIL) aims to recognize patterns from weakly supervised data. Contrary to standard supervised learning, where each training instance is labeled, in the MIL paradigm a *bag of instances* share a label, and the instance labels are hidden. In Chapters 4 and 5, we introduced HCRF models for MIL by incorporating cardinality-based potential functions. These cardinality potentials permit the modeling of the counts of inputs that contribute to an overall label.

A graphical representation of the cardinality model is shown in Figure 6.7. Each instance and its label are modeled by two nodes in a clique. The potential function of this clique (ϕ_I) specifies a classifier for an individual instance. There is also an optional clique potential between the global representation of the whole bag and the bag label (ϕ_B). Finally, a third clique potential (ϕ_C) contains all instance labels and the bag label. This clique is used to define what makes a bag positive or negative. Varying this clique potential will lead to different multi-instance assumptions. To this end, two different cardinality-based functions

[4]Our global model is the same as the model proposed in [115] for SNV. However, we could not get the same accuracy (92.73 vs 93.09) with our duplication of their experiments.

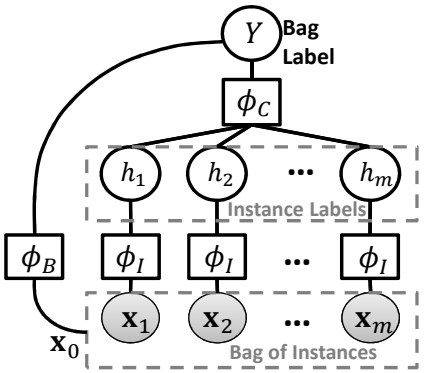


Figure 6.7: A graphical representation of the cardinality model. The instance labels are hidden variables.

are defined, one for positive bags ($C^{(+1)}$) and one for negative bags ($C^{(-1)}$):

$$\phi_C(Y, \mathbf{h}) = C^{(Y)}(\sum_i h_i). \tag{6.21}$$

In general, $C^{(+1)}$ and $C^{(-1)}$ could be expressed by any cardinality function which can model MIL constraints. However, in our work we focus on the Normal cardinality model:

$$\begin{aligned} C^{(+1)}(c) &= \left(-(\frac{c}{m} - \mu)^2/2\sigma^2\right), \\ C^{(-1)}(c) &= \left(-(\frac{c}{m})^2/2\sigma^2\right). \end{aligned} \tag{6.22}$$

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

In this work, we use our proposed HCRF-Boost to train these cardinality models. We evaluate our method for event detection on the challenging TRECVID MED11 dataset [79].

Recently, Lai et al. [60] proposed novel multi-instance methods (single–g $\propto$SVM and multi–g $\propto$SVM) for video event detection, by treating a video as a bag of temporal video segments of different granularity (single–g $\propto$SVM uses only single frames but multi–g $\propto$SVM uses both the single frames and video segments). In Chapter 5, we followed a similar MIL approach to video event detection by embedding the cardinality models into a powerful kernel, "Cardinality Kernel." We evaluate the performance of our HCRF-Boost algorithm compared to these methods. In our framework, each video is treated as a bag of ten temporal video segments, where each segment is represented by pooling the features inside it. As the cardinality potential, we use the Normal model in (6.22) with $\mu = 1$ and $\sigma = 0.1$ to embed a soft and intuitive constraint on the number of positive instances: "the more relevant segments in a video, the higher the probability of the event occurring".

Similar to the experiments in [60, 50], we use dense SIFT features quantized into bag-of-words vectors for each video segment[5]. The results are shown in Table (6.4). The HCRF method (used to train the cardinality model) performs poorly in this task because of using a linear feature representation. Our method outperforms multi–g $\propto$SVM (which is the best in [60]) by around 25%. It can be also observed that HCRF-Boost is comparable with the Cardinality Kernel method. Note that the Cardinality Kernel only induces nonlinearlity to bag classification and still has log-linear models for instance classification. Further, its computational complexity grows quadratically with the number of instances, and needs quadratic space w.r.t. the number of bags. However, HCRF-Boost is a general and flexible method, learns nonlinear potential functions, and provides scalability and efficiency.

Table 6.4: Comparing our proposed HCRF-Boost with $\propto$SVM algorithms in [60] and the Cardinality Kernel in on TRECVID MED11. The best AP for each event is highlighted in bold

Event	single–g $\propto$SVM [60]	multi–g $\propto$SVM [60]	Cardinality Kernel [50]	HCRF	HCRF-Boost
6	1.9 %	**3.8** %	2.8 %	1.2 %	2.6 %
7	2.6 %	**5.8** %	**5.8** %	1.8 %	5.3 %
8	11.5 %	11.7 %	17.0 %	9.7 %	**22.4** %
9	4.9 %	5.0 %	**8.8** %	3.0 %	6.3 %
10	0.8 %	0.9 %	**1.3** %	0.8 %	1.1 %
11	1.8 %	2.4 %	3.4 %	1.3 %	**3.7** %
12	4.8 %	5.0 %	10.7 %	4.0 %	**11.3** %
13	1.7 %	2.0 %	**4.7** %	0.8 %	**4.7** %
14	10.5 %	**11.0** %	4.9 %	1.4 %	3.7 %
15	2.5 %	**2.5** %	1.4 %	1.3 %	1.6 %
mAP	4.3 %	5.0 %	6.1 %	2.5 %	**6.3** %

6.5 Conclusion and Summary

We presented a novel and general framework for learning latent structured models. This algorithm uses gradient boosting to train a CRF with hidden variables in functional space. The functional approach helps to learn the structured model directly with respect to the potential functions without direct interaction with the potentially high-dimensional parameter space. By using this method, the potential functions are learned as an ensemble of nonlinear feature functions represented by regression models. This introduces nonlinearity into the model, enhances its feature abstraction and representational power, and finally reduces the chance of overfitting (due to the ensemble effect). We evaluated the performance of

[5]We use VLFeat, as in [60, 50], with the same number of codewords as [50] but with fewer codewords than [60] – 1500 for ours but 5000 in [60]). Note that this is not the best setting for the SIFT features. For example, if the codewords are increased to 20,000, the mean average precision is nearly doubled. Also by combining or fusing other sets of features, better results can be achieved (e.g. [91, 112]).

the proposed method on three challenging tasks: group activity recognition, human action recognition, and multimedia video event detection. The results showed that our nonlinear ensemble model leads to significant improvement of classification performance compared to the log-linear structured models. Further, the proposed method is very flexible and can be simply integrated with a variety of off-the-shelf nonlinear fitting functions.

Chapter 7

Conclusions and Future Directions

The primary focus of this dissertation was to propose novel and flexible frameworks for multiple instance learning. These frameworks can model a variety of multi-instance assumptions, including the standard assumption, ratio-constrained assumptions, probabilistic cardinality assumptions, linguistic assumptions, and metadata assumptions. It was shown that encoding these general multi-instance assumptions and cardinality constraints in visual recognition can improve recognition performance by either capturing the intrinsic relations in the problems or increasing robustness against clutter and ambiguity. We demonstrated the efficacy of the proposed frameworks in various applications such as image classification, human group activity recognition, human action recognition, cyclist helmet recognition, unconstrained video event detection, and video summarization.

In Chapter 3, a boosting framework was proposed which can softly explore different levels of ambiguity in multi-instance data using linguistic aggregation functions. Next, in Chapter 4, we introduced a class of probabilistic graphical models, namely multi-instance cardinality models, which can encode any cardinality-based relations between instance labels. They can also integrate the instance-level and bag-level information in a bag by capturing the interactions and dependencies between the local and global representations of the bag. Further, efficient and exact inference of the cardinality models makes it tractable to perform structured learning and prediction in real-world computer vision applications. We proposed novel learning algorithms for training the cardinality models. A latent max-margin methods was introduced in Chapter 4. Next, in Chapter 5, a kernel was defined on the cardinality models to classify bags in a discriminative vectorial embedding space. Finally, we proposed a gradient boosting algorithm for learning general hidden conditional random fields in Chapter 6. As a special case, the cardinality models can be trained by using this algorithm.

7.1 Future Directions

The following directions can be taken in the future to extend the proposed frameworks.

Integrating spatial, temporal and cardinality relations in a unified model: The proposed graphical model in Chapter 4 can be extended by embedding new potentials which capture spatial or temporal relations between instances. For example, the spatial potentials can be used to model the spatial relations between the individuals in an image. On the other hand, temporal potentials can be used to model the temporal relations between frames of a video. The only problem with adding these potentials is that the inference becomes more complicated. One solution would be to employ dual decomposition.

Deep cardinality models for multiple instance learning: Inspired by the recent success of deep learning methods [9] in computer vision applications, proposing deep multi-instance learning methods seems fruitful. Recently, Wu et al. [110] proposed deep MIL models. However, their models are not as general as our proposed cardinality models. One directions is to propose learning algorithms for deep hidden conditional random fields, which can be consequently used to train deep multi-instance cardinality models. Actually, our proposed HCRF-Boost algorithm in Chapter 6 can be easily integrated with deep models. However, we have not evaluated this method in an end-to-end learning problem yet.

HCRF Kernels for latent structured prediction: The proposed kernel in Chapter 5 can be extended to classify any HCRF (not necessarily cardinality models). The intuition is that instead of aggregating over instance kernels, the aggregation should be taken over clique kernels. By computing this kernel, the HCRF model is implicitly mapped to a highly discriminative and possibly infinite-dimensional space, where margin maximization and structured prediction can be performed easily. Note that the kernel learning algorithm proposed in Appendix A can be also extended to learn these new kernels. As a result, a kernel learning method is obtained for latent structured prediction. However, a major drawback to kernel methods is that they need quadratic space w.r.t. the size of training data.

Bibliography

[1] Tameem Adel, Ruth Urner, Benn Smith, Daniel Stashuk, and Daniel J Lizotte. Generative multiple-instance learning models for quantitative electromyography. In *Uncertainty in Artificial Intelligence (UAI)*, pages 1–11, 2013.

[2] Mohamed R. Amer, Peng Lei, and Sinisa Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *European Conference on Computer Vision (ECCV)*, 2014.

[3] Mohamed R. Amer and Sinisa Todorovic. A chains model for localizing participants of group activities in videos. In *International Conference on Computer Vision (ICCV)*, 2011.

[4] Mohamed R. Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *European Conference on Computer Vision (ECCV)*, 2012.

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

[6] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Neural Information Processing Systems (NIPS)*, pages 561–568. MIT Press, 2002.

[7] Boris Babenko. Multiple instance learning: algorithms and applications. `http://vision.ucsd.edu/~bbabenko/data/bbabenko_re.pdf`, 2008.

[8] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 983–990. IEEE, 2009.

[9] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[10] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision (ECCV)*, 2014.

[11] Peter Bühlmann and Torsten Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007.

[12] Razvan C. Bunescu and Raymond J. Mooney. Multiple instance learning for sparse positive bags. In *International Conference on Machine Learning (ICML)*, pages 105–112. ACM, 2007.

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

[14] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1-3):131–159, 2002.

[15] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Action recognition from depth sequences using depth motion maps-based local binary patterns. In *WACV*, 2015.

[16] Liang-Chieh Chen, Alexander G Schwing, and Raquel Urtasun. Learning deep structured models. *arXiv*, 2015.

[17] Tianqi Chen, Sameer Singh, Ben Taskar, and Carlos Guestrin. Efficient second-order gradient boosting for conditional random fields. In *AISTATS*, 2015.

[18] Yixin Chen, Jinbo Bi, and James Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 28(12):1931–1947, 2006.

[19] Yixin Chen and James Z Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[20] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision (ECCV)*, 2012.

[21] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *VS*, 2009.

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

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

[24] Thomas G Dietterich, Guohua Hao, and Adam Ashenfelter. Gradient tree boosting for training conditional random fields. *Journal of Machine Learning Research*, 9(10), 2008.

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

[26] Trinh-Minh-Tri Do and Thierry Artières. Large margin training for hidden markov models with partially observed states. In *International Conference on Machine Learning (ICML)*, pages 265–272. ACM, 2009.

[27] Trinh-Minh-Tri Do and Thierry Artières. Neural conditional random fields. In *AISTATS*, 2010.

[28] Trinh-Minh-Tri Do and Thierry Artières. Regularized bundle methods for convex and non-convex risks. *Journal of Machine Learning Research*, 13(1):3539–3583, 2012.

[29] Lin Dong. *A comparison of multi-instance learning algorithms*. PhD thesis, The University of Waikato, 2006.

[30] 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.

[31] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[32] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 32(9):1627–1645, 2010.

[33] James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *Knowledge Engineering Review*, 25(1):1, 2010.

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

[35] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *annals of statistics*, pages 1189–1232, 2001.

[36] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[37] Carolina Galleguillos, Boris Babenko, Andrew Rabinovich, and Serge Belongie. Weakly supervised object localization with stable segmentations. In *European Conference on Computer Vision (ECCV)*, pages 193–207. Springer, 2008.

[38] Thomas Gärtner. Kernel-based feature space transformation in inductive logic programming. *Master's thesis, University of Bristol*, 2000.

[39] Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-instance kernels. In *International Conference on Machine Learning (ICML)*, pages 179–186, 2002.

[40] Peter V. Gehler and Olivier Chapelle. Deterministic annealing for multiple-instance learning. In *AISTATS*, 2007.

[41] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[42] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision (ECCV)*, pages 634–647. Springer, 2010.

[43] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[44] R. Gupta, A.A. Diwan, and S. Sarawagi. Efficient inference with cardinality-based clique potentials. In *International Conference on Machine Learning (ICML)*, pages 329–336. ACM, 2007.

[45] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision (ECCV)*, pages 505–520. Springer, 2014.

[46] Hossein Hajimirsadeghi, Jinling Li, Greg Mori, Mohamed Zaki, and Tarek Sayed. Multiple instance learning by discriminative training of markov networks. In *UAI*, 2013.

[47] Hossein Hajimirsadeghi and Greg Mori. Multiple instance real boosting with aggregation functions. In *ICPR*, 2012.

[48] Hossein Hajimirsadeghi and Greg Mori. Learning ensembles of potential functions for structured prediction with latent variables. In *International Conference on Computer Vision (ICCV)*, 2015.

[49] Hossein Hajimirsadeghi and Greg Mori. Multi-instance classification by max-margin training of cardinality-based markov networks. In *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2015.

[50] Hossein Hajimirsadeghi, Wang Yan, Arash Vahdat, and Greg Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2596–2605, 2015.

[51] David Haussler. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

[52] Yuxiao Hu, Liangliang Cao, Fengjun Lv, Shuicheng Yan, Yihong Gong, and Thomas S Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *International Conference on Computer Vision (ICCV)*, pages 128–135. IEEE, 2009.

[53] Jeremy Jancsary, Sebastian Nowozin, Toby Sharp, and Carsten Rother. Regression tree fields – an efficient, non-parametric approach to image labeling problems. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2376–2383. IEEE, 2012.

[54] Sameh Khamis, Vlad I. Morariu, and Larry S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *European Conference on Computer Vision (ECCV)*, 2012.

[55] A. Khosla, R. Hamid, C. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[56] G. Kim, L. Sigal, and E. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Neural information processing systems (NIPS)*, 2012.

[58] James T Kwok and Pak-Ming Cheung. Marginalized multi-instance kernels. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 901–906, 2007.

[59] John Lafferty, Xiaojin Zhu, and Yan Liu. Kernel conditional random fields: representation and clique selection. In *International Conference on Machine Learning (ICML)*, 2004.

[60] K.-T. Lai, Felix X. Yu, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[61] Tian Lan, Yang Wang, Weilong Yang, Stephen N Robinovitch, and Greg Mori. Discriminative latent models for recognizing contextual group activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 34(8):1549–1562, 2012.

[62] Yuh-Jye Lee and Olvi L Mangasarian. Rsvm: Reduced support vector machines. In *International conference on data mining (ICDM)*, pages 5–7. SIAM, 2001.

[63] Christian Leistner, Amir Saffari, and Horst Bischof. Miforests: Multiple-instance learning with randomized trees. In *European Conference on Computer Vision (ECCV)*, pages 29–42. Springer, 2010.

[64] Thomas Leung, Yang Song, and John Zhang. Handling label noise in video classification via multiple instance learning. In *International Conference on Computer Vision (ICCV)*, pages 2056–2063. IEEE, 2011.

[65] Daxiang Li, Jing Wang, Xiaoqiang Zhao, Ying Liu, and Dianwei Wang. Multiple kernel-based multi-instance learning algorithm for image classification. *Journal of Visual Communication and Image Representation*, 25(5):1112–1117, 2014.

[66] F. Li and C. Sminchisescu. Convex multiple-instance learning by estimating likelihood ratio. *Neural Information Processing Systems (NIPS)*, pages 1360–1368, 2010.

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

[68] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–14. IEEE, 2010.

[69] Weixin Li and Vasconcelos Nuno. Multiple instance learning for soft bags via top instances. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4277–4285, 2015.

[70] Yan Li, David MJ Tax, Robert PW Duin, and Marco Loog. Multiple-instance learning as a classifier combining problem. *Pattern Recognition*, 46(3):865–874, 2013.

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

[72] 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.

[73] J. Malik, S. Belongie, T. K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[74] Olvi L Mangasarian and Edward W Wild. Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications*, 137(3):555–568, 2008.

[75] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. *Neural Information Processing Systems (NIPS)*, pages 570–576, 1998.

[76] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. In *Neural information processing systems (NIPS)*, 1999.

[77] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint âĎŞ2, 1-norms minimization. In *Advances in Neural Information Processing Systems*, pages 1813–1821, 2010.

[78] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision (ECCV)*, 2010.

[79] Paul Over, George Awad, Jon Fiscus, Brian Antonishek, Martial Michel, Alan F Smeaton, Wessel Kraaij, Georges Quénot, et al. Trecvid 2011-an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2011.

[80] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[81] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European Conference on Computer Vision (ECCV)*, 2014.

[82] N. Quadrianto, A. Smola, T. Caetano, and Q. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009.

[83] Ariadna Quattoni, Sybor Wang, L-P Morency, Michael Collins, and Trevor Darrell. Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 29(10):1848–1852, 2007.

[84] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, Yves Grandvalet, et al. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[85] S. Rueping. Svm classifier estimation from group probabilities. In *International Conference on Machine Learning (ICML)*, 2010.

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

[87] Alexander Schwing, Tamir Hazan, Marc Pollefeys, and Raquel Urtasun. Efficient structured prediction with latent variables for general graphical models. In *International Conference on Machine Learning (ICML)*, 2012.

[88] Alexander G Schwing, Alan L Yuille, and Raquel Urtasun. Fully connected deep structured networks. *arXiv*, 2015.

[89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

[90] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[91] Kevin Tang, Bangpeng Yao, Li Fei-Fei, and Daphne Koller. Combining the right features for complex event recognition. In *International Conference on Computer Vision (ICCV)*, 2013.

[92] Daniel Tarlow, Kevin Swersky, Richard Zemel, Ryan Adams, and Brendan Frey. Fast exact inference for recursive cardinality models. In *Uncertainty in Artificial Intelligence (UAI)*, 2012.

[93] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[94] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Neural information processing systems (NIPS)*, 2014.

[95] Arash Vahdat, Kevin Cannons, Greg Mori, Sangmin Oh, and Ilseo Kim. Compositional models for video event detection: A multiple kernel learning latent variable approach. In *International Conference on Computer Vision (ICCV)*, 2013.

[96] Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In *International Conference on Machine Learning*, pages 1065–1072. ACM, 2009.

[97] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 34(3):480–492, 2012.

[98] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, 2014.

[99] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Neural information processing systems (NIPS)*, pages 1417–1424, 2006.

[100] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[101] H. Wang, A. Kläser, C.Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[102] Hua Wang, Heng Huang, Farhad Kamangar, Feiping Nie, and Chris H Ding. Maximum margin multi-instance learning. In *Advances in Neural Information Processing Systems*, pages 1–9, 2011.

[103] Hua Wang, Feiping Nie, and Heng Huang. Learning instance specific distance for multi-instance classification. In *AAAI*, 2011.

[104] Hua Wang, Feiping Nie, and Heng Huang. Robust and discriminative distance for multi-instance learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2919–2924. IEEE, 2012.

[105] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3d action recognition with random occupancy patterns. In *European Conference on Computer Vision (ECCV)*, 2012.

[106] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[107] Jun Wang and Jean-Daniel Zucker. Solving multiple-instance problem: A lazy learning approach. In *International Conference on Machine Learning (ICML)*, pages 1119–1125, 2000.

[108] Yang Wang and Greg Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 33(7):1310–1323, 2011.

[109] Jonathan Warrell and Philip HS Torr. Multiple-instance learning with structured bag models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 369–384. Springer, 2011.

[110] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3460–3469, 2015.

[111] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2834–2841. IEEE, 2013.

[112] Zhongwen Xu, Ivor W Tsang, Yi Yang, Zhigang Ma, and Alexander G Hauptmann. Event detection using multi-level relevance labels and multiple features. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

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

[114] Weilong Yang, Yang Wang, Arash Vahdat, and Greg Mori. Kernel latent svm for visual recognition. In *Neural information processing systems (NIPS)*, 2012.

[115] Xiaodong Yang and YingLi Tian. Super normal vector for activity recognition using depth sequences. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[116] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *ACM MM*, 2012.

[117] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *International Conference on Machine Learning (ICML)*, 2009.

[118] Felix X Yu, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. $\propto$ svm for learning with label proportions. In *International Conference on Machine Learning (ICML)*, 2013.

[119] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *International Conference on Computer Vision (ICCV)*, 2013.

[120] Q. Zhang and S.A. Goldman. Em-dd: An improved multiple-instance learning technique. *Neural Information Processing Systems (NIPS)*, 14:1073–1080, 2001.

[121] Z.H. Zhou, Y.Y. Sun, and Y.F. Li. Multi-instance learning by treating instances as non-iid samples. In *International Conference on Machine Learning (ICML)*, pages 1249–1256, 2009.

[122] Zhi-Hua Zhou and Jun-Ming Xu. On the relation between multi-instance learning and semi-supervised learning. In *International conference on Machine learning (ICML)*, pages 1167–1174. ACM, 2007.

[123] Y. Zhu, N. Nayak, and A. Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

Appendix A

Joint Cardinality Kernel Learning and SVM Training

In this chapter, we show how to jointly learn the parameters of the Cardinality Model ($\boldsymbol{\theta}$) embedded in the Cardinallity kernel, integrated with a kernel SVM classifier. As a result, instead of separate pre-training of a kernel by likelihood maximization of the Cardinality Model, the Cardinality Kernel is directly trained for the target max-margin classification problem. This joint learning algorithm forms the instance-level labeling assignments and the bag-level discriminative classification together, with a direct goal of improved classification performance. In the experiments, we empirically show that this joint learning can improve classification results.

A.1 The Proposed Method: Cardinality Kernel Learning

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 $Y = \text{sign}\left(f(\mathbf{X})\right) \in \{-1, +1\}$. To this end, we follow the generalized multiple kernel learning framework [96] to optimize the SVM primal objective function w.r.t. the classifier parameters $\mathbf{w}$ and b, and the kernel parameters $\boldsymbol{\theta}$. First the SVM primal is rewritten as a nested optimization

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

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 $\nabla_{\boldsymbol{\theta}} T = \nabla_{\boldsymbol{\theta}} W$ [96], where W is the dual formulation of T:

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

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 [22, 14] 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 (A.2), is unique, $\nabla_{\boldsymbol{\theta}} W$ exists, and the derivatives are expressed by

$$\frac{\partial T}{\partial \theta_d} = \frac{\partial W}{\partial \theta_d} = \frac{r}{\partial \theta_d} - \frac{1}{2}\boldsymbol{\alpha}^{*t}\mathbf{Y}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_d}\mathbf{Y}\boldsymbol{\alpha}^*. \tag{A.3}$$

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) [62]. 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 (A.2) given $\boldsymbol{\theta}$ fixed, and next updating $\boldsymbol{\theta}$ by moving in the direction of derivities calculated in (A.3) 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 [86] 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:

$$\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_d} = \Big[\frac{\partial k_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q)\Big]_{N\times N}. \tag{A.4}$$

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

$$\begin{aligned}\frac{\partial k_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q) =& \frac{\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q)}{\sqrt{\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_p)}\sqrt{\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_q, \mathbf{X}_q)}} \\ &- \frac{k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q)\,\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_p)}{2\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_p)} - \frac{k_{\boldsymbol{\theta}}(\mathbf{X}_p, \mathbf{X}_q)\,\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_q, \mathbf{X}_q)}{2\tilde{k}_{\boldsymbol{\theta}}(\mathbf{X}_q, \mathbf{X}_q)},\end{aligned} \tag{A.5}$$

where

$$\begin{aligned}&\frac{\partial \tilde{k}_{\boldsymbol{\theta}}}{\partial \theta_d}(\mathbf{X}_p, \mathbf{X}_q) = \sum_{i=1}^{m_p}\sum_{j=1}^{m_q} k_x(\mathbf{x}_{pi}, \mathbf{x}_{qj}) \\ &\Big(\frac{\partial P_{\boldsymbol{\theta}}(y_{pi}|\mathbf{X}_p)}{\partial \theta_d}\Big|_{y_{pi}=1} \cdot P_{\boldsymbol{\theta}}(y_{qj}=1|\mathbf{X}_q) + P_{\boldsymbol{\theta}}(y_{pi}=1|\mathbf{X}_p)\cdot \frac{\partial P_{\boldsymbol{\theta}}(y_{qj}|\mathbf{X}_q)}{\partial \theta_d}\Big|_{y_{qj}=1} \\ &+ \frac{\partial P_{\boldsymbol{\theta}}(y_{pi}|\mathbf{X}_p)}{\partial \theta_d}\Big|_{y_{pi}=0} \cdot P_{\boldsymbol{\theta}}(y_{qj}=0|\mathbf{X}_q) + P_{\boldsymbol{\theta}}(y_{pi}=0|\mathbf{X}_p)\cdot \frac{\partial P_{\boldsymbol{\theta}}(y_{qj}|\mathbf{X}_q)}{\partial \theta_d}\Big|_{y_{qj}=0}\Big).\end{aligned} \tag{A.6}$$

So, the only thing needed is to find $\frac{\partial P_{\boldsymbol{\theta}}(y_i|\mathbf{X})}{\partial \theta_d}$ for all y_i in a bag $\mathbf{X}$. Actually, calculating these derivities is not straightforward, but taking derivities of $\log P_{\boldsymbol{\theta}}(y_i|\mathbf{X})$ is quite similar to taking the derivatives of the log likelihood function in HCRFs [83]. Thus, by exploiting the relations in [83] 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

$$\frac{\partial P_{\boldsymbol{\theta}}(y_i|\mathbf{X})}{\partial \theta_d} = P_{\boldsymbol{\theta}}(y_i|\mathbf{X})\Big(\sum_{i'}\sum_{y_{i'}} P_{\boldsymbol{\theta}}(y_{i'}|y_i, \mathbf{X})\, x_{i'd}\, y_{i'} - \sum_{i'}\sum_{y_{i'}} P_{\boldsymbol{\theta}}(y_{i'}|\mathbf{X})\, x_{i'd}\, y_{i'}\Big). \tag{A.7}$$

The calculation of $P_{\boldsymbol{\theta}}(y_i|\mathbf{X})$ was described in (5.13). $P_{\boldsymbol{\theta}}(y_{i'}|y_i,\mathbf{X})$ 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 also computed in $O\left(m\log^2 m\right)$ time.

Putting all above together we propose a new algorithm, namely Cardinality Kernel Learning. The pseudo-code of this algorithm is provided in Algorithm 3.

Algorithm 3 Cardinality Kernel Learning

Input: Training data $\{(\mathbf{X}_n, Y_n)\}_{n=1}^N$, Cardinality potential parameters μ and σ, Regularization parameters λ 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 (A.2) with $\mathbf{K}$.
  Find $\frac{\partial \mathbf{K}}{\partial \theta_d}$ using (A.4), (A.5), (A.6), (A.7).
  $\theta_d = \theta_d - \eta(\frac{r}{\partial\theta_d} - \frac{1}{2}\boldsymbol{\alpha}^{*t}\mathbf{Y}\frac{\partial \mathbf{K}}{\partial \theta_d}\mathbf{Y}\boldsymbol{\alpha}^*)$.
until converged or maximum number of iterations

A.1.1 Computational Complexity

Here, we show the computational complexity of the Cardinality Kernel Learning algorithm. According to what was explained in Section A.1, 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 (A.7) takes $O(m^2\log^2 m + m^2\, d)$. Consequently, the time complexity of finding the kernel derivatives in (A.6) and (A.5) 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 (A.2) takes $O(N^3)$, the whole computational complexity of the algorithm would be $O(N_{iter}\, N_{sv}^2\,\bar{m}^2\, d + N_{iter}\, N_{sv}\, d\,\bar{m}^2\log^2\bar{m} + N_{iter}\, N^3)$ for N_{iter} iterations.

It is seen the running time of the algorithms is dependent on different parameters, including number of bags, number of support vectors, number of instances per bag, number of iterations, etc. However, as a numerical example, we show in Table A.1 the running time in seconds for different methods on the Musk1 dataset (introduced in Section A.2.1), using the termination criterion of less than 0.01 change in objective function. To remove the effect of number of iterations in the reported running times, we also provide the average training time per function evaluation (i.e., evaluating objective function + gradient function). Note that Musk1 has 92 bags with about 5 instances per bag, and the results are the averaged time of 10 runs of cross-validation. The experiment was performed on a 64 bit Linux machine with 8 cores @ 3.40GHz and 16GB system memory.

A.2 Experiments

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

Table A.1: Running time for different methods on Musk1 dataset

Method	Cardinality Model	Cardinality Kernel	Cardinality Kernel Learning
Average training time for each training fold (s)	115.17	119.16	424.30
Average prediction time for each test fold (s)	1.53	1.64	1.66
Average training time per function evaluation (s)	1.72	1.72	9.84

A.2.1 MIL Benchmark Datasets

The standard MIL benchmark datasets are the *Elephant*, *Fox*, *Tiger* image categorization datasets [6], and the *Musk1* and *Musk2* drug activity prediction datasets [25]. Though dated, these are the standard benchmark on which MIL algorithms are evaluated. In the image data sets, each bag is an image, and the instances inside the bag represent 230-D feature vectors of different segmented blobs of the image. These data sets contain 100 positive and 100 negative bags. 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). In all the experiments, we have preprocessed datasets by scaling the features of the original datasets to the range $[0, 1]$. Musk1 dataset has been prepared for the purpose of drug activity prediction, in which the molecules are classified into “musk" or “non-musk" type. In fact, because of twisting or bending, each molecule can have different configurations (i.e. shapes). However, it is unknown or significantly hard to figure out which configuration results in the musk label. Thus, each molecule is represented as a bag of configurations, where in a positive bag at least one of the configurations is of musk type and in a negative bag all the configurations are of non-musk type. In the Musk data set, the instances inside each bag describe 166-D feature vectors of the low-energy configurations of a molecule. 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 the Cardinality Model (the same values were then used for the Cardinality Kernel and the Cardinality Kernel Learning methods). The regularization weights λ and C are also roughly optimized on 10-fold cross-validation accuracy. As the primitive instance kernels in the Cardinality Kernel we use RBF kernel, similar to the MI-Kernel setting in [39].

The results are reported based on 10-fold cross-validation classification accuracy and compared with the state-of-the-art MIL methods in Table A.2. It can be seen that the Cardinality Kernel Learning algorithm performs well compared to the other methods. More specifically, it achieves the best accuracy on the Elephant, Fox, and Tiger data sets. In addition, the Cardinality Kernel alone is by and large comparable to the best methods although it is more computationally efficient than the Cardinality Kernel Learning method.

Table A.2: 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
Cardinality Model	84	65	86	81	83	79.8
Cardinality Kernel	*88*	63	*87*	89	89	*83.2*
Cardinality Kernel Learning	**89**	**71**	**88**	89	89	**85.2**
MIMN [46]	**89**	64	*87*	86	**90**	*83.2*
MIRealBoost [47]	83	63	73	**91**	77	77.4
ClassSetMaxRBM$^{\text{XOR}}$ [71]	*88*	60	83	84	84	79.8
MI-CRF [23]	85	*68*	83	88	85	81.8
MIGraph [121]	85	61	82	*90*	**90**	81.6
miGraph [121]	87	62	86	*90*	**90**	83.0
ALP-SVM [40]	84	66	86	86	86	81.6
MILES [18]	81	62	80	88	83	78.8
MI-Kernel [39]	84	60	84	88	89	81.0
mi-SVM [6]	82	58	79	87	84	78.0
MI-SVM [6]	81	59	84	78	84	77.2

A.2.2 Collective Activity Dataset

In this section, we perform experiments on the collective activity dataset [21], which comprises 44 videos (equivalent to about 2500 video frames) of crossing, waiting, queuing, walking, and talking. Our goal is to detect the single collective activity in each frame, where the collective activity is the action that the majority of people in the scene are doing. To this end, we model the scene as a bag of people represented by the *action context* feature descriptors developed in [61], and use our proposed algorithms with the majority cardinality potential model (i.e., the ratio-constrained model with $\rho = 0.5$).

Table A.3: Comparing different methods based on classification accuracy (in percent) on the collective activity dataset.

Class	Cardinality Model	Cardinality Kernel	Cardinality Kernel Learning	[3]	AOG[4]	HiRF[2]	HiRFnt[2]
Cross	75.3	86.1	86.9	69.9	77.2	76.8	81.2
Wait	88.7	84.4	84.2	74.1	78.3	74.3	78.4
Queue	95.1	95.6	96.1	96.8	95.4	81.1	96.2
Walk	81.5	86.7	86.5	72.2	74.7	84.1	77.3
Talk	91.5	99.8	99.8	99.8	98.4	99.3	99.6
Avg	86.4	90.5	90.7	82.5	84.8	83.1	86.6

Similar to [61, 3, 4, 2], in our experiments 1/3 of the video clips were selected for test and the rest for training. The results are shown in Table A.3 and Table A.4 based on classification accuracy and precision of detection, respectively. We compare with the relevant frame-wise

methods in [3, 4, 2]. The comparison shows the efficacy of the proposed kernel methods, where the results compare favourably to the state-of-the-art.

Table A.4: Comparing different methods based on average precision (in percent) on the collective activity dataset.

Class	Cardinality Model	Cardinality Kernel	Cardinality Kernel Learning	[3]	S-AOG[4]	HiRFnt[2]
Cross	42.7	73.7	78.7	61.5	69.6	75.0
Wait	83.3	67.3	65.3	59.2	68.3	74.1
Queue	94.1	99.7	99.8	65.5	76.2	78.7
Walk	20.8	47.0	46.8	58.1	65.3	68.1
Talk	99.0	99.7	99.7	67.5	82.1	84.4
Avg	68.0	77.5	78.0	62.4	72.3	76.0

A.3 Summary and 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. The proposed adaptive kernel can help to perform classification within a task-tuned discriminative embedded space of even infinite dimensionality. The results of our experiments on standard MIL benchmark datasets and the retrieval of collective activities in video showed the efficacy of the proposed kernel learning approach, achieving state-of-the-art classification accuracy.