Learning with multiple representations : algorithms and applications

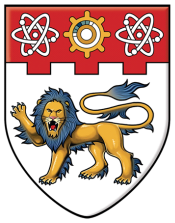
Xu, Xinxing

2014

Xu, X. (2014). Learning with multiple representations : algorithms and applications. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/62188

https://doi.org/10.32657/10356/62188



NANYANG
TECHNOLOGICAL
UNIVERSITY

Learning with Multiple Representations: Algorithms and Applications

A thesis submitted to
The School of Computer Engineering
The Nanyang Technological University

by

Xu Xinxing

in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

December 2014

Abstract

Recently, lots of visual representations have been developed for computer vision applications. As different types of visual representations may reflect different kinds of information about the original data, their differentiation ability may vary greatly. As the existing machine learning algorithms are mostly based on the single data representation, it becomes more and more important to develop machine learning algorithms for tackling data with multiple representations. Therefore, in this thesis we study the problem of learning with multiple representations. We develop several novel algorithms to tackle data with multiple representations under three different learning scenarios, and we apply the proposed algorithms to a few computer vision applications. Specifically, we first study the learning with multiple kernels under fully supervised setting. Based on a hard margin perspective for the dual form of the traditional ℓ_1-norm Multiple Kernel Learning (MKL), we introduce a new "kernel slack variable" and propose a Soft Margin framework for Multiple Kernel Learning (SMMKL). By incorporating the hinge loss for kernel slack variables, a new box constraint for the kernel coefficients is introduced for Multiple Kernel Learning. The square hinge loss and the square loss soft margin MKLs naturally incorporate the family of elastic-net MKL and ℓ_2MKL, respectively. We demonstrate the effectiveness of our proposed algorithms on benchmark data sets as well as several computer vision data sets. Second, we study the learning with multiple kernels for weakly labeled data. Based on "input-output kernels", we propose a unified Input-output Kernel Learning (IOKL) framework for handling weakly labeled data with multiple representations. Under this framework, the general data ambiguity problems such as SSL, MIL and clustering with multiple representations are solved in a unified framework. We formulate the learning problem as a group sparse MKL problem to incorporate the intrinsic group structure among the input-output kernels. A group sparse soft margin regularization is

further developed to improve the performance. The promising experimental results on the challenging NUS-WIDE dataset for a computer vision application (i.e., text-based image retrieval), SSL benchmark datasets and MIL benchmark datasets demonstrate the effectiveness of our proposed IOKL framework. Third, we study the learning with privileged information for distance metric learning, where the distance metric is learnt with extra privileged information which is available only in the training data but unavailable in the test data. We propose a novel method called Information-theoretic Metric Learning with Privileged Information (ITML+) to model the learning scenario. An efficient cyclical projection method based on analytical solutions for all the variables is also developed to solve the new objective function. The proposed algorithm is applied to face verification and person re-identification in RGB images by learning from the RGB-D data. The extensive experiments are conducted on the real-world EUROCOM, CurtinFaces and BIWI RGBD-ID datasets and the results demonstrate the effectiveness of our newly proposed ITML+ algorithm.

Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor Dr. Xu Dong. His valuable advice and guidance have drived me over the past years of my Ph.D. journey. I am deeply touched by his patience and down-to-earth research attitude, which have influenced and changed me. Under his guidance, not only have I learnt the cutting-edge research on computer vision, but also the way to become an independent researcher for writing qualified scientific papers.

I would also like to give my special thanks to my main collaborator Dr. Ivor Wai-Hung Tsang, who has provided me a lot of inspirations and advice. His wisdom and guidance always give me confidence and encourage me to explore interesting ideas in the field of machine learning. Moreover, I would like to take this opportunity to thank all the three reviewers and committee members of my oral defence including Prof. Jianfei Cai, Prof. Tat Jen, Cham and Prof. Jianmin Zheng for their insightful comments and constructive suggestions, which significantly improve the quality of this thesis.

I would like to thank all my friends and the staffs in the Centre for Multimedia and Network Technology (CeMNet) of the Nanyang Technological University for their kind support and help. Many thanks to those excellent colleagues and friends in my research journey in Singapore, Chen Lin, Cheng Xiangang, Duan Lixin, Fu Huazhu, Gao Shenghua, Hoang Anh, Huang Yi, Kan Meina, Li Jia, Li Wen, Liu Huiying, Liu Jianyi, Liu Yiming, Nie Feiping, Niu Li, Ren Zhixiang, Shou Wei, Sun Hucheng, Tan Mingkui, Wang Jian, Wei Shikui, Wu Xinxiao, Xiao Shijie, Xiang Liu, Xiong Zhiwei, Xu Yanwu, Xu Zheng, Yan Shengye, Zhai Yiteng, Zhang Wenjie, Zeng Zinan. Special thanks to my great roommates, Chen Lin, Feng Liang, and Mao Qi. Thanks to all of them, I can fight against and finally survive my Ph.D. grind and make it a memorable journey in my life.

Last but not the least, I sincerely thank my parents, my elder brother and my girlfriend as well as her family for their love, supports and encouragements.

To my parents.

Contents

List of Figures

List of Tables

List of Notations and Abbreviations

In this thesis, a lowercase character represents a scalar (*i.e.*, b) and a bold lowercase character denotes a vector (*i.e.*, $\mathbf{x}$). A matrix is denoted by a bold uppercase character (*i.e.*, A). We list other frequently used notations and abbreviations as follows.

List of Notations:

$\mathbf{x}_i$	Feature vector of the i-th sample
y_i	Label of the i-th sample
$\mathbf{y}$	Label vector of the given training data
l	The number of labeled training samples
u	The number of unlabeled training samples
n	Number of training samples
$'$	Transpose of a vector or matrix
$\mathbf{0}_n$	$n \times 1$ vector of all zeros
$\mathbf{1}_n$	$n \times 1$ vector of all ones
$\mathbf{y} \odot \boldsymbol{\alpha}$	Element-wise product of two vectors $\mathbf{y}$ and $\boldsymbol{\alpha}$
$\mathbf{A} \odot \mathbf{B}$	Element-wise product of two matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{K}$	Kernel matrix
$\mathbf{K}_m$	the m-th Kernel matrix
$\mathbf{I}_n$	$n \times n$ identity matrix
$\mathrm{diag}(\boldsymbol{\alpha})$	Diagonal matrix with its diagonal elements as $\boldsymbol{\alpha}$
$\mathcal{H}$	Reproducing Kernel Hilbert Space (RKHS)
$\mathcal{R}$	Set of real numbers
$\mathcal{R}^n$	n-dimensional real value space
$\phi(\cdot)$	Nonlinear feature mapping function
$k(\cdot,\cdot)$	Kernel function induced by $\phi(\cdot)$
$\|\cdot\|_{\mathcal{H}}$	Norm in the RKHS $\mathcal{H}$

List of Abbreviations:

RKHS	Reproducing Kernel Hilbert Space
MIL	Multi-instance Learning
SSL	Semi-Supervised Learning
LUPI	Learning using Privileged Information
SVM	Support Vector Machine
SVM+	Support Vector Machine using Privileged Information
LapSVM	Laplacian SVM
TSVM	Transductive SVM
MKL	Multiple Kernel Learning
ℓ_1MKL	ℓ_1-norm Multiple Kernel Learning
ℓ_pMKL	ℓ_p-norm Multiple Kernel Learning
SMMKL	Soft Margin Multiple Kernel Learning
SM1MKL	Hinge Loss Soft Margin Multiple Kernel Learning
SM2MKL	Square Hinge Loss Soft Margin Multiple Kernel Learning
IOKL	Input-output Kernel Learning
ITML	Information-theoretic Metric Learning
ITML+	Information-theoretic Metric Learning with Privileged Information
QP	Quadratic Programming
QCQP	Quadratically Constrained Quadratic Programming
SDP	Semi-definite Programming
SMO	Sequential Minimal Optimization
MIP	Mixed Integer Programming

Chapter 1

Introduction

With the advancements of the electric and information technologies in the last century, more and more digital equipments such as cameras, smart phones are more and more popularized to the public. People can capture pictures or videos conveniently, and upload them into social networking sites such as Facebook[1], Instagram[2], Twitter[3], photo sharing website Flickr[4] and video sharing website Youtube[5]. Digital data especially the image and video data have being exploding drastically especially in the recent ten years. Needless to say, the world has entranced into the *big data era*. It is extremely important to organize, retrieve, and manage those visual data available at hand. However, although we have witnessed that a large quantity of visual data is emerging rapidly, recognizing and automatically managing the visual data is still in the research stage, but is attracting more and more attention in recent years both from the computer vision and machine learning fields.

When compared with the practical text retrieval techniques used by search engine such as Google[6] and Baidu[7], image and video retrieval is in its infant stage. Lots of efforts have been spent to catch up with the possible real-world applications. To this end, many computer vision benchmark data sets have been collected and released re-

[1] https://www.facebook.com/
[2] http://instagram.com/
[3] https://twitter.com/
[4] https://www.flickr.com/
[5] http://www.youtube.com/
[6] https://www.google.com.sg/?gws_rd=ssl
[7] http://www.baidu.com/

cently, including Caltech-101[8], Caltech-256[9] , Pascal VOC[10] and ImageNet[11] for image classification, Scene15[12] and Sun Database[13] for scene classification, NUS-WIDE[14] for image retrieval, UCF Sports[15] and KTH[16] for action recognition, TREC Video Retrieval Evaluation (TRECVID)[17] for video concept detection and video retrieval, Yale Face[18] and CMU Multi-PIE[19] for face recognition, Labeled Face in the Wild (LFW)[20] for face verification, VIPeR[21] for person re-identification and so on.

Despite of various data sets released for various computer vision applications, the prevailing approaches in the computer vision for those tasks all rely on the machine learning techniques, and can be briefly summarized into five steps: data collection, feature extraction, data labeling, model learning and prediction. Specifically, for a given task, the data capturing devices are utilized to capture the original data such as RGB images, depth images and videos. After collecting a data set, feature extraction methods are utilized to extract feature representations for the given data. Then, the labels of the data can be obtained by either human labeling or using any learning techniques. The classification model is further trained based on the extracted features as well as the labels. Finally, the new test data is predicted by using the learnt model for the corresponding tasks.

The image capturing devices include surveillance cameras, single-lens reflex camera (DLSR), depth cameras such as Microsoft Kinect sensors, smart phones, video recorders, and Google glasses. As a result, for the same scenario, different devices can capture different types of images. For instance, the surveillance cameras may have low resolution, while the DLSR may obtain another image with much higher resolution under the same

[8] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[9] http://www.vision.caltech.edu/Image_Datasets/Caltech256/
[10] http://pascallin.ecs.soton.ac.uk/challenges/VOC/
[11] http://www.image-net.org/
[12] http://www-cvr.ai.uiuc.edu/ponce_grp/data/
[13] http://vision.princeton.edu/projects/2010/SUN/
[14] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
[15] http://crcv.ucf.edu/data/UCF_Sports_Action.php
[16] http://www.nada.kth.se/cvap/actions/
[17] http://trecvid.nist.gov/
[18] http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/YaleFaceDatabase.htm
[19] http://www.multipie.org/?3e3ea140
[20] http://vis-www.cs.umass.edu/lfw/
[21] VIPeR:ViewpointInvariantPedestrianRecognition

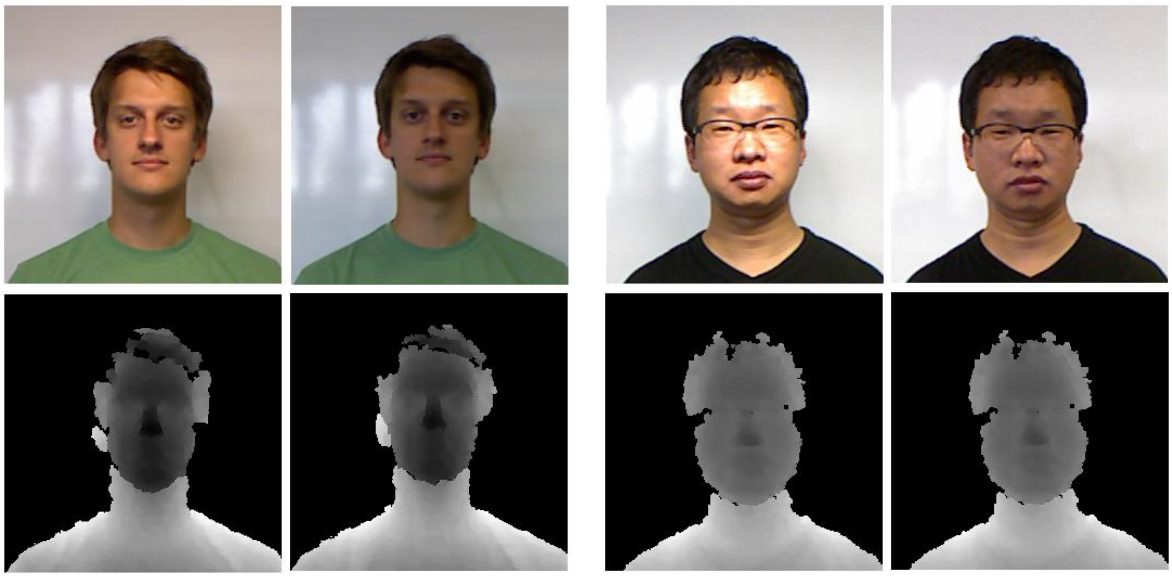


Figure 1.1: The RGB images and depth images of two similar pairs in the EUROCOM dataset. The first row shows the RGB images captured under different lighting conditions, and the second row shows the corresponding depth images.

condition. Another example is the Kinect sensor, which can simultaneously capture the RGB and depth images (see Fig. 1.1). Obviously, the depth image contains information that has been missed in the traditional RGB image.

The term "*representation*" is defined as "*the description or portrayal of someone or something in a particular way*" by the Oxford Dictionaries[22]. In computer vision, a good data representation [15] is crucial for the success of machine learning algorithms. Exploiting effective data representation for images and videos has been one of most fundamental problems for computer vision, and has attracted lots of attention accompanying the birth of the computer vision. For one image or video, the visual representations in the form of visual features can be extracted. These feature representations include simple raw pixels, color moment, handcrafted local features such as SIFT [139], HOG [50], LBP [2], high-level features such as attributes [159], and the deep representations by using deep learning methods [15],[106]. In this way, we are easily to obtain multiple types of feature representations for visual data such as images and videos.

Machine learning plays an important role in computer vision. After obtaining the feature representation, the tasks from computer vision are transferred into different ma-

[22]`http://www.oxforddictionaries.com/definition/english/representation`

chine learning problems. The ground truth labels of the data is another important factor for the training of the models in both the computer vision and machine learning. Most of the existing data sets are labeled by human manually. Recently, the crowdsourcing techniques such as Amazon Mechanical Turk are utilized to obtain the labels more conveniently. Besides, the web resources are also utilized to construct weak labels for the data set such as NUS-WIDE [39].

For data with complete label information, the supervised learning scenarios can be applied. The boosting such as the Adaboost [195] has been successfully applied to the face detection, and sparse representation has been utilized for face recognition [208]. The Nearest Neighbor (NN) and distance metric learning algorithms have been applied to tasks such as face verification [80] and person re-identification [78]. One of the most popular algorithms for computer vision tasks has been the Support Vector Machine (SVM), which is shown to achieve promising results for a broad range of applications such as object recognition [106], object detection [67], pedestrian detection [50], scene classification [211], action recognition [210] and video understanding [93].

Obtaining the labels is always a big issue for both the academic work and engineering applications, but the data with incomplete labels which are referred to as weakly labeled data are easily to be obtained. Learning with weakly labeled data has been a hot topic in machine learning. Specifically, Semi-supervised Learning (SSL) trains the model by using a small number of labeled data and a large number of unlabeled data, while Multi-instance Learning (MIL) is proposed to tackle data with label constraints given in bag manner, and Clustering tackles the case where no labeled data are available.

Although machine learning techniques have been widely applied to computer vision tasks and some multi-view learning methods have been developed recently (see [215] for a comprehensive review), most of the current prevailing machine learning algorithms such as SVM, TSVM, mi-SVM and LMNN are based on single data representation. As new data capturing devices become available as well as the developments of new feature extraction methods, the learning algorithms that can tackle those multiple representations are needed in real-world applications due to the fact that different representations may contain different information. To this end, in this thesis we study the problem of learning with multiple representations. According to different problem settings in real-world applications, we study the following three problems:

- How to train classifier for supervised learning with multiple representations?
- How to train classifiers for weakly labeled data with multiple representations?
- How to train classifiers for data with additional privileged representation that is only available to the training data but unavailable to the test data?

We aim to develop several novel learning algorithms for learning with multiple representations for these three different learning scenarios, and also apply the proposed algorithms to a few real-world computer vision applications. In the following, we summarize the contributions of this thesis and also introduce the structure of this thesis.

1.1 Thesis Contributions

This thesis studies the learning with multiple representations with novel algorithms and real-world applications to both the computer vision and machine learning fields. We briefly summarize the proposed contributions as follows:

- We propose a novel *Soft Margin Multiple Kernel Learning (SMMKL)* framework to learn robust supervised classifiers for data with multiple representations. Specifically, we show that the traditional ℓ_1MKL can be deemed as hard margin MKL. Then, based on the so-called kernel slack variable, we propose a novel soft margin framework for MKL. Our Soft Margin MKL framework incorporates the ℓ_2MKL and family of elastic net constraint/regularizer based MKL formulations naturally as square loss and square hinge loss Soft Margin MKL, respectively. Moreover, by using the hinge loss under our soft margin MKL framework, we develop a new box constraint for the kernel combination coefficients for MKL problem, which inherently bridges the method using average kernel and ℓ_1MKL. The soft margin MKL is found to be good at handling the noisy base kernels.
- We present a unified kernel learning framework named *Input-Output Kernel Learning (IOKL)* to handle weakly labeled data with multiple representations. Specifically, based on the so-call input-output kernel, we formulate the learning with

general data ambiguity to be an Input-Output Kernel Learning problem. The classifier is trained to cope with weakly labeled data with multiple representations using multiple kernel learning techniques. Based on our framework, we further propose a novel soft margin group sparse Multiple Kernel Learning formulation by introducing a group kernel slack variable to each group of base input-output kernels. Moreover, an efficient block-wise coordinate descent algorithm with an analytical solution for the kernel combination coefficients is obtained to solve the proposed formulation. The proposed framework is applied to semi-supervised learning, multi-instance learning setting with multiple data representations.

- We propose a novel approach to learn distance metric with privileged information. Specifically, we propose a new formulation called *Information-theoretic Metric Learning with Privileged Information (ITML+)* to learn a more robust distance metric with additional privileged information available in the training set. We also present an efficient algorithm based on the cyclical projection method for solving the proposed ITML+ formulation. The proposed algorithm is applied to improve face verification and person re-identification in RGB images by leveraging a set of RGB-D data captured by using depth cameras (*i.e.*,Kinect). Visual features and depth features are extracted from the RGB images and depth images, respectively. As the depth features are only available in the training data, we treat the depth features as privileged information, and we demonstrate both the effectiveness of the additional depth feature in the training set as well as our ITML+ algorithm to utilize the additional privileged information in the training set.

1.2 Thesis Structure

This thesis contains six chapters. The structure of this thesis is shown in Fig. 1.2. The introduction about background and motivations is in Chapter 1 (this chapter). In Chapter 2, we review traditional machine learning algorithms for learning with single representation, learning with multiple kernels, learning with privileged information as well as data representation in Computer Vision applications. In Chapter 3, we study the problem of learning with multiple representations under the supervised setting, and we

present a novel Soft Margin Multiple Kernel Learning (SMMKL) framework. In Chapter 4, we study the problem of learning with multiple representations under the weakly supervised setting. We propose a novel unified Input-Output Kernel Learning (IOKL) framework to handle general weakly labeled data with multiple representations. The proposed framework is applied to text-based image retrieval task. In Chapter 5, we study the problem of learning using privileged information, where additional information is only available to the training data, but not available to the test data. We propose a novel Information-theoretic Metric Learning with Privileged Information (ITML+) algorithm. The algorithm is applied to the RGB face verification and person re-identification tasks by learning distance metrics from RGB-D data. In Chapter 6, we conclude our work and also discuss future extensions for this thesis.

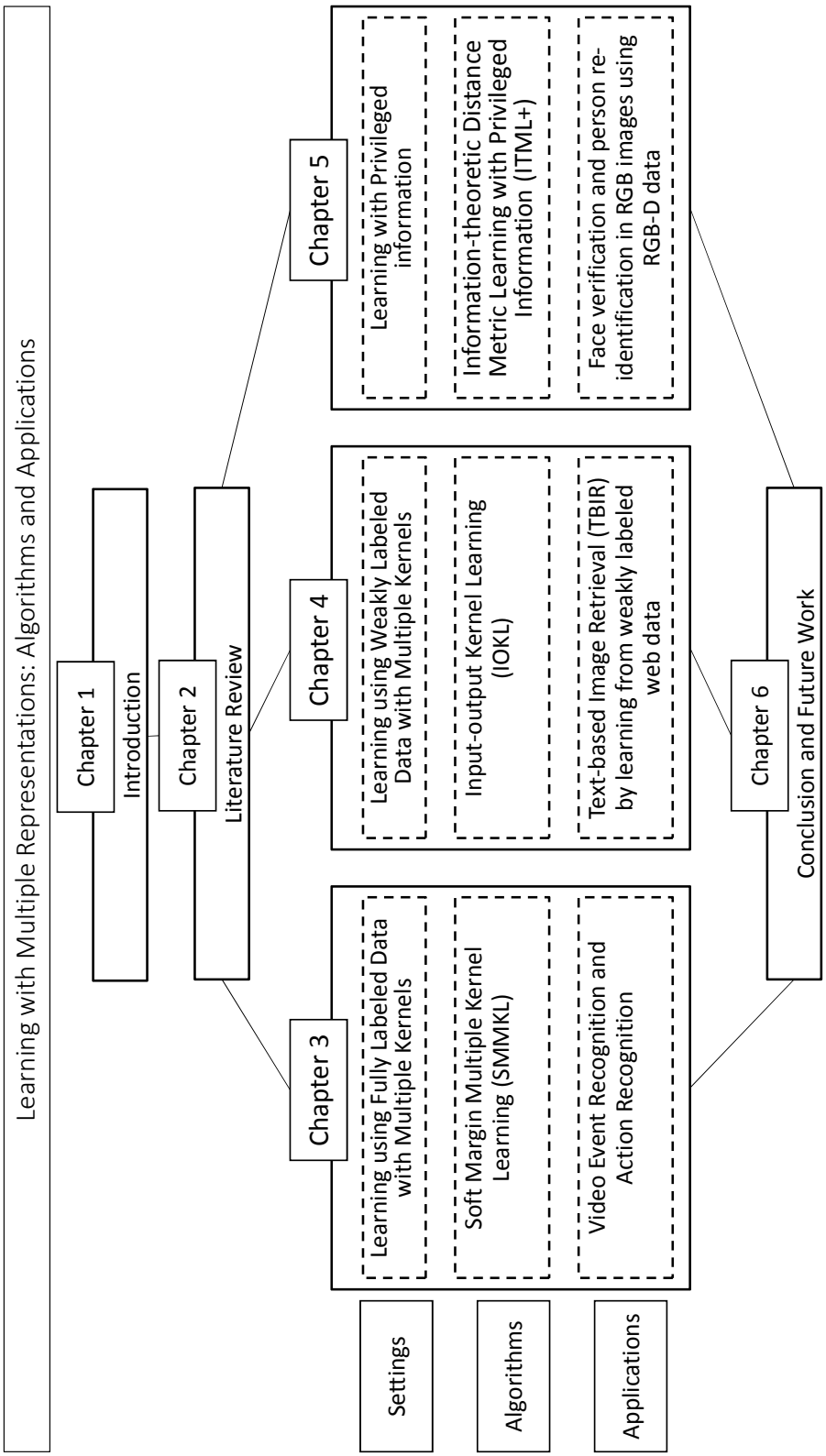


Figure 1.2: The structure of this thesis.

Chapter 2

Literature Review

Machine Learning plays an important role in artificial intelligence areas such as data mining as well as computer vision. It concerns how to construct a learning system that can learn from data. The core problems of the machine learning have been representation and generalization. Based on the different settings of the data representation, in this thesis, we summarize the different existing learning algorithms mainly into the following three categories:

- *Learning with Single Representation*, which refers to the learning setting where both the training data and test data are given with single representation;
- *Learning with Heterogeneous Information*, which refers to the learning setting where both the training data and test data are given with multiple representations;
- *Learning with Privileged Information*, which refers to the learning setting where the training data are associated with with multiple representations, but the test data are associated with single representation.

Throughout the rest of this thesis, we use the superscript $'$ to denote the transpose of a vector, and $\mathbf{0}, \mathbf{1} \in \mathcal{R}^l$ denote the zero vector and the vector of all ones, respectively. We also define $\boldsymbol{\alpha} \odot \mathbf{y}$ as the element-wise product between two vectors $\boldsymbol{\alpha}$ and $\mathbf{y}$. Moreover, $||\boldsymbol{\mu}||_p$ represents the ℓ_p-norm of a vector $\boldsymbol{\mu}$ and we specially denote the ℓ_2-norm of $\mathbf{d}$ as $\|\mathbf{d}\|$, and the inequality $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_l]' \geq \mathbf{0}$ (*resp.*, $\mathbf{D} \geq \mathbf{0}$, $\mathbf{D} \in \mathcal{R}^{M \times T}$) means that $\mu_i \geq 0$ for $i = 1, \ldots l$ (*resp.*, $d_{m,t} \geq 0$ for $m = 1, \ldots, M, t = 1, \ldots, T$).

2.1 Distance and Kernel

The data is usually represented by a vector $\mathbf{x} \in \mathcal{R}^h \subset \mathcal{X}$, where h is the feature dimension of the data, and $\mathcal{X}$ is the distribution. Given two data points $\mathbf{x}$ and $\mathbf{x}'$ from the distribution $\mathcal{X}$, the relationship between them is an important issue for many machine learning algorithms. We introduce the two most popular concepts between the given two points in the following. The two concepts are *kernel* and *distance*.

2.1.1 Distance

Another commonly used concept is the distance, and the distance can be directly used for nearest neighbor search, k-means clustering, as well as the construction of the RBF kernel. The Euclidean distance directly measures the relationship between two points by using Pythagorean formula, specifically,

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{h} (x_i - x_i')^2} = ||\mathbf{x} - \mathbf{x}'||. \tag{2.1}$$

The Euclidean distance may be sensitive to some measurements that have a large range, and the Mahalanobis distance is introduced by considering a matrix $\mathbf{M} \in \mathcal{R}^{h \times h}$, and it is defined as

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')\mathbf{M}(\mathbf{x} - \mathbf{x}')}, \tag{2.2}$$

where $\mathcal{R}$ is positive semi-definite to ensure that the distance metric $d_{\mathbf{M}}$ satisfies the non-negativity and the triangle inequality.

2.1.2 Kernel

The definition of the kernel is given by (Aizerman et al., 1964 [4]) as the following:

Definition 2.0 *A kernel is a function k that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ satisfies*

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle, \tag{2.3}$$

where ϕ is a mapping from $\mathcal{X}$ to an (inner product) Hilbert space $\mathcal{H}$

$$\phi : \mathcal{X} \mapsto \mathcal{H}. \tag{2.4}$$

The nonlinear mapping $\phi(\cdot)$ maps the data from the original space $\mathcal{R}^h$ to the high dimensional reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. If we are given a set of n data points $\{\mathbf{x}_i\}|_{i=1}^n$, we can obtain a kernel matrix $\mathbf{K} \in \mathcal{R}^{n \times n}$, where $\mathbf{K}(i,j) = k(\mathbf{x}_i, \mathbf{x}_j)$. Usually, the linear classifier $f(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b$ can be constructed in the original low dimensional space $\mathcal{R}^h$, and by using the kernel trick, the linear classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ in RKHS $\mathcal{H}$ is essentially the nonlinear classifier in the original low dimensional space. Thus, the kernel is very important for constructing the nonlinear classifier, and the kernel trick has been successfully applied to algorithms such as SVM [19] and kernel PCA [169].

2.2 Learning with Single Representation

In this Section, we review related machine learning algorithms for learning with single representation.

2.2.1 Supervised Learning

Supervised learning is the task of learning classifiers by using given training samples as well as their full labels to learn a classifier that can be applied to unseen new data. Specifically, given a set of labeled training data $S = \{(\mathbf{x}_i, \mathrm{y}_i)|i = 1, \ldots, l\}$ sampled independently from $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} \subset \mathcal{R}^h$ and $\mathcal{Y} = \{-1, +1\}$, the task is to learn a classifier that could classify the training data and can also be applied to generalize to the new unseen test data $\{\mathbf{x}_i\}|_{i=l+1}^{l+m}$. The most widely used supervised learning algorithms are the Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) classifier.

Support Vector Machine The Support Vector Machine has been a powerful tool for many real world applications, and it employs the maximum margin criteria to the classification tasks. The hard margin SVM was first proposed in the year of 1992 [19], and the soft margin SVM was further developed three years later in [45]. We firstly introduce the hard margin SVM. If the data are mapped into a Hilbert space $\mathcal{H}$, a linear classifier on the space to separate the data can be constructed as

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b, \tag{2.5}$$

where $\phi(\mathbf{x})$ is the mapping function.

Suppose that the hyperplane can separate the positive samples from the negative samples, then we have that the points satisfying $\mathbf{w}'\phi(\mathbf{x}) + b = 0$ lie on the hyperplane, then the *margin* of the hyperplane is defined as the sum of the shortest distances from the separating hyperplane to the negative sample as well as the positive sample. It can be easily shown that $2/||\mathbf{w}||_2$ is the margin for the separating hyperplane.

The *hard margin* Support Vector Machine (SVM) proposed in [19] constructs the classifier by minimizing the following objective function:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||_2^2 \\ \text{s. t.} \quad & \mathrm{y}_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1, i = 1, \ldots, l, \end{aligned} \tag{2.6}$$

and the dual can be formulated as

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}(\boldsymbol{\alpha} \odot \mathbf{y}) \\ \text{s. t.} \quad & \boldsymbol{\alpha} \geq 0, \boldsymbol{\alpha}'\mathbf{y} = 0, \end{aligned} \tag{2.7}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_l]'$ and α_i is the non-negative Lagrangian multiplier for each of the inequality constraint as in the primal problem.

In the year of 1995, the *soft margin* SVMs were first proposed [45] by introducing the slack variables ξ_is for each of the training point. By considering the hinge loss function for the slack variables, the primal SVM objective function is given as in the following:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{l} \xi_i \\ \text{s. t.} \quad & \mathrm{y}_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \end{aligned} \tag{2.8}$$

and compared to the hard margin SVM formulation as in (2.6), the soft margin SVMs in (2.8) performs well for real-world applications by tuning the additional regularization parameter C. This is due to the fact that the introduced slack variables can cope with data with different levels of noise. The square hinge loss and the square loss can also be investigated similarly.

The dual of (2.8) can be derived [45] similarly by using the Lagrangian method with the derivation with hard margin SVM as the following form:

$$\max_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}'\mathbf{1} - \frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}(\boldsymbol{\alpha} \odot \mathbf{y}), 0 \leq \boldsymbol{\alpha} \leq C, \boldsymbol{\alpha}'\mathbf{y} = 0, \tag{2.9}$$

which is a Quadratic Programming (QP) problem. Efficient algorithms have been developed for solving (2.8) effectively such as the Sequential Minimal Optimization (SMO) algorithm [162]. Training the SVM in primal has also been studied in [29] [97]. Many softwares have been developed to solve the SVM problem effectively, and the widely used ones are LibSVM[1] [28], LibLinear[2] [63] for linear SVM, and SVM-Light[3] [95] [94].

Since its birth, SVM has been successfully applied to a broad range of applications, such as data mining [209], bioinformatics [86], information retrieval [9] and computer vision *etc.*. For example, the SVM has been applied to text categorization [94]. For computer vision, SVM has achieved promising state-of-the-arts results for applications such as object recognition [106], image retrieval [37], pedestrian detection [50], scene classification [211], action recognition [210] and video understanding [93], *etc.*.

Beyond the original formulation for binary classification problem as proposed in [45], lots of extensions based on SVM have been done recently. The representative works include multi-class SVM [46], [84] for multi-class classification problem, structural SVM [191], [98] for structural prediction, Support Vector Regression (SVR) for regression problem [180], latent SVM [67], latent structural SVM [233], Transductive SVM (TSVM) [96] as well as Laplacian SVM [36] for semi-supervised learning, and Multi-instance SVM [3] for multi-instance learning *etc.*.

Distance Metric Learning Compared with SVM, the k-Nearest Neighbor classifier (kNN) is in a simpler form for the classification, which usually employs the Euclidean distance based on the k-nearest neighbors of the samples for classification or regression tasks. As the accuracy of the kNN classification depends highly on the metric used to calculate the distance between different samples, the distance metric learning has extracted lots of attention in recent works [214] [204], [80], [52], [232], [107], [13]. The early work for the Mahalanobis distance metric learning in [214] formulates the distance metric learning problem as a convex optimization problem that maximizes the sum of distances between dissimilar pairs while minimizing the sum of distances between similar pairs. A projected gradient descent method was proposed to solve the proposed objective function,

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[2] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`
[3] `http://svmlight.joachims.org/`

but the SVD operation on the distance metric $\mathbf{M}$ makes the algorithm only applicable to small scale problems. Following [214], a large number of methods were proposed in literature (see the surveys [13, 107, 232] for comprehensive reviews of different metric learning methods). The two representative methods for distance metric learning are the Large Margin Nearest Neighbors (LMNN) method [204] and the Information-theoretic Metric Learning (ITML) [52] method.

2.2.2 Weakly Labeled Learning

Data labeling has always been a time consuming task, and many learning settings with limited number of labeled data or even with no labeled data have been proposed to tackle the data with partial labels, data with implicitly known labels, and data with completely unknown labels, respectively. These learning scenarios correspond to semi-supervised learning, multi-instance learning, and unsupervised learning (*i.e.*, clustering). Due to the different information available for the label, algorithms differ significantly for the different learning scenarios. In this thesis, we uniformly refer to the learning scenarios where the labels of the training data are incomplete as *Weakly Labeled Learning* following [132]. We briefly introduce and review the related works from literature as follows:

Semi-Supervised Learning For Semi-Supervised Learning (SSL), a large set of u unlabeled training samples $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$ and a small set of labeled training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ are assumed to be given. The task of SSL is to learn a more robust classifier that can take advantage of the additional unlabeled data when compared with the supervised classifier trained by using the labeled data only. Based on the assumptions on the given unlabeled training data, SSL methods can be roughly classified into cluster assumption [32] based methods and manifold assumption [36] based methods.

The cluster assumption assumes that the data from the same cluster are more likely to share the same class label. Specifically, if an edge connecting given two data points lies in a high density region of the data distribution, they are more likely to share the same class label. Therefore, the decision boundary of the learnt classifier should lie in low density regions such that the classification error could be minimized. Many SSL methods such as Transductive SVM (TSVM) [96], Low Density Separation (LDS) [32]

and meanS3svm [131] are based on the cluster assumption. The most representative work for the cluster assumption based methods is the TSVM, which is based on inferring the labels of unlabeled data and learning the classifier simultaneously. As the inferring of the unlabeled data is a NP-hard problem, the proposed algorithm in [96] cannot be readily applied to large scale applications. The concave-convex procedure (CCCP) is utilized to solve the large scale TSVM in [40] and makes TSVM applicable to large scale real-world applications.

Another sets of works such as Laplacian SVM (LapSVM) [12], Laplacian Regularized Least Square (LapRLS) [12], Laplacian Embedded Regression (LapREMR) [36] are based on the manifold assumption, which assumes that each class of the data lies on a separate low-dimensional manifold which is embedded in a high dimensional feature space. The manifold regularization framework [12] developed algorithms such as Laplacian Support Vector Machine, Laplacian Regularized Least Squares, and Laplacian Support Vector Regression (LapSVR) and have shown the state-of-the-art performance for a broad range of applications. The solution for Laplacian SVM in [12] involves an inversion of the matrix to recover the dual variables, and becomes inefficient for large scale applications. In [147], the training of Laplacian SVM in primal form is studied while in [36] a Laplacian embedded regression method is proposed to solve the large scale manifold regularization problem efficiently.

Multi-Instance Learning Multi-Instance Learning [188] [3], [23], [128], [73] is under the setting where labels of training data are implicitly known. Specifically, the training data are provided in the form of bags. Only the label of each bag is known, while the labels of instances in each bag remain unknown. In MIL, the constraints are that all instances in the negative bags are negative and at least one instance (or a portion of the instances) in each positive bag is positive [3], [128]. Let us denote $\mathcal{B}_I$ as the I-th training bag and Y_I as the corresponding given bag label. Then we can define the label candidate set as $\mathcal{Y} = \{\mathbf{y} | \sum_{i:\mathbf{x}_i \in \mathcal{B}_I} (\mathrm{y}_i + 1)/2 \geq \varepsilon, \text{ if } Y_I = 1; \mathrm{y}_i = -1, \text{if } Y_I = -1\}$, where we have $\varepsilon = 1$ for the traditional MIL constraint [3] and $\varepsilon = \mu|\mathcal{B}_I|$ for the general MIL constraint [128] with μ being the portion parameter and $|\cdot|$ being the cardinality function.

The different algorithms for Multi-Instance Learning include Non-SVM-based methods (*i.e.,* DD [144], EM-DD [238]) graph-based methods (*i.e.,* MIGraph [244], miGraph

[244], HSR-MIL [120]), similarity-based method (*i.e.,* SMILE [212]) and SVM-based methods (*i.e.,* MI-SVM [3], mi-SVM [3], MI-Kernel [73], sMIL [23], KISVM [130], MIL-CPB [128], $\propto$SVM [113]). The SVM-based algorithms focus on inferring the labels of ambiguous samples in a large margin regularization manner. The MIL has also been utilized for real-world applications such as drug activity prediction [144], content-based image retrieval (CBIR) [239], text-based image retrieval [128], image classification [224] [130], tracking [5], [125] and video event detection [113] *etc.*.

Clustering Another setting with incomplete labeled data is the unsupervised learning. In this case, all the given training data are with no label information, and the task is to cluster the data into a few clusters. Similar to the case for SSL, the assumptions for the clustering methods can also be roughly classified into manifold assumption based methods as well as cluster assumption based methods. For manifold based approaches, the manifold structure of the training data is explored. The most commonly used algorithms for clustering are the k-means [209] and spectral clustering [92]. On the other hand, the cluster assumption based methods for clustering are mainly based on the maximum margin criteria, such as the maximum margin clustering (MMC) [217], [133], [237], which aim to infer the labels for the unlabeled data in a maximum margin manner. The k-means algorithm has been applied to lots of computer vision tasks such as code book generation for image classification [118], while normalized cuts for spectral clustering has been successfully applied to image segmentation task [92].

2.3 Learning with Multiple Kernels

In this Section, we review the existing works for learning with heterogeneous information.

2.3.1 Theory, Algorithm and Applications

We now review the most popular Multiple Kernel Learning (MKL) algorithm from [115]. The ℓ_1MKL tries to learn the optimum linear combination coefficients and the decision function simultaneously from a set of given M base kernels $\{\mathbf{K}_1, ..., \mathbf{K}_M\}$. Lanckriet *et al.* [114], [115] proposed to learnt the kernel $\mathbf{K}$ as the linear combination from the given M base kernels as $\mathbf{K} = \sum_{m=1}^{M} \mu_m \mathbf{K}_m$.

Margin Based Kernel Learning methods The pioneer work in [114] proposed to use the maximum margin criteria similarly to the SVM, and if the ν-SVC [170] is used as the classification model, the MKL problem is firstly given in [114], [115] as the following form:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y}) \\ \text{s. t.} \quad & \boldsymbol{\alpha}'\mathbf{y}=0, \boldsymbol{\alpha}'\mathbf{1}=1, 0\le\boldsymbol{\alpha}\le C, \\ & \mathbf{K}\in\mathcal{K}, \end{aligned} \tag{2.10}$$

where $\mathcal{K}$ is the feasible set constructed from the given M base kernels.

In the following, we will list some representative works for improving the efficiency of the MKL problems.

- *QCQP*

  If only restricting the combined kernel matrix to be semi-positive definite $\mathbf{K}=\sum_{m=1}^{M}\mu_m\mathbf{K}_m\succeq 0$, the previous problem is formulated as the Semi-definite Programming (SDP) problem [114, 115]. If the combined kernel matrix is restricted to be the convex combination where $\sum_{m=1}^{M}\mu_m=1, \mu_m\ge 0$, the problem can be formulated [114, 115] as the quadratically constrained quadratic programming (QCQP) problem.

$$\begin{aligned} \max_{\boldsymbol{\alpha},\lambda} \quad & -\lambda \\ \text{s. t.} \quad & \boldsymbol{\alpha}'\mathbf{y}=0, \boldsymbol{\alpha}'\mathbf{1}=1, \\ & 0\le\boldsymbol{\alpha}\le C, \\ & \frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})\le\lambda, m=1,\ldots,M. \end{aligned} \tag{2.11}$$

- *Mixed Norm Primal Form* [7]

  The primal form for Multiple Kernel Learning was proposed in [7] as follows:

$$\begin{aligned} \min_{\{f_m\},b,\rho} \quad & \frac{1}{2}\Big(\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}\Big)^2 + C\sum_{i=1}^{l}\xi_i-\rho \\ \text{s. t.} \quad & \mathrm{y}_i(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b)\ge\rho-\xi_i, \xi_i\ge 0. \end{aligned} \tag{2.12}$$

As has been pointed out in [164] that the objective function is non-smooth due to the first term as in the objective of (2.12). The mixed-norm is type of group sparse ℓ_{21}-norm, which is non-smooth due to the fact that ℓ_1-norm is non-smooth.

- *SILP* [182]

  Based on the formulation in (2.10), the Semi-Infinite Linear Program formulation was proposed in [182] by solving the following problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha},\lambda} \quad & \lambda \\ \text{s. t.} \quad & \sum_{m=1}^{M} \mu_m = 1, \mu_m \geq 0, \\ & \frac{1}{2}\sum_{m=1}^{M} \mu_m (\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) \geq \lambda \\ & \boldsymbol{\alpha}'\mathbf{y} = 0, \boldsymbol{\alpha}'\mathbf{1} = 1, 0 \leq \boldsymbol{\alpha} \leq C, \end{aligned} \tag{2.13}$$

  This formulation makes the large-scale MKL problem applicable by recycling the single kernel SVM, thus the complexity of the problem is transformed to the optimization of the SVM.

- *SimpleMKL* [164]

  The above problem is non-smooth due to the mixed-norm structure for the complexity. Actually, as in [164], $\left(\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}\right)^2$ is equal to $\min_{\boldsymbol{\mu}} \sum_{m=1}^{M} \frac{\|f_m\|^2_{\mathcal{H}_m}}{\mu_m}$ with the simplex constraint for $\boldsymbol{\mu} \in \{\boldsymbol{\mu} | 0 \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m = 1\}$. Then the above formulation was further formulated in [164] as:

$$\begin{aligned} \min_{\boldsymbol{\mu},\{f_m\},b,\rho} \quad & \frac{1}{2}\sum_{m=1}^{M} \frac{\|f_m\|^2_{\mathcal{H}_m}}{\mu_m} + C\sum_{i=1}^{l} \xi_i - \rho \\ \text{s. t.} \quad & \mathrm{y}_i(\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b) \geq \rho - \xi_i, \xi_i \geq 0, \\ & 0 \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m = 1. \end{aligned} \tag{2.14}$$

  The above problem is formulated as the convex smooth problem by introducing the kernel coefficients explicitly in the objective function.

- *Block-wise coordinate descent* [222][99]

  The recent proposed work [222][99] apply the analytical updating rule for the kernel coefficients by directly solving the primal problem (2.14) due to the jointly convex property of the problem. The f_m, ρ, ξ are updated based on the dual of the standard SVM formulation, then the kernel coefficients $\boldsymbol{\mu}$ are updated by using the closed form formulation $\mu_m = \frac{\|f_m\|_{\mathcal{H}_m}}{\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}}$. The convergence of this updating rule has't be provided, but the stable and faster convergence property has been observed in [222][99].

We observe that these new formulations were proposed for more efficiency of the original problem for learning the kernel combination coefficients, and the efficiency of for solving the margin based optimization problem has been improved greatly. Due to the simplex constraint for the kernel coefficients for the above mentioned works, these formulations are referred to as the ℓ_1MKL. However, these objective functions are all equivalent to the original proposed one as in (2.10) for a long time in the early development of the Multiple Kernel Learning.

Extensions of the Margin Based methods The ℓ_1MKL can get very sparse solution for the kernel coefficients, and the performance of the ℓ_1MKL is not always good in real applications, thus many works recently focus on the improvement of the ℓ_1MKL formulation. In the following, we review some recent work about this direction.

- *ℓ_pMKL* [99]

  The non-sparse ℓ_2MKL [102][42][99][4] was proposed by substituting the simplex constraint with the ℓ_2-norm ball constraint $\boldsymbol{\mu} \in \{\boldsymbol{\mu} | 0 \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m^2 \leq 1\}$. The ℓ_pMKL [99] [158] directly extends the MKL formulation (2.14) by simply substituting the simplex constraint with the ℓ_p-norm constraint

[4]The original formulation is using the ridge regression problem which is equivalent to square loss soft margin SVM, but the main idea is to use the ℓ_2-norm for the kernel coefficients.

$\boldsymbol{\mu} \in \{\boldsymbol{\mu} | 0 \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m^p \leq 1\}$ for $p \geq 1$. The ℓ_pMKL primal were proposed as [99]

$$\begin{aligned} \min_{\boldsymbol{\mu}, \{f_m\}, b, \rho} \quad & \frac{1}{2} \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m} + C \sum_{i=1}^{l} \xi_i - \rho \\ \text{s. t.} \quad & \mathrm{y}_i (\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b) \geq \rho - \xi_i, \xi_i \geq 0, \\ & 0 \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m^p \leq 1. \end{aligned} \tag{2.15}$$

or equivalently in the mixed norm as[5]:

$$\begin{aligned} \min_{\{f_m\}, b, \rho} \quad & \frac{1}{2} \Big(\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}^{\frac{2p}{p+1}} \Big)^2 + C \sum_{i=1}^{l} \xi_i - \rho \\ \text{s. t.} \quad & \mathrm{y}_i (\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b) \geq \rho - \xi_i, \xi_i \geq 0. \end{aligned} \tag{2.16}$$

- *Switch between the constraint and regularization*

  Another equivalent form of the ℓ_pMKL is still in the primal form, which is

$$\begin{aligned} \min_{\boldsymbol{\mu}, \{f_m\}, b, \rho} \quad & \frac{1}{2} \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m} + C \sum_{i=1}^{l} \xi_i - \rho + \lambda (\sum_{m=1}^{M} \mu_m^p)^{\frac{2}{p}} \\ \text{s. t.} \quad & \mathrm{y}_i (\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b) \geq \rho - \xi_i, \xi_i \geq 0, \\ & 0 \leq \boldsymbol{\mu}. \end{aligned} \tag{2.17}$$

  This form appears in [193],[194],[196] for $p = 1$, $p = 2$ and $p \geq 1$ respectively.

- *Elastic-net Regularization*

  Some recent works also focus on the combining the different regularization strategies directly to get the new formulations for Multiple Kernel Learning, such as the [228][185][138][103] [156].

  Yang et el. [228] proposed to unify the ℓ_1MKL and the ℓ_2MKL by introducing the new constraint as $\boldsymbol{\mu} \in \{\boldsymbol{\mu} | 0 \leq \boldsymbol{\mu}, \theta \sum_{m=1}^{M} \mu_m + (1-\theta) \sum_{m=1}^{M} \mu_m^2 \leq 1\}$, where

[5] If $(\sum_{m=1}^{M} \mu_m^p)^{\frac{1}{p}} \leq 1$, then the corresponding objective function can be formulated as $\frac{1}{2} \Big(\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}^{\frac{2p}{p+1}} \Big)^{\frac{1+p}{p}} + C \sum_{i=1}^{l} \xi_i - \rho$, which is exactly the one as in [222].

$v \in [0, 1]$ is the parameter that balances the ℓ_1-norm and ℓ_2-norm for the kernel coefficients.

Besides the above developments for Multiple Kernel Learning, there are also works for extending the kernel learning methods for other kernel-based methods. Such as the multiple kernel learning for dimensional reduction [135][227], clustering [240], large margin nearest neighbor [145], fisher discriminant analysis [223].

Alignment Based Kernel Learning methods The notion of the Kernel Target Alignment (KTA) was proposed by Cristianini *et al.* [48]. The kernel target alignment is defined as

$$\hat{\rho}(\mathbf{K}, \mathbf{Y}) = \frac{\langle \mathbf{K}, \mathbf{y}\mathbf{y}' \rangle_F}{\sqrt{\langle \mathbf{K}, \mathbf{K} \rangle_F \langle \mathbf{y}\mathbf{y}', \mathbf{y}\mathbf{y}' \rangle_F}}. \tag{2.18}$$

The empirical value of this definition can be viewed as the cosine of the angle between the kernel $\mathbf{K}$ and the target ideal kernel $\mathbf{y}\mathbf{y}'$, and reveals the similarity between the kernel and the target. This criteria has been applied as the criteria for learning the kernel coefficients in [114, 115]. By substituting the $\mathbf{K} = \sum_{m=1}^{M} \mu_m \mathbf{K}_m$ into (2.20) with the constraint $\boldsymbol{\mu} \geq 0$, the maximization of the alignment can be formulated as the following QCQP problem:

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \boldsymbol{\mu}' \boldsymbol{a} \\ \text{s.t.} \quad & \boldsymbol{\mu}' \mathbf{M} \boldsymbol{\mu} \leq 1, \\ & \boldsymbol{\mu} \geq 0, \end{aligned} \tag{2.19}$$

where $a_m = \mathbf{y}' \mathbf{K}_m \mathbf{y}, m \in \{1, \cdots, M\}$ and $\mathbf{M}_{pq} = \langle \mathbf{K}_p, \mathbf{K}_q \rangle_F, p, q \in \{1, \cdots, M\}$.

The recent work by Cortes [44] proposed to utilize the centered kernel[6] for maximizing the kernel alignment, and also propose to optimize the alignment based on the norm constraints such as the simplex constraint or the ℓ_2-ball constraint for learning the linear combination coefficients. Thus the maximization problem[7] is formulated as

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \frac{\boldsymbol{\mu}' \boldsymbol{a} \boldsymbol{a}' \boldsymbol{\mu}}{\boldsymbol{\mu}' \mathbf{M} \boldsymbol{\mu}} \\ \text{s.t.} \quad & ||\boldsymbol{\mu}|| = 1, \boldsymbol{\mu} \geq 0, \end{aligned} \tag{2.20}$$

[6]The centered kernel $\mathbf{K}^c$ is defined as $\mathbf{K}^c_{ij} = \mathbf{K}_{ij} - \frac{1}{l}\sum_{i=1}^{l} \mathbf{K}_{ij} - \frac{1}{l}\sum_{j=1}^{l} \mathbf{K}_{ij} + \frac{1}{l^2}\sum_{i,j=1}^{l} \mathbf{K}_{ij}$.

[7]The vector $\boldsymbol{a}$ and matrix $\mathbf{M}$ are calculated by using the centered base kernels correspondingly.

where the $||\boldsymbol{\mu}||$ is the general norm (*e.g.*,ℓ_1-norm and ℓ_2-norm).

The above problem is then formulated as the QP problem,

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \frac{1}{2}\mathbf{v}'\mathbf{M}\mathbf{v} - \mathbf{v}'\boldsymbol{a} \\ \text{s.t.} \quad & \mathbf{v} \geq 0, \end{aligned} \tag{2.21}$$

and then $\boldsymbol{\mu}$ is obtained as $\boldsymbol{\mu} = \mathbf{v}^\star/||\mathbf{v}^\star||$, where $\mathbf{v}^\star$ is the optimal solution to problem (2.21). After learning the kernel coefficients, the single kernel SVM can further be trained separately. Thus, learning the kernel and training the classifier are performed independently in two steps, and the computational complexity can be reduced greatly.

Generalization Bound The generalization bound for the standard ℓ_1MKL was firstly given in [114] with the complexity of $\tilde{\mathcal{O}}\left(\sqrt{(\frac{M}{\gamma^2})/l}\right)$, where M is the number of base kernels, γ is the margin of learnt classifier, and ℓ is the number of training samples. The generalization bound is the multiplicative between the number of base kernels M and the margin complexity term $\frac{1}{\gamma^2}$. The work in [183] developed a new bound with complexity of $\tilde{\mathcal{O}}\left(\sqrt{(d_\phi + \frac{1}{\gamma^2})/l}\right)$, where d_ϕ is the pseudodimension of the given kernel family and equals to the number of base kernels M for the linear combination of base kernels. In this new bound, the total complexity term between margin complexity term and the family-of-kernels term is shown to be additive rather than multiplicative as from [183]. In the recent work [88], [89], the bound was further improved to $\tilde{\mathcal{O}}\left(\sqrt{\left(\log(M) + \frac{1}{\gamma^2} + 2M_c\right)/l}\right)$, where M_c is the number of selected base kernels for the final learned classifier.

The generalization bound using Rademacher complexity has also been studied. In [43], the estimation error of the learnt classifier is upper bounded by a term of the order $\tilde{\mathcal{O}}\left(\sqrt{(\frac{\log(M)}{\gamma^2})/l}\right)$, which improves the bound from [183] for ℓ_1MKL due to the introduced $\log(M)$ term. The work in [88] further improves the Rademacher complexity bound for ℓ_1MKL to be the order of $\tilde{\mathcal{O}}\left(\sqrt{(\log(M) + \frac{1}{\gamma^2})/l}\right)$, which is tighter than the ones obtained in [183] and [43]. The Rademacher complexity for ℓ_pMKL was also studied in [43] and the local Rademacher complexity for ℓ_pMKL was studied in [100]. Based on the local Rademacher complexity, new algorithm was designed in [41]. The convergence rate in order of $\tilde{\mathcal{O}}(l^{\frac{\alpha}{\alpha+1}})$ with α being the minimum eigenvalue decay rate of the individual kernels was derived in [101] for ℓ_pMKL. New learning rates for both the ℓ_1MKL and elastic-net MKL were also derived in [184].

Applications The applications of the Multiple Kernel Learning algorithms can be categorized into traditional supervised learning and weakly supervised leaning. The traditional supervised learning mainly focuses on fusing/choosing the different types of features. Applications from directly applying the MKL algorithms include object recognition [22], image classification [193], [75], [157], [225], [219], speaker verification [138], computational biology [14]. A few works are based on the adaptation of the Multiple Kernel Learning algorithms such as DTMKL [57],AMKL [59] for domain adaptation, AFMKL [210] for action recognition and GA-MKL for image classification [225].

Recently, the MKL techniques have been applied to solve the Weakly Labeled Learning problems. These works formulate the mixed integer programming problems for inferring the labels of ambiguous data to be MKL problems. In this way, the original NP-hard problem is converted into convex optimization problem. These works include [133] for maximum margin clustering, [131] for semi-supervised learning, [130] for multi-instance learning. A unifying of these works is summarized in [132] as weakly labeled SVMs and the experiments on benchmark data sets show that the algorithms can achieve the state-of-the-art performance when compared with the traditional methods under the specific learning settings. The using of MKL techniques has been also applied to the tasks of text-based image retrieval (TBIR) [128], [126], [125], [56], heterogenous domain adaptation [127], video event recognition from multiple heterogenous sources [34], video event detection [113], high dimensional feature selection [187], multivariate performance measure [143], multitemplate learning for structured prediction [142], class hierarchy learning [229] [33], outlier detection [124] and multi-source domain adaptation [171].

2.3.2 Relationship to Multi-view Learning

Another learning setting with multiple data representation is the multi-view learning, which refers to learning with multiple views of data. It is related to multiple kernel learning but also is different from multiple kernel learning. In this part, we summarize the related work from the literature which are deemed as multi-view learning, and then discuss their connections and differences with multiple kernel learning.

Weakly Labeled Learning Setting In multi-view learning, the training data are represented with multiple views of features. Typically, a classifier f^v (*e.g.* an SVM classifier) is trained on the v-th view and the final classifier is fused by using the classifiers from all views, *i.e.* $\tilde{f}(\mathbf{x}) = \frac{1}{V}\sum_{v=1}^{V} f^v(\mathbf{x}^v)$.

The multi-view learning was originally proposed to solve semi-supervised learning problem. When training data is represented with multiple feature representations, researchers have developed many multi-view learning approaches to improve performance by utilizing information from different views [16], [153], [105], [178], [177], [234], [109], [110], [51], [20]. Most of those works (*e.g.*, co-training [16]) in multi-view learning were proposed for the multi-view SSL scenario. Most traditional multi-view learning methods were proposed for semi-supervised learning (SSL). One of the pioneering works is the co-training method [16], and it was originally proposed for the SSL problem with two views. The basic idea of co-training is to iteratively add some pseudo-labeled samples into the pool of labeled training samples to re-train the classifiers on both views. The pseudo-labeled samples are selected from the pool of unlabeled training samples, and are labeled by at least one classifier which has a confident prediction. Finally, the classifiers from different views are fused to perform the classification.

The co-training algorithm was further extended to co-EM [153], in which they label all the unlabeled data at each iteration without considering confidence. It was also extended by using SVMs as base classifiers in [20], which was subsequently adapted for unsupervised learning [51]. Co-training was also extended to tri-training [243] and co-forest [122] to handle more than two views. However, the co-training style algorithms work under strict assumptions that each view is sufficient to train a low-error classifier and both views are conditionally independent, which might not be satisfied on real world datasets [200]. Many works attempted to relax those assumptions from various perspectives, such as weak dependence [1], α-expansion [11], large diversity [200] and label propagation [201]. Recently, co-training with insufficient views has also been theoretically analyzed in [202]. Besides co-training style methods, other methods such as co-regularization based approaches [105], [178], [177], [234] were also proposed to train classifiers on different views based on a so-called *co-regularization* criterion, which is used to minimize the differences of the decision values from the classifiers on different views. Similar ideas have also been employed in multi-view clustering [109], [110].

Supervised Learning Setting Under the supervised learning setting for two-view learning from literature is the SVM-2K [65], which is based on the KCCA [81]. For SVM-2K, it employs the key idea of KCCA formulation, and enforces the co-regularization term so that the predictions from the two views are consistent for all the training samples. Specifically, we learn the target decision function $f^A(\mathbf{x}) = \mathbf{w}'\psi(\mathbf{x}) + b$ and $f^B(\mathbf{x}) = \mathbf{v}'\phi(\mathbf{x}) + \varrho$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\mathbf{v},\rho} \quad & \frac{1}{2}\left(||\mathbf{w}||^2 + ||\mathbf{v}||^2\right) + C^A \sum_{i=1}^{l} \xi_i^A + C^B \sum_{i=1}^{l} \xi_i^B + D \sum_{i=1}^{l} \eta_i, \\ \text{s.t.,} \quad & |\mathbf{w}'\psi(\mathbf{x}_i) + b - \mathbf{v}'\phi(\mathbf{x}_i) - \varrho| \leq \eta_i + \epsilon, \\ & \mathrm{y}_i(\mathbf{w}'\psi(\mathbf{x}_i) + b) \geq 1 - \xi_i^A, \\ & \mathrm{y}_i(\mathbf{v}'\phi(\mathbf{x}_i) + \varrho) \geq 1 - \xi_i^B, \\ & \xi_i^A \geq 0, \xi_i^B \geq 0, \end{aligned} \tag{2.22}$$

where C^A, C^B and D are the regularization parameters. The final classifier is given as $f(\mathbf{x}) = \frac{1}{2}(f^A(\mathbf{x}) + f^B(\mathbf{x}))$. The problem in (2.22) is converted into its dual form, and can be solved by using the QP solver. As more constraints have been introduced into the optimization problem, the complexity becomes high when the number of samples is large. Besides, the prediction will be more complex when compared with the classifiers obtained by using the MKL.

Connections between Multi-view Learning and MKL The two types of learning settings are all based on the multiple data representations, and fall into our learning with multiple representations setting. The MKL focus more on the algorithm representation point of view based on the multiple kernels [114]. Those multiple kernels can be constructed with different parameters for the same data representation as well as from different data representations such as multiple views. While the multi-view learning focus more on the original data representation, in which we have multiple original data. It is possible to construct different or multiple kernels for multi-view learning problems.

In the literature, the multiple kernel learning algorithms are mostly for supervised learning setting without specific learning assumptions, while the multi-view learning is

mostly for semi-supervised learning, and the views are required to be independent. Besides, the Multiple Kernel Learning learns a single classifier while multi-view learning usually introduces one classifier for one view, and learns multiple classifiers simultaneously. In this way, most of the existing multi-view learning algorithms are only restricted to two-view cases [16], [105], [65], and extending them to more than three views usually suffers from high computational complexity [243] [122]. The final output for Multiple Kernel Learning is a single classifier which usually has one common dual variables for all the base kernels, but the final outputs for multi-view learning algorithms are classifiers corresponding to each of the considered views. In this way, the prediction by using the classifier from the multiple kernel learning is more efficient than that of the classifiers trained by using multi-view learning algorithms. Therefore, in terms of learning with multiple representations, they are similar. However, in terms of settings as well as efficiency, they differ from each other significantly.

2.4 Learning with Privileged Information

Another learning setting with multiple representations is the learning using privileged information (LUPI) recently proposed in [192]. This learning scenario introduces a "*teacher*" into the learning process, which supplies training example with additional information such as comments, explanation, description, logical reasoning, and so on. For instance, we want to classify biopsy images into cancer and non-cancer categories based on the given images captured by using the medical devices. At the same time, the available training images are possibly associated with a report written by a pathologist that describes the images using high level holistic descriptions.

In this setting, the additional privileged information is only available to the given training examples but it is not available for the new test examples. The LUPI learning setting applies to almost any existing learning problems as long as privileged information exists. The privileged information plays an important role in the learning process as it can significantly increase the speed of learning. In this Section, we first review the theoretical foundations for LUPI using SVM algorithm, and then review SVM+ as well as recent extensions and their applications to real-world tasks.

2.4.1 Problem Setting, Theory and Algorithms

Problem Setting Different from the learning with heterogeneous information setting such as multiple kernel learning and the multi-view learning, the learning with privileged information setting is recently proposed by [192], [161]. The learning setting assumes that training data in training set have additional information for the training task. Specifically, given a set of n training data $\{\mathbf{x}_i\}|_{i=1}^n$ with $\mathbf{x}_i \in \mathcal{R}^h \subset \mathcal{X}$, where h is the feature dimension of each sample, we refer to $\mathcal{X}$ as the *decision space* as suggested in [161] because the final decision is based on the features of the testing samples in the space $\mathcal{X}$. Except for the training data in the decision space $\mathcal{X}$, the additional privileged feature $\{\mathbf{z}_i\}|_{i=1}^n$ with $\mathbf{z}_i \in \mathcal{R}^g \subset \mathcal{Z}$ in the *correcting space* $\mathcal{Z}$ [161] is only available for the training set, but it is not available for the test set.

Generalization bound It is shown in [174] that the convergence rate of SVM based algorithms can be improved by using a correcting function to incorporate privileged information. Lets review the theoretic results from [192] in the following. For the binary classification task, suppose the given data in the decision space $\mathbf{x} \in \mathcal{X}$ is only associated with one of the two classes, and a best linear classifier $\mathbf{w}_0, b_0$ exists. There exists an oracle function $\xi(\mathbf{x}) = \max(0, 1 - \mathrm{y}_i(\mathbf{w}_0'\mathbf{x} + b_0))$ that can correctly classify all the training samples (*i.e.*, $\mathrm{y}_i(\mathbf{w}_0'\mathbf{x}_i + b_0) \geq 1 - \xi(\mathbf{x}_i)$). We have the following proposition[8] [192]:

Proposition 1 *If any vector $\mathbf{x} \in \mathcal{X}$ belongs to one and only one of the classes and there exists an Oracle function with respect to the best decision rule in the admissible set of hyperplanes, then with probability 1 - η the following bound holds true,*

$$\Pr(\mathrm{y}(\mathbf{w}'\mathbf{x} + b) < 0) \leq \Pr(1 - \xi(\mathbf{x}_i) < 0) + A\frac{h\ln\left(\frac{l}{h}\right) - \ln(\eta)}{l}, \tag{2.23}$$

where $Pr(\mathrm{y}(\mathbf{w}'\mathbf{x} + b) < 0)$ is the probability of error for the Oracle SVM solution on the training set of size ℓ, $Pr(1 - \xi(\mathbf{x}_i) < 0)$ is the probability of error for the best solution in the admissible set of functions, h is the VC dimension of the admissible set of hyperplanes and A is a constant.

[8] Proposition 1 in [192].

As we do not know either the values of slack variables or the oracle function, but we can utilize the privileged information $\mathbf{z}_i \in \mathcal{Z}$ to construct the correcting functions $\phi(\mathbf{z}, \delta)$ that have a low VC dimension, and the generalization bound using this correcting function is given as the following proposition[9] [192]:

Proposition 2 *If any vector $\mathbf{x} \in \mathcal{X}$ belongs to one and only one of the classes and there exists an Oracle function with respect to the best decision rule in the admissible set of hyperplanes, then with probability 1 - η the following bound holds true,*

$$\Pr(\mathrm{y}(\mathbf{w}'\mathbf{x} + b) < 0) \leq \Pr(1 - \phi(\mathbf{z}, \delta) < 0) + A\frac{(h+g)\ln\left(\frac{2l}{(h+g)}\right) - \ln(\eta)}{l}, \tag{2.24}$$

where $Pr(\mathrm{y}(\mathbf{w}'\mathbf{x}+b) < 0)$ is the probability of error for the training problem with training set of size ℓ, $Pr(1-\phi(\mathbf{z},\delta) < 0)$ is the probability of the event $\{\phi(\mathbf{z},\delta) > 1\}$, h is the VC dimension of the admissible set of hyperplanes, g is the VC dimension of the admissible set of correcting functions and A is a constant.

SVM+ In [174], the task is to utilize the training data $\{\mathbf{x}_i, \mathbf{z}_i\}|_{i=1}^n$ as well as their labels $\{\mathrm{y}_i\}|_{i=1}^n$ to train a classifier for classifying the test data $\{\mathbf{x}_i\}|_{i=n+1}^{n+m}$ under the SVM framework for the supervised binary classification problem. Specifically, the linear target classifier $f(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b$ is learnt on the decision space $\mathcal{X}$ only in order to classify the test data. At the same time, another function $\xi = \mathbf{v}'\mathbf{z} + \rho$ is learnt on the correcting space $\mathcal{Z}$ by modeling privileged information as the loss function. The objective function of SVM+ is proposed as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{v},b,\rho} \quad & \frac{1}{2}\left(||\mathbf{w}||^2 + \lambda||\mathbf{v}||^2\right) + C\sum_{i=1}^{l}(\mathbf{v}'\mathbf{z}_i + \rho) \\ \text{s.t.,} \quad & \mathrm{y}_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - (\mathbf{v}'\mathbf{z}_i + \rho), \forall i = 1, \ldots, l, \\ & \mathbf{v}'\mathbf{z}_i + \rho \geq 0, \, \forall i = 1, \ldots, l, \end{aligned} \tag{2.25}$$

where C and λ are the two regularization parameters.

[9] Proposition 2 in [192].

By introducing the lagrangian multipliers α_i's and β_i to 2.25, the dual form of the SVM+ can be formulated as the following optimization problem:

$$\begin{aligned}
\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{l} \alpha_i \alpha_j \mathrm{y}_i \mathrm{y}_j k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{2\lambda} \sum_{i=1,j=1}^{l} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) k^*(\mathbf{z}_i, \mathbf{z}_j) \\
\text{s.t.,} \quad & \sum_{i=1}^{l} (\alpha_i + \beta_i - C) = 0, \\
& \sum_{i=1}^{l} \alpha_i \mathrm{y}_i = 0, \\
& \alpha_i \geq 0, \beta_i \geq 0,
\end{aligned} \tag{2.26}$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $k^*(\mathbf{z}_i, \mathbf{z}_j)$ is the kernel function between $\mathbf{z}_i$ and $\mathbf{z}_j$. The optimization problem in (2.26) is in the form of a standard Quadratic Programming (QP) problem, which can be solved by using any state-of-the-art QP solvers. In [160], the Sequential Minimal Optimization (SMO) algorithm is developed to solve the dual form in (2.26) efficiently.

2.4.2 Extensions and Applications

In the original work of SVM+, the proposed algorithms are used for applications such as proteins classification in bioinformatics using the 3D-structures of proteins as privileged information, finance market prediction using future events as privileged information and digit recognition problem using the poetic description as privileged information. The empirical studies show that the privileged information improves the specific tasks. The SVM+ algorithm has also been utilized in [60] to perform the glaucoma detection task in medical images using the genetic information as privileged information. The work in [174] applies to SVM+ to the task of image classifications using the attributes, bounding box, textual descriptions or rationales as privileged information. The work in [35] utilizes the SVM+ to the task of object recognition in RGB images by using the depth images captured by the Kinect sensors as privileged information.

The work in [129] extends the SVM+ algorithm to multi-instance learning setting, and then applies the algorithm to the task of image categorization by using the descriptions to the image from the web as privileged information. Following the LUPI method [192], the

work in [68] extended [192] for the clustering problem, while the work in [174] extended it into the Ranking SVM for the ranking problem. The recent work in [69] proposed an extension of the learning scenario to distance metric leaning. However, their proposed method is a two-step approach to utilize privileged information [69]. They firstly trained a distance metric based on ITML using privileged information. Based on the distance metric learnt from privileged information, some pairs of training samples are removed. Then ITML is retrained again by using the remained pairs based on the main features.

2.5 Visual Representation for Computer Vision

The data representations play important roles for computer vision tasks. For any computer vision task, the features are needed to represent the data. Computer vision applications have evolved from the descriptions such as raw pixel values, edges information, color histogram to local histogram based descriptors such as SIFT, HOG and even descriptions learnt by using the deep learning methods such as the deep convolutional neural networks. In this Section, we review the recent progress for feature extraction methods for computer vision tasks and we also discuss the related learning scenarios such as the sparse representation, deep convolutional neural networks for the feature extraction.

2.5.1 Visual Representation for Vision Data

Low-level visual descriptions The low-level features for representing an image or a video can be simply the gray pixel value or other complex designed features. The global features incorporate the global information such as block-wise color moments for color, wavelet texture for textual information. The local features have been shown to obtain state-of-the-art results for plenty of vision applications. The GIST descriptor is proposed in [154] for scene recognition, and it represents the image by computing a wavelet image representation. The Scale Invariant Feature Transform (SIFT) descriptor [139] is the most prevailing local features in the recent years. The SIFT features are invariant to image scale and rotation, and are found to be robust [149] among the interest point descriptors and have been applied to almost all the computer vision applications. The Histograms of Oriented Gradient (HOG) [50] descriptor is another local descriptors and has been

proposed to the task of human detection task. The Local Binary Pattern (LBP) [2] has been proposed to encode the local structure of the pixel. The HOG-LBP [203] combines the HOG and LBP features and get improved performance for the human detection task with partial occlusion. Other local descriptors include the Self-similarity descriptors (SSIM) [176], Spatio-Temporal Interest Point (STIP) [116] for incorporate the temporal information from videos.

Bag of words model for visual Representation As the features such as SIFT, HOG are local features, and thousands of local descriptors could be extracted from each image or video, the local features have to be converted into a single feature vector that can effectively and efficiently represent the visual data. To this end, the bag of words model [179] has been successfully adapted from text representation to visual representation. In [118], the spatial pyramid matching is further proposed to encode the spatial information for visual representation. Since then, the procedures for representing an image can be summarized into the following procedures: *local feature extraction*, *dictionary learning*, *coding*, and *spatial pooling*. After obtaining the local descriptors, k-means algorithm is used to obtain the cluster centres of the local descriptors, and then each local descriptor is assigned to a cluster center, and finally spatial pooling is done to obtain a long dimensional feature vector. Lots of works in the literature are proposed to improve the procedure for image classification procedures from [179] and [118]. The most successful works are for the coding and spatial pooling stages. For the coding stage, some works uses learning algorithms to learn a better representation, and we will review these methods in the next subsection. For the pooling stage, a dense spatial sampling procedure [225] is proposed to improve the performance for image classification.

2.5.2 Representation Learning

Different from the handcrafted global features, lots of works have been developed for the learning of representations [15] for the visual data, which is shown to be effective in the recent years. These works include sparse representation, attribute learning (*i.e.*, classifier-based features) as well as deep convolutional neural networks.

Sparse Representation The sparse representation is based on the ℓ_1-norm regularization originally proposed in [189], and has been a popular topic recently. Following [189], the least angle regression [62] is further proposed to solve the ℓ_1-norm regularized least square problem efficiently. The ℓ_1-norm regularized least square problem enforce the regression coefficients to be as sparse as possible for each element of the coefficient. The group lasso is based on the ℓ_{21}-norm regularization, and is proposed in [235] for group variable selection. It also has been pointed out that the group lasso is a special case of multiple kernel learning [6]. Recently, there are also works that proposed dictionary learning methods [245] for visual recognition, utilized evolutionary computation methods [173] to image classification, and learned discriminative representations [137] for RGB-D video classification. These statistical models have been applied to the computer vision field for learning a better data representation.

In [208], the sparse representation is found to be perform well for the faces with occlusion. The sparse coding has been applied to the task of image super-resolution [230], image restoration [141]. The group sparse coding has also been applied to the task of human gait recognition [216]. For the visual feature extraction of the bag of words model, [231] formulates the quantization (pooling) stage as an ℓ_1-norm regularized least square problem, and use the regression coefficient to represent each of the local descriptors. The [199] further improves [231] in efficiency to code the local descriptors in a locally linear embedding (LLE) [166] manner. The Laplacian regularization is enforced to the ℓ_1-norm regularized least square problem in [72] [70] to enforce that similar descriptors have similar codes while the kernel trick [71] is applied to sparse representation to improve the results for face recognition and image classification.

Classifier-level Representation: Attributes and Pre-trained Classifiers Another high-level visual description is the attributes or the pre-trained classifiers. The attributes are based on the predictions from classifiers trained for specific meaning for describing the visual content. The binary attributes classifiers is obtained from classifiers trained to recognize the presence or absence of visual appearances such as age, race and gender for face verification task in [111], while the object bank [121] directly represents the images using the response map of a large number of pre-trained generic object detectors. The relative attributes [159] is also studied for the ranking problem. In these

works, the high-level representation is further utilized to perform the classification tasks and is found to improve the performance when compared with the simple low-level visual features. The fusing of the high-level representation with the low-level visual features is found to be helpful for applications such as video concept detection [220], content based-image retrieval [37], video event detection [59], action recognition [210], image classification [225] [226].

Deep Representations from Deep Convolutional Neural Networks The learning of visual features using deep neural networks has been found to outperform the handcrafted features recently. In [106], due to the "dropout" regularization procedure applied to the deep convolutional neural network as well as the large scale training data in ImageNet dataset [53], the performances from the obtained visual representation for image classification in ImageNet data set have been significantly improved. The DeCAF [54] has been released with the trained models on ImageNet to extract features for the other data sets such as object recognition on Caltech-101 [66], domain adaptation on Office data set [167], subcategory recognition on Caltech-UCSD birds dataset [205], and scene recognition on SUN-397 data set [211]. The representations from each layer of the deep convolutional networks are also visualized in [236].

2.6 Summary

In this Chapter, we have reviewed the related data representation using kernel and distance metric. Then we review the existing learning algorithms with single data representation. The learning algorithms are summarized into supervised learning such as SVM and weakly supervised learning including semi-supervised learning, multi-instance learning and clustering. After that, we review the related work for learning with multiple representations, which consists of learning with multiple kernels setting and learning with privileged information setting. The learning with multiple kernels is mainly about algorithms for multiple kernel learning, and it learns the classifier from multiple kernels that can be constructed from multiple representations. We review the learning algorithms for multiple kernel learning from literature and also discuss the connections and differences with multi-view learning. The learning with privileged information setting is the

case where training data has additional privileged information. We review the existing works for learning with privileged information. Finally, we review the visual representations for computer vision applications. We summarize the visual representation for vision data with handcrafted low-level visual representation, and learning-based representation. The learning based representation includes the classifier-based representation and deep representations from deep convolutional networks.

In this thesis, we focus on the learning with multiple representations. For the setting where heterogeneous information is available for both the training and test data with complete label information, in Chapter 3 we propose a unified Soft Margin Multiple Kernel Learning framework to learn a more robust classifier to utilize the available information from multiple representations. Compared with the existing works for MKL, our newly proposed hinge loss soft margin MKL(SM1MKL) is more robust to noisy base kernels. For the setting where multiple representations are available for both the training and test data, in Chapter 4 we propose a unified Input-output Kernel Learning (IOKL) to learn a more robust classifier under weakly labeled setting. Compared with the existing works for weakly labeled learning, we are the first work to study the weakly labeled learning with multiple representations. For the setting where training data have additional privileged information, in Chapter 5 we propose a distance metric learning with privileged information framework to learn a more robust distance metric. Compared with the existing work for distance metric learning, we are the first to model the privileged information in a unified objective function.

Chapter 3

Soft Margin Multiple Kernel Learning

Multiple Kernel Learning (MKL) has been proposed for kernel methods by learning the optimal kernel from a set of predefined base kernels. However, the traditional ℓ_1MKL method often achieves worse results than the simplest method using the average of base kernels (*i.e., average kernel*) in some practical applications. In order to improve the effectiveness of MKL, this chapter presents a novel soft margin perspective for MKL. Specifically, we introduce an additional slack variable called *kernel slack variable* to each quadratic constraint of MKL, which corresponds to one support vector machine model using a single base kernel. We first show that ℓ_1MKL can be deemed as hard margin MKL, and then we propose a novel soft margin framework for MKL. Three commonly used loss functions, including the hinge loss, the square hinge loss and the square loss, can be readily incorporated into this framework, leading to the new soft margin MKL objective functions. Many existing MKL methods can be shown as special cases under our soft margin framework. For example, the hinge loss soft margin MKL leads to a new box constraint for kernel combination coefficients. Using different hyper-parameter values for this formulation, we can inherently bridge the method using average kernel, ℓ_1MKL, and the hinge loss soft margin MKL. The square hinge loss soft margin MKL unifies the family of elastic net constraint/regularizer based approaches; and the square loss soft margin MKL incorporates ℓ_2MKL naturally. Moreover, we also develop efficient algorithms for solving both the hinge loss and square hinge loss soft margin MKL. Comprehensive experimental studies for various MKL algorithms on several benchmark

data sets, and two real world applications including video action recognition and event recognition demonstrate that our proposed algorithms can efficiently learn an effective yet sparse solution for MKL.

3.1 Introduction

Kernel methods such as Support Vector Machine (SVM) [24, 74], kernel principal component analysis have been shown as powerful tools for numerous applications. However, their generalization performances are often decided by the choice of the *kernel* [165, 168], which represents the similarity between two data points. For kernel methods, a poor kernel can lead to impaired prediction performance, thus many works [31, 48, 85, 115, 155, 190] have been proposed for learning the optimal kernel for kernel methods.

One of pioneering works for kernel learning was proposed to simultaneously train the SVM classifier and learn the kernel matrix [115]. However, learning the general kernel matrix is a non-trivial task. The learning problem is generally formulated as a semi-definite programming (SDP) problem, which suffers from the high computational cost even for one hundred training samples. Thus, this approach can be applicable for small scale data sets only. To reduce the computational cost, Lanckriet *et al.*[115] further assumed that the kernel is in the form of a linear combination of a set of predefined base kernels. Then the SVM classifier and the kernel combination coefficients are learned simultaneously, which is known as Multiple Kernel Learning (MKL). Since the proposed objective function has a simplex constraint for the kernel combination coefficients, it is also known as ℓ_1MKL.

There are two major research directions for MKL methods, in which the first one focuses on the development of efficient learning algorithms, while second one focuses on the improvement the generalization performance. For the first direction, Bach *et al.*[7] employed a sequential minimization optimization (SMO) method for solving medium-scale MKL problems. Sonnenburg *et al.*[182] applied a semi-infinite linear programming (SILP) strategy by reusing the state-of-the-art SVM implementations for solving the subproblems inside the MKL optimization more efficiently, which makes MKL applicable to large scale data sets. The similar SILP strategy was also used by [246] for multiclass

MKL problem. Following [182], the sub-gradient based method [164] and the level-method [221] have been proposed to further improve the convergence for solving MKL problems.

Although the optimization efficiency for ℓ_1MKL has been improved in recent years, Cortes *et al.*[42] and Kloft *et al.*[99] showed that the ℓ_1MKL formulation from [115] cannot achieve better prediction performance when compared with the simplest method using the average of base kernels (*i.e.*, average kernel) for some real world applications. To improve the effectiveness, lots of new MKL formulations [42], [59], [175], [99], [157], [99], [194], [138], [228], [103], [185], [44], [156], [158], [57], [210], [225] have recently been proposed.

The simplex constraint for the traditional ℓ_1MKL formulation usually yields a sparse solution. The recent works [42] and [99] conjectured that such a sparsity constraint may omit some useful base kernels for the prediction. Thereafter, they introduced a ℓ_2-norm constraint to replace the ℓ_1-norm constraint in ℓ_1MKL, leading to a non-sparse solution for the kernel combination coefficients. The ℓ_2-norm constraint was further extended to the ℓ_p-norm (p >1) constraint in [99]. Other MKL variants (*e.g.*, [193], [194], [196]) were proposed by removing the ℓ_1-norm constraint, while directly adding one regularization term based on the ℓ_1-norm, ℓ_2-norm or ℓ_p-norm of the kernel combination coefficients to the objective function, which are indeed equivalent to the formulation as in [99]. To further improve the efficiency of ℓ_pMKL, Xu *et al.*[222] and Kloft *et al.*[99] proposed an analytical way to update the kernel combination coefficients. The SMO strategy was also employed in [196] to accelerate the optimization for the ℓ_pMKL problem. In [138], a ℓ_2-norm regularizer of the kernel combination coefficients is directly added to the objective function while keeping the simplex constraint fixed. Alternatively, Yang *et al.*[228] used the elastic net regularizer on the kernel combination coefficients as a constraint for MKL. Note that the elastic net regularizer in the block norm form first appeared in [7] as a numerical tool for optimizing the ℓ_1MKL and was further discussed in [185] with a variant form. Moreover, the extensions of elastic net regularizer for MKL in primal form with more general block norm regularization were also discussed in [103] and in [156]. However, it is still unclear why these regularizers can enhance the prediction performances for MKL.

To answer this question, in this chapter, we first show that the traditional ℓ_1MKL can be deemed as hard margin MKL which only selects the base kernels with the minimum objective and throws away other useful base kernels. Then, we propose a novel soft margin perspective for MKL problems by starting from the dual of the traditional MKL method. The proposed soft margin framework for MKL is in analogy to the well-known soft margin SVM [45], which makes SVM robust in real applications by introducing a slack variable for each of the training data. Similarly, with the introduction of a slack variable for each of the base kernels, we propose three novel soft margin MKL formulations, namely, the hinge loss soft margin MKL, the square hinge loss soft margin MKL and the square loss soft margin MKL by using different loss functions.

The square loss soft margin MKL formulation incorporates ℓ_2MKL naturally. The square hinge loss soft margin MKL connects a few MKL methods using the elastic net like regularizers or constraints. The hinge loss soft margin MKL leads to a totally new formulation, which bridges ℓ_1MKL and the simplest approach based on the average kernel by using the different hyper-parameter values. These three cases reveal the connections between many independently proposed algorithms in the literature under our framework of soft margin MKL for the kernel learning, thus explaining why the regularization such as the ℓ_2-norm or the elastic net like regularizer/constraint help improve the performance over ℓ_1MKL in a new perspective.

In summary, the core contributions of this chapter are listed in the following:

(i) A novel soft margin framework for Multiple Kernel Learning is proposed. Particularly, a kernel slack variable is first introduced for each of the base kernels when learning the kernel. Three new MKL formulations, namely the hinge loss soft margin MKL, the square hinge loss soft margin MKL and the square loss soft margin MKL are also developed under this framework.

(ii) A new block-wise coordinate descent algorithm based on the analytical updating rule for learning the kernel combination coefficients is developed to efficiently solve the new hinge loss soft margin MKL problem. With our proposed framework, a simplex projection method is also introduced to solve the square hinge loss soft margin MKL, leading to a more efficient optimization procedure when compared with the existing optimization algorithms for elastic net MKL.

(iii) Comprehensive experimental results on the benchmark data sets and two video applications including real video event recognition and action recognition demonstrate the effectiveness and efficiency of our proposed soft margin MKL learning framework. Compared with ℓ_2MKL (ℓ_pMKL), the new hinge loss soft margin MKL and the square hinge loss soft margin MKL have much sparser solution for kernel combination coefficients; nevertheless, these two MKL models can achieve better generalization performance. This defends the efficiency using sparse kernel combination coefficients in MKL.

The chapter is organized as follows. In Section 3.2, we first review ν-SVM and MKL. In Section 3.3, a unified framework for soft margin MKL is proposed, and three novel soft margin MKL formulations are developed based on different loss functions for the kernel slack variables. New formulations for MKL are developed under our proposed soft margin MKL framework, and some existing formulations for MKL can also be revisited as the special cases under this framework. Then, a new block-wise coordinate descent algorithm for solving the hinge loss soft margin MKL and a simplex projection based algorithm for solving the square hinge loss soft margin MKL are introduced in Section 3.4. Experimental results on the standard benchmark data sets and the YouTube and Event6 data sets from computer vision applications are shown in Section 3.5. Finally, the conclusive remarks and the future work are presented in the last section.

3.2 Related works

3.2.1 ν-SVM

Given a set of labeled training data $S = \{(\mathbf{x}_i, \mathrm{y}_i)|i = 1, \ldots, l\}$ sampled independently from $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} \subset \mathcal{R}^n$ and $\mathcal{Y} = \{-1, +1\}$, a kernel matrix $\mathbf{K} \in \mathcal{R}^{l \times l}$ is usually constructed by using a mapping function $\phi(\mathbf{x})$ to map the data $\mathbf{x}$ from $\mathcal{X}$ to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ such that $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)$. Then, ν-SVM [170], [134], [27] learns the decision function:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i \mathrm{y}_i k(\mathbf{x}, \mathbf{x}_i) + b, \tag{3.1}$$

where α_i's are the coefficients associated with training samples, b is the bias of the decision function f. Let us define $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_l]'$ and $\mathbf{y} = [y_1, \cdots, y_l]'$. We minimize the model complexity $\|f\|_{\mathcal{H}}^2 = (\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}(\boldsymbol{\alpha} \odot \mathbf{y})$ and the training errors (represented by slack variables ξ_i's) for the decision function f simultaneously, then we arrive at the corresponding optimization problem[1]

$$\begin{aligned} \min_{\boldsymbol{\alpha}, b, \rho, \xi_i} \quad & \frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}(\boldsymbol{\alpha} \odot \mathbf{y}) + C\sum_{i=1}^{l} \xi_i - \rho \\ \text{s. t.} \quad & \mathrm{y}_i f(\mathbf{x}_i) \geq \rho - \xi_i,\ \xi_i \geq 0,\ \forall i = 1, \ldots, l, \end{aligned} \tag{3.2}$$

where $\rho/\|f\|_{\mathcal{H}}$ is the margin separation between two opposite classes and $C > 0$ is the regularization parameter. Note one can show that $C = \frac{1}{l\nu}$, where ν is the lower bound of fraction of outliers [170], [134], [28]. By using the duality property, it is easy to show that the dual of the objective in (3.2) is:

$$\max_{\boldsymbol{\alpha} \in A} \mathrm{SVM}\{\mathbf{K}, \boldsymbol{\alpha}\}, \tag{3.3}$$

where $\mathrm{SVM}\{\mathbf{K}, \boldsymbol{\alpha}\} = -\frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}(\boldsymbol{\alpha} \odot \mathbf{y})$ is the dual of the objective in SVM, and

$$A = \{\boldsymbol{\alpha} | \boldsymbol{\alpha}'\mathbf{1} = 1, \boldsymbol{\alpha}'\mathbf{y} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}\} \tag{3.4}$$

is the domain for $\boldsymbol{\alpha}$. From the Karush-Kuhn-Tucker (KKT) conditions of (3.2), one can show that the optimal solution $\boldsymbol{\alpha}^\star$ in the dual (3.3) is the same as that in the primal (3.2). Hence, for the given training set S and $\mathbf{K}$, the maximization of $\mathrm{SVM}\{\mathbf{K}, \boldsymbol{\alpha}\}$ with respect to $\boldsymbol{\alpha} \in A$ indeed gives the solution of the SVM classifier in (3.1).

3.2.2 ℓ_1MKL

Now, we review Multiple Kernel Learning (MKL) [7, 164]. With a set of given M base kernels $\mathcal{K} = \{\mathbf{K}_1, ..., \mathbf{K}_M\}$, ℓ_1MKL tries to learn the optimal kernel combination coefficients and the decision function f simultaneously. When the ν-SVM model is used, the primal problem of ℓ_1MKL with block norm regularization is written as:

$$\begin{aligned} \min_{f_m, b, \rho, \xi_i} \quad & \frac{1}{2}\Big(\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}\Big)^2 + C\sum_{i=1}^{l} \xi_i - \rho \\ \text{s. t.} \quad & \mathrm{y}_i\Big(\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b\Big) \geq \rho - \xi_i, \xi_i \geq 0. \end{aligned} \tag{3.5}$$

[1] Although this formulation looks different from the original one proposed in [170], they are essentially equivalent according to [27, 28].

The first term in (3.5) is the group lasso regularizer [6, 235] to choose groups of nonlinear features with small model complexity $\|f_m\|_{\mathcal{H}_m}$, in which each group of nonlinear features is induced by using one base kernel. By using the Lagrangian method, the dual of ℓ_1MKL is

$$\max_{\boldsymbol{\alpha}\in A,\tau} \tau \quad : \quad \mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \geq \tau, \forall m = 1, \ldots, M, \tag{3.6}$$

where $\mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} = -\frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha} \odot \mathbf{y})$. Alternatively, the dual (3.6) of ℓ_1MKL can also be written as:

$$\max_{\boldsymbol{\alpha}\in A} \min_{\boldsymbol{\mu}\in\mathcal{M}} \sum_{m=1}^{M} \mu_m \mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \tag{3.7}$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]'$, μ_m is the coefficient to measure the importance of the m^{th} base kernel, and $\mathcal{M} = \{\boldsymbol{\mu} | \mathbf{0} \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m = 1\}$ is the domain for $\boldsymbol{\mu}$. Then the final classifier is given by

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i \mathrm{y}_i \Big(\sum_{m=1}^{M} \mu_m k_m(\mathbf{x}, \mathbf{x}_i) \Big) + b.$$

3.2.3 The Hard Margin Perspective for ℓ_1MKL

From the constraints in (3.6), each SVM dual objective is no less than τ. The "error" is not allowed for each SVM dual objective which is below τ, and only the base kernels with the objective equal to τ are retained. In other words, the objective of ℓ_1MKL is essentially the same as $\max_{\boldsymbol{\alpha}\in A} \min_{m=1,\ldots,M} \mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\}$, which learns the SVM classifier by first choosing the model with the minimal objective. Ideally, only one base kernel will be chosen. Hence, ℓ_1MKL usually gets a very sparse solution for the kernel combination coefficients, and some useful base kernels may not be used. Since the constraints in (3.6) "push" the SVM dual objectives as large as possible, the variable τ can be considered as the *hard margin* in ℓ_1MKL, which reveals the hard margin property of ℓ_1MKL in the margin point of view, and paves the way for soft margin MKL formulations in the subsequent sections.

Recall that the non-sparse ℓ_2MKL [42] was proposed by substituting the simplex constraint with the ℓ_2-norm ball constraint $\boldsymbol{\mu} \in \{\boldsymbol{\mu} | \mathbf{0} \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m^2 \leq 1\}$. ℓ_pMKL

[99] directly extends the MKL formulation in (3.7) by simply substituting the simplex constraint with the ℓ_p-norm constraint $\boldsymbol{\mu} \in \{\boldsymbol{\mu} | \mathbf{0} \leq \boldsymbol{\mu}, \sum_{m=1}^{M} \mu_m^p \leq 1\}$ for $p > 1$. However, the kernel combination coefficients of these two models are always non-zeros, resulting in impaired prediction performance especially when many noisy or irrelevant base kernels are included. Therefore, how to remove noisy or irrelevant base kernels, and how to keep and emphasize the useful base kernels are the key issues for MKL methods.

3.3 A Soft Margin Framework for MKL

MKL learns the classifier and the optimal kernel simultaneously with a set of predefined base kernels. These base kernels can be obtained by using any well-known kernel functions (*e.g.* Gaussian kernel function, polynomial kernel function, spline kernel function, etc) with different kernel parameters or specially designed by domain experts for the learning task. Moreover, in computer vision tasks such as image classification or video event recognition, different types of features are extracted from lots of feature extraction methods (*e.g.* SITF [139], STIP [116], HOG [50], etc). Even with the same feature extraction method, there are many parameters. Usually, each type of features can be used to form a base kernel [75] for representing images/videos. However, only some base kernels are informative for classification, and others may be irrelevant or even harmful. Recent studies show that the combination of several features can achieve better prediction performance for computer vision applications. However, ℓ_1MKL usually chooses only one or few base kernels due to its hard margin property. On the other hand, we always obtain the dense solution for kernel combination coefficients by using ℓ_2-MKL, ℓ_p-MKL and the simplest method based on average kernel. Some noisy or irrelevant base kernels are inevitably included for prediction.

As pointed out in Section 3.2.3, ℓ_1MKL is indeed a hard margin MKL, which only selects the base kernels with the minimum objective. This could easily suffer from the over-fitting problem especially when some base kernels are formed by using noisy features. Recall that hard margin SVM assumes that the data of two opposite classes can always be separated with the hard margin, and the error is not allowed for training the model. However, to make SVM applicable for real applications, the slack variables were

introduced to hard margin SVM in [45]. The introduction of the slack variables allows some training errors for the training data, thus alleviating over-fitting encountered in hard margin SVM.

Inspired by the success of slack variables for SVM [45, 47], in this section, we introduce the concept of the *kernel slack variable* for each of the base kernels, and develop a soft margin MKL framework, which is the counterpart to soft margin SVM. Herein, we have the following definition:

Definition 3.2 *Given M base kernels $\mathcal{K} = \{\mathbf{K}_1, ..., \mathbf{K}_M\}$ for the training data $S = \{(\mathbf{x}_i, \mathrm{y}_i)|i = 1, \ldots, l\}$ sampled independently from $\mathcal{X} \times \mathcal{Y}$, we define kernel slack variable to be the difference of the target margin τ and the SVM dual objective $SVM\{\mathbf{K}_m, \boldsymbol{\alpha}\}$ for the given kernel $\mathbf{K}_m \in \mathcal{K}$ as*

$$\zeta_m = \tau - SVM\{\mathbf{K}_m, \boldsymbol{\alpha}\}, \forall m = 1, \ldots, M. \tag{3.8}$$

Then, the loss introduced from the kernel slack variable is defined as

$$\mathrm{z}_m = \ell(\zeta_m), \forall m = 1, \ldots, M, \tag{3.9}$$

where $\ell(\cdot)$ is any general loss function.

In the following, we mainly consider three loss functions, namely, the hinge loss (*i.e.*, $\ell(\zeta_m) = \max(0, \zeta_m)$), the square hinge loss (*i.e.*, $\ell(\zeta_m) = (\max(0, \zeta_m))^2$) and the square loss $\ell(\zeta_m) = \zeta_m^2$. Based on these loss functions on the kernel slack variables, we will present our proposed soft margin MKL formulations respectively. Note that our soft margin MKL framework can cater for not only the above-mentioned loss functions but also many other loss functions. These three loss functions are studied due to their simplicity and successful utilization in the standard soft margin SVM formulations.

3.3.1 Hinge Loss Soft Margin MKL

Based on the definition of the kernel slack variable for each base kernel, we are now ready to propose our soft margin MKL formulations. When the hinge loss is used for the kernel

slack variables, we have the following objective function for the *Hinge Loss Soft Margin MKL*:

$$\begin{aligned} \min_{\tau,\boldsymbol{\alpha}\in A,\zeta_m} \quad & -\tau + \theta \sum_{m=1}^{M} \zeta_m \\ \text{s. t. } & \text{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \geq \tau - \zeta_m, \zeta_m \geq 0, m = 1, \ldots, M. \end{aligned} \tag{3.10}$$

The objective of the above hinge loss soft margin MKL is to maximize the margin τ while considering the "errors" from the given M base kernels. The parameter θ balances the contribution of the loss term represented by slack variables ζ_m's and the margin τ. To further discover the properties of the newly proposed hinge loss soft margin MKL formulation, we have the following proposition:

Proposition 3 *The solution of the following optimization problem is also the solution of hinge loss soft margin MKL:*

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_1} \max_{\boldsymbol{\alpha}\in A} \quad \mathbf{J}(\boldsymbol{\mu}, \boldsymbol{\alpha}) \tag{3.11}$$

where the objective function is $\mathbf{J}(\boldsymbol{\mu}, \boldsymbol{\alpha}) = -\frac{1}{2}\sum_{m=1}^{M} \mu_m (\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y})$ *and* $\mathcal{M}_1 = \{\boldsymbol{\mu} | \sum_{m=1}^{M} \mu_m = 1, \mathbf{0} \leq \boldsymbol{\mu} \leq \theta \mathbf{1}\}$.

The proof of this proposition is shown in Appendix A.1. Note that the objective function $\mathbf{J}(\boldsymbol{\mu}, \boldsymbol{\alpha})$ is the same as the one in the hard margin MKL formulation, and the difference is in the constraint for the coefficients $\boldsymbol{\mu}$. In hard margin MKL, the constraint for $\boldsymbol{\mu}$ is the simplex constraint $\boldsymbol{\mu} \in \mathcal{M} = \{\boldsymbol{\mu} | \sum_{m=1}^{M} \mu_m = 1, \mathbf{0} \leq \boldsymbol{\mu}\}$. In contrast, we have $\boldsymbol{\mu} \in \mathcal{M}_1$ in our new hinge loss soft margin MKL. This new constraint enforces the values of the $\boldsymbol{\mu}$ no more than the regularization parameter θ, which can prevent extreme large values of kernel combination coefficients frequently encountered in hard margin MKL. We similarly observe the counterpart property of this formulation from the relationship between the hard margin SVM [19] and the hinge loss soft margin SVM [45]. For the hard margin SVM, the boundary constraint for $\boldsymbol{\alpha}$ is given by $0 \leq \boldsymbol{\alpha}$, while the constraint is $0 \leq \boldsymbol{\alpha} \leq C$ for the hinge loss soft margin SVM. Note C in the soft margin SVM and θ in our soft margin MKL are the regularization parameters that balance the training error and the complexity of the model.

We also have the following interesting observations for this new objective function:

(i) θ should be in the range $\{\theta|\theta \geq 1/M\}$, otherwise there is no solution to the proposed problem. This can be easily verified from the constraints in (3.11);

(ii) when $\theta = \frac{1}{M}$, according to constraint $\mathcal{M}_1$ in (3.11), we can obtain the uniform solution for $\boldsymbol{\mu}$ (*i.e.*, $\boldsymbol{\mu} = \frac{1}{M}\mathbf{1}$);

(iii) when $\theta \geq 1$, the constraint $\mathcal{M}_1$ in (3.11) becomes the same as $\mathcal{M}$ in the hard margin MKL (*i.e.*, ℓ_1MKL [164]).

We clearly observe that the structural risk function is well controlled by introducing the penalty parameter θ, and the solution of the MKL problem can be changed by varying this parameter, which gives a novel perspective to the MKL problems. This objective function also bridges ℓ_1MKL and the simple approach using average kernel by choosing different regularization parameter θ.

3.3.2 Square Hinge Loss Soft Margin MKL

When we define the loss function for the kernel slack variables as the square hinge loss, then we can arrive at the following objective function for the *Square Hinge Loss Soft Margin MKL*:

$$\begin{aligned} \min_{\tau,\boldsymbol{\alpha}\in A,\zeta_m} \quad & -\tau + \frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^2 \\ \text{s. t.} \quad & \mathrm{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\} \geq \tau - \zeta_m, m = 1,\ldots,M. \end{aligned} \tag{3.12}$$

Similar to the hinge loss soft margin MKL, τ is the margin of the final classifier, and each SVM's dual objective for the base kernels is lower bounded by the difference between the margin τ and the kernel slack variable ζ_m. We have the following proposition:

Proposition 4 *The solution of the following optimization problem is equivalent to that of square hinge loss soft margin MKL:*

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_2}\max_{\boldsymbol{\alpha}\in A} \quad \mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}) + \frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2, \tag{3.13}$$

where the function $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})$ *is* $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$ *and* $\mathcal{M}_2 = \{\boldsymbol{\mu}|\sum_{m=1}^{M}\mu_m = 1, \mathbf{0} \leq \boldsymbol{\mu}\}$.

The proof of this proposition is similar to that of Proposition 3, and thus it is omitted. Compared with ℓ_1MKL, this formulation shares the same simplex constraint for $\boldsymbol{\mu}$, but it has one more ℓ_2-norm regularization term $\frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2$ for the coefficients in the objective function. The regularization parameter θ balances the regularization for $\boldsymbol{\mu}$ and the margin of the classifier $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})$.

The relationship between the hard margin MKL and the square hinge loss soft margin MKL is also similar to that between hard margin SVM and the square hinge loss soft margin SVM [45], where the constraint for $\boldsymbol{\alpha}$ remains the same while one more regularization term $\frac{\sum_{i=1}^{l}\alpha_i^2}{2C}$ is added in the objective function of the hard margin SVM formulation.

Note the simplex constraint is removed from ℓ_2MKL to ℓ_1MKL. In contrast, this formulation still has the simplex constraint. The previous work [138] has used such type of regularization by directly adding the ℓ_2-norm regularization term for the kernel combination coefficients in the objective function of ℓ_1MKL. To further discover the connections of our square hinge loss soft margin MKL with previous works, we have the following proposition.

Proposition 5 *The primal form of the square hinge loss soft margin MKL is given as:*

$$\begin{aligned}\min_{\boldsymbol{\mu},f_m,b,\rho,\xi_i}\ & \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m}+C\sum_{i=1}^{l}\xi_i-\rho+\frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2 \\ s.\ t. & \mathrm{y}_i\Big(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b\Big)\geq\rho-\xi_i,\xi_i\geq 0,\sum_{m=1}^{M}\mu_m=1,\mathbf{0}\leq\boldsymbol{\mu}.\end{aligned} \tag{3.14}$$

Proof: With fixed $\boldsymbol{\mu}$, we can write the Lagrangian as $\mathcal{L}=\frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m}+C\sum_{i=1}^{l}\xi_i-\rho+\frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2-\sum_{i=1}^{l}\alpha_i\Big(\mathrm{y}_i(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b)-\rho+\xi_i\Big)-\sum_{i=1}^{l}\xi_i\beta_i$, where $\alpha_i\geq 0$, $\beta_i\geq 0$ are the Lagrange multipliers of the corresponding constraints. By setting the derivatives of the primal variables f_m, b, ρ, ξ_i to be zeros, we can get the corresponding KKT conditions. By replacing the primal variables in the Lagrangian with the KKT conditions, we can arrive at the min max optimization problem as shown in (3.13). Together with Proposition 4, we prove the proposition.

In the primal form, the objective function can be denoted as $f=\arg\min_f\Omega(f)+R_{emp}(f)$, where $\Omega(f)$ is the regularization term for the functional f, $R_{emp}(f)$ is the empirical risk term from the given training samples. Specifically, for (3.14), we have

$\Omega(f) = \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m} + \frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2$ with $\boldsymbol{\mu} \in \mathcal{M}_2$ and $R_{emp}(f)$ is the standard hinge loss from the training samples. In this formulation, we can see that the ℓ_1-norm of the kernel combination coefficients is enforced in the constraint, and the ℓ_2-norm of the kernel combination coefficients is penalized in $\Omega(f)$. Therefore, it is essentially the elastic net regularization [247] for MKL. In [228], the regularization is given as $\Omega(f) = \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m}$ under the elastic net constraint $v\sum_{m=1}^{M}\mu_m + (1-v)\sum_{m=1}^{M}\mu_m^2 \leq 1, \boldsymbol{\mu} \geq \mathbf{0}$. This can be regarded as a variant of (3.14) after considering the general conversion between Tikhonov regularization and Ivanov regularization as shown in Theorem 1 from [99].

Several existing works [7, 103, 156, 175, 185] have also been proposed for MKL with the (generalized) elastic net regularization in the primal form with the block norm regularization, without explicitly containing the kernel combination coefficients $\boldsymbol{\mu}$. For instance, the work in [7] utilized the regularization term $\Omega(f) = \frac{\lambda}{2}\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}^2 + \frac{1}{2}\left(\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}\right)^2$ to facilitate the optimization of the ℓ_1MKL problem. Interestingly, it can be shown that the primal form of square hinge loss soft margin MKL in (3.14) is equivalent to the formulations in [7, 175, 185] by seeking the entire regularization path [8].

Note that, the square norm $\sum_{m=1}^{M}\zeta_m^2$ for the kernel slack variables ζ_m in (3.12) can be readily extended to a more general norm $\sum_{m=1}^{M}\zeta_m^{\frac{p}{p-1}}$ with $1 < p < \infty$ in a similar fashion as from ℓ_2MKL to ℓ_pMKL (see Section 3.3.3 for more discussions). We can then obtain a more general $\frac{p}{p-1}$-hinge loss soft margin MKL as $\min_{\tau,\boldsymbol{\alpha}\in A,\zeta_m} -\tau + \frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^{\frac{p}{p-1}}$ s.t. $\text{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \geq \tau - \zeta_m, \zeta_m \geq 0, m = 1, \ldots, M$. The extensions of elastic net MKL in the primal form with more general block norm regularization are also discussed in [103, 156], which can also be deemed as the soft margin MKL.

3.3.3 Square Loss Soft Margin MKL

By setting the margin $\tau = 0$ in (3.12), the loss function for the kernel slack variables becomes the square loss, and we can get the following *Square Loss Soft Margin MKL*:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in A,\zeta_m} \quad & \frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^2 \\ \text{s. t.} \quad & -\text{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \leq \zeta_m, m = 1, \ldots, M, \end{aligned} \tag{3.15}$$

The ℓ_2MKL comes out naturally from (3.15) under our soft margin MKL framework according to the proposition:

Proposition 6 *The solution of the following problem gives the solution of square loss soft margin MKL:*

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_3}\max_{\boldsymbol{\alpha}\in A} \quad \mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}), \tag{3.16}$$

where the function $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})$ *is* $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$ *and* $\mathcal{M}_3 = \{\boldsymbol{\mu}|\sum_{m=1}^{M}\mu_m^2 \leq 1, \mathbf{0}\leq\boldsymbol{\mu}\}$.

Proof: It can be also proven by introducing the Lagrangian multipliers, *i.e.,* the dual variables μ_m for each of the inequality constraint, we can arrive at the following dual form: $\max_{\boldsymbol{\mu}\geq 0}\min_{\boldsymbol{\alpha}\in A} -\frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2 + \frac{1}{2}\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$. By using Theorem 1 from [99], it is easy to show that for one specific parameter θ the above optimization problem is equivalent to: $\min_{\boldsymbol{\mu}\geq 0,\boldsymbol{\mu}'\boldsymbol{\mu}\leq 1}\max_{\boldsymbol{\alpha}\in A} -\frac{1}{2}\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$, which is essentially ℓ_2MKL [42], [99]. Thus we conclude that our proposed soft margin MKL framework also incorporates ℓ_2MKL as one special case.

By considering a more generalized norm on ζ_m beyond the ℓ_2-norm, the formulation can be further extended to $\min_{\boldsymbol{\alpha}\in A,\zeta_m}\frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^{\frac{p}{p-1}}$ s.t. $-\text{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\}\leq\zeta_m, m = 1,\ldots,M$, which can be similarly reformulated as ℓ_pMKL ($p>1$) [99]. In general, ℓ_pMKL [99] can be regarded as a special case of our soft margin MKL as well.

3.4 Optimization for Soft Margin MKL

In this section, we propose new optimization algorithms for our proposed soft margin MKLs.

As the optimization problems can be changed to the min max optimization problem, we adopt the alternating optimization approach, which was widely used in previous works [164, 182], to alternatively learn the kernel combination coefficients and the model parameter by leveraging the standard SVM implementations. Note that the recent works [99, 222] proposed a new analytical updating rule for ℓ_pMKL by considering the special structure in the primal form of ℓ_pMKL. This type of solution can avoid the time consuming procedure for searching the new updating point for the kernel combination coefficients.

Although the convergence when using $p = 1$ is not proven, the stable convergence to the optimal solution was experimentally observed in [99, 222]. Besides, the similar analytical updating rule was also adopted in [186]. In this chapter, we also propose a new analytical solution for updating the kernel combination coefficients based on the structure of our new objective function for the hinge loss soft margin MKL. For the square hinge loss soft margin MKL, a simplex projection method is proposed.

3.4.1 Block-wise coordinate descent algorithm for solving the primal hinge loss soft margin MKL

We have the following proposition:

Proposition 7 *The following problem is the primal form for hinge loss soft margin MKL:*

$$\begin{aligned} \min_{\boldsymbol{\mu}\in\mathcal{M}_1, f_m, b, \rho, \xi_i} \quad & \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m} + C\sum_{i=1}^{l}\xi_i - \rho \\ s.\ t. \quad & \mathrm{y}_i\Big(\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b\Big) \geq \rho - \xi_i, \xi_i \geq 0. \end{aligned} \tag{3.17}$$

Proof: The Lagrangian can be written as:

$$\begin{aligned} \mathcal{L} \;=\; & \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m} + C\sum_{i=1}^{l}\xi_i - \rho \\ & - \sum_{i=1}^{l}\alpha_i\Big(\mathrm{y}_i\Big(\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b\Big) - \rho + \xi_i\Big) - \sum_{i=1}^{l}\xi_i\beta_i \\ & - \sum_{m=1}^{M}\mu_m\eta_m - \sum_{m=1}^{M}\zeta_m(\theta - \mu_m) + \tau\Big(\sum_{m=1}^{M}\mu_m - 1\Big), \end{aligned} \tag{3.18}$$

where $\alpha_i \geq 0$, $\beta_i \geq 0$, $\eta_m \geq 0$, $\zeta_m \geq 0$ and τ are the Lagrange multipliers for the corresponding constraints.

By setting the derivatives of the Lagrangian in (3.18) with respect to the primal variables f_m, b, ρ, ξ_i and μ_m to be zeros, and substituting the primal variables back into

the Lagrangian according to the corresponding KKT conditions, we have:

$$\begin{aligned} \max_{\tau,\boldsymbol{\alpha}\in A,\zeta_m} \quad & -\tau - \theta \sum_{m=1}^{M} \zeta_m \\ \text{s. t.} \quad & -\frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) \geq -\tau - \zeta_m, \\ & \zeta_m \geq 0, m = 1, \ldots, M. \end{aligned} \tag{3.19}$$

By multiplying -1 to the objective function in (3.19), it is converted into a minimization problem. Substituting τ with $-\tau$, we arrive at the same formulation as the hinge loss soft margin MKL in (3.10). This completes the proof.

In the primal form as in (7), we have the box constraint for $\boldsymbol{\mu}$, *i.e.*, $\boldsymbol{\mu} \in \mathcal{M}_1 = \{\boldsymbol{\mu} | \sum_{m=1}^{M} \mu_m = 1, \mathbf{0} \leq \boldsymbol{\mu} \leq \theta \mathbf{1}\}$. The work in [148] proposed a family of structured sparsity to improve the lasso for linear regression problem. Specifically, a box constraint is directly enforced on the unknown regression variables to enforce the structured sparsity. By simplifying our model to the linear case without the group structure [6, 235], we could incorporate [148] as a special case.

The primal problem in (3.17) is convex in the objective function [164] and linear in the constraints, thus it is convex. It can be solved by using the block-wise coordinate descent algorithm [99, 222].

3.4.1.1 Fix $\boldsymbol{\mu}$, update f_m, b, ρ, ξ_i

With a fixed $\boldsymbol{\mu}$, the optimization problem in (3.17) becomes a standard maximum margin SVM problem, which can be equivalently reformulated as a standard Quadratic Programming (QP) problem with respect to $\boldsymbol{\alpha}$ as shown in (3.20), and many efficient QP solvers can be readily used to obtain the optimal $\boldsymbol{\alpha}$.

$$\max_{\boldsymbol{\alpha}\in A} \quad -\frac{1}{2} \sum_{m=1}^{M} \mu_m (\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) \tag{3.20}$$

After obtaining the optimal $\boldsymbol{\alpha}$, the primal variables f_m, b, ρ, ξ_i can be recovered accordingly.

3.4.1.2 Fix f_m, b, ρ, ξ_i, update $\boldsymbol{\mu}$

With fixed f_m, b, ρ, ξ_i, the optimization problem in (3.17) reduces to the following convex programming problem:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_1} \quad \sum_{m=1}^{M} \frac{a_m}{\mu_m} \tag{3.21}$$

with $a_m = \frac{1}{2}\|f_m\|^2_{\mathcal{H}_m} = \frac{1}{2}\mu_m^2(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$.

The remaining problem is how to efficiently solve the subproblem (3.21). Let us suppose all the base kernels are semi-positive definite, and then we have $a_m > 0$ for $m = 1, \cdots, M$. Without the loss of generality, we also assume that a_m has been sorted such that $a_1 \geq a_2 \geq \cdots \geq a_M$. Inspired by the Lagrangian multipliers method [172] used for simplex projection, we introduce the Lagrangian multipliers λ, η_m's, and ζ_m's for the constraints in (3.21). Then we can get the following Lagrangian:

$$\begin{aligned} \mathcal{L} &= \sum_{m=1}^{M} \frac{a_m}{\mu_m} - \sum_{m=1}^{M} \mu_m\eta_m + \sum_{m=1}^{M} \zeta_m(\theta - \mu_m) \\ &+ \lambda(\sum_{m=1}^{M} \mu_m - 1). \end{aligned} \tag{3.22}$$

Setting the derivative of $\mathcal{L}$ with respect to μ_m to be zeros, we have the following KKT condition,

$$-\tfrac{a_m}{\mu_m^2} - \eta_m - \zeta_m + \lambda = 0, \tag{3.23}$$

with the complementary KKT conditions $\mu_m\eta_m = 0$, $\zeta_m(\theta - \mu_m) = 0$, and $\lambda(\sum_{m=1}^{M}\mu_m - 1) = 0$.

Thus, for $0 < \mu_m < \theta$, we have

$$-\tfrac{a_m}{\mu_m^2} + \lambda = 0, \text{ or } \mu_m = \sqrt{\tfrac{a_m}{\lambda}}. \tag{3.24}$$

If $a_m > 0$, we have $\mu_m > 0$. Thus, for the case that all the a_m's are larger than 0, the constraint $0 \leq \mu_m$ can be replaced by $0 < \mu_m$. If we know ω, the number of elements in $\boldsymbol{\mu}$ whose value strictly equals to θ, the solution of the above problem can be directly obtained as:

$$\mu_m = \begin{cases} \theta & m \leq \omega \\ \frac{(1-\omega\theta)\sqrt{a_m}}{\sum_{p=\omega+1}^{M}\sqrt{a_p}} & m > \omega \end{cases}. \tag{3.25}$$

We have the following two lemmas to obtain the solution for the problem in (3.21).

Lemma 3.1 *Let $\boldsymbol{\mu}^*$ be the optimal solution to problem (3.21), and suppose $a_p > a_q$ for any two given indices p,q $\in \{1, \cdots, M\}$. If $\mu_q^* = \theta$, then we have $\mu_p^* = \theta$.*

Proof: Suppose that $\boldsymbol{\mu}^*$ is the optimal solution to the problem in (3.21), and $\mu_q^* = \theta$. If using proof by contradiction, we have $\mu_p^* < \theta$. Let $\tilde{\boldsymbol{\mu}}$ be another vector whose elements have the same value with $\boldsymbol{\mu}^*$ except that $\tilde{\mu}_p = \mu_q^*$ and $\tilde{\mu}_q = \mu_p^*$. Then, we observe that $\tilde{\boldsymbol{\mu}}$ satisfies all the constraints in (3.21). Thus, $\sum_{m=1}^{M} \frac{a_m}{\mu_m^*} - \sum_{m=1}^{M} \frac{a_m}{\tilde{\mu}_m} = \frac{a_p}{\mu_p^*} + \frac{a_q}{\mu_q^*} - \frac{a_p}{\tilde{\mu}_p} - \frac{a_q}{\tilde{\mu}_q} = (a_p - a_q)(\frac{1}{\mu_p^*} - \frac{1}{\theta}) > 0$. So we have $\sum_{m=1}^{M} \frac{a_m}{\mu_m^*} > \sum_{m=1}^{M} \frac{a_m}{\tilde{\mu}_m}$, which contradicts with the assumption that $\boldsymbol{\mu}^*$ is the optimal solution to (3.21). So the original assumption is incorrect and we thus complete the proof.

Lemma 3.2 *Let $\boldsymbol{\mu}^*$ be the optimal solution to the problem in (3.21), and suppose that $a_1 \geq a_2 \geq \cdots \geq a_M$. Then ω, the number of elements whose value strictly equals to θ in $\boldsymbol{\mu}^*$, is*

$$\min \Big\{ p \in \{0, 1, \cdots, M-1\} \Big| \frac{\sqrt{a_{p+1}}(1 - p\theta)}{\sum_{m=p+1}^{M} \sqrt{a_m}} < \theta \Big\}.$$

The proof is similar with that of Lemma 3.1 by using the proof by contradiction and thus it is omitted here.

3.4.1.3 The whole optimization procedure

Based on the above derivations, we can easily develop the whole optimization procedure for the hinge loss soft margin MKL, and the detailed block-wise coordinate descent algorithm is shown in Algorithm 1.

3.4.2 Simplex projection method for solving the square hinge loss soft margin MKL

For solving the square hinge loss soft margin MKL, we directly solve the problem in (3.13). With a fixed $\boldsymbol{\mu}$, the optimization problem with respect to $\boldsymbol{\alpha}$ is a standard QP problem, which can be optimized by using the QP solver.

With a fixed $\boldsymbol{\alpha}$, the projected gradient descent based algorithm is used to update the kernel combination coefficients. Following [31], the gradient $\mathbf{p}^t$ of the optimization problem in (3.13) with respect to $\boldsymbol{\mu}$ can be calculated as

$$\mathrm{p}_m = -h_m + \frac{1}{\theta}\mu_m, m = 1, \ldots, M, \tag{3.26}$$

Algorithm 1 : Procedure of the block-wise coordinate descent algorithm for hinge loss soft margin MKL

1: Initialize $\boldsymbol{\mu}^1$.
2: $t = 1$
3: **while** stop criteria is not reached **do**
4: Obtain $\boldsymbol{\alpha}^t$ by solving the subproblem in (3.20) using the standard QP solver with $\boldsymbol{\mu}^t$
5: Calculate a_m and update $\boldsymbol{\mu}^{t+1}$ by solving the subproblem in (3.21)
6: $t = t + 1$
7: **end while**

Algorithm 2 : Procedure of the iterative approach for square hinge loss soft margin MKL

1: Initialize $\boldsymbol{\mu}^1$.
2: $t = 1$.
3: **while** stop criteria is not reached **do**
4: Obtain $\boldsymbol{\alpha}^t$ by solving the subproblem in (3.20) using the standard QP solver with $\boldsymbol{\mu}^t$
5: Calculate $\boldsymbol{\mu}^{\star}_{sub}$ that can reduce the objective function value for the problem in (3.13)
6: Update $\boldsymbol{\mu}^{t+1} = \boldsymbol{\mu}^{\star}_{sub}$
7: $t = t + 1$
8: **end while**

where $h_m = \frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y})$.

Then, the coefficient $\boldsymbol{\mu}$ is updated by using the coefficients $\boldsymbol{\mu}^t$ at the current iteration, namely,

$$\boldsymbol{\mu}^{\star}_{sub} = \Pi_{\mathcal{M}_2}(\boldsymbol{\mu}^t - \eta_t \mathbf{p}^t), \tag{3.27}$$

where $\Pi_{\mathcal{M}_2}(\cdot)$ is the simplex projection operation and η_t is the updating step size determined by the standard line search strategy. The simplex projection operation is a standard QP problem, which can also be solved by using the general QP solver. However, due to the special simplex constraint for $\boldsymbol{\mu}$, the efficient simplex projection method in [61] is used in this work The detailed optimization procedure is shown in Algorithm 2.

3.4.3 Computational Complexity

Now we can analyze the computational complexity for Algorithm 1 and Algorithm 2, we observe that obtaining $\boldsymbol{\alpha}^t$ by solving the QP subproblem in (3.20) shares the similar form with the standard SVM, therefore its complexity is just $O(n^{2.3})$. For computing the a_m's, the complexity is $MO(n)$, and for updating $\boldsymbol{\mu}$, the complexity is just $O(M \log(M))$. Therefore, for one iteration, the complexity is $O(n^{2.3} + Mn + M \log(M))$, and the whole complexity is $O(MKL) = T \times O(n^{2.3} + Mn + M \log(M))$. The different types of regularization may lead to different number of iterations, and it is still a very challenging issue to estimate the number of iterations, and therefore it is interesting to further study the convergence rate of the different MKL algorithms.

3.5 Experiments on real world data sets

In this section, we first evaluate different MKL algorithms on the benchmark data sets. Then we show the experimental studies on two real computer vision applications (*i.e.*, video action recognition and video event recognition).

3.5.1 Comparison algorithms

We evaluate the following algorithms:

(i) *AveKernel*: we use average combination of the base kernels. Specifically the kernel combination coefficients is given by $\boldsymbol{\mu} = \frac{1}{M}\mathbf{1}$, then the maximum margin classifier is learnt by SVM;

(ii) *SimpleMKL* [164]: the classifier and the kernel combination coefficients are optimized by solving the ℓ_1MKL problem as in (3.5);

(iii) ℓ_2*MKL* [42, 99]: the classifier and the kernel combination coefficients are optimized under the constraint $||\boldsymbol{\mu}||_2 \leq 1$;

(iv) ℓ_p*MKL* [99]: the classifier and the kernel combination coefficients are optimized under the constraint $||\boldsymbol{\mu}||_{\mathrm{p}} \leq 1$ with $p \geq 1$;

(v) *SGMKL* [228]: the sparse generalized multiple kernel learning as in [228], where the constraint for the kernel combination coefficients is the elastic net constraint, *i.e.*, $v||\boldsymbol{\mu}||_1 + (1-v)||\boldsymbol{\mu}||_2 \leq 1$ with $0 \leq v \leq 1$;

(vi) *SM1MKL*: our proposed hinge loss soft margin MKL, in which the classifier and the kernel combination coefficients are optimized by solving the hinge loss soft margin MKL problem;

(vii) *SM2MKL*: the square hinge loss soft margin MKL, in which the classifier and the kernel combination coefficients are optimized by solving the square hinge loss soft margin MKL problem.

To be consistent with previous works [164], [228], [99], the experiments for different MKL algorithms are all based on the C-SVC formulation as used in [164], and the SVM QP problem is solved by using the LibSVM C-SVC QP solver[2]. For the SimpleMKL codes downloaded from the web[3], we additionally change the SVM solver in their implementation with the LibSVM QP solver. For ℓ_pMKL, the implementation is available in Shogun toolbox [181], however we implement the algorithm by using the analytical updating rule for the kernel combination coefficients exactly as in [99, 222] for better utilization of the LibSVM QP solver for fair comparison. For SGMKL [228], we download

[2]http://www.csie.ntu.edu.tw/ cjlin/libsvm/
[3]http://asi.insa-rouen.fr/enseignants/∼arakotom/code/mklindex.html

their matlab implementation[4], and replace the SVM QP solver with the LibSVM QP solver, and also use Mosek[5] to solve the subproblem for updating the kernel combination coefficients in their implementation.

The SVM regularization parameter C is set in the range of $\{0.01, 0.1, 1, 10, 100\}$ for all the algorithms on all the data sets in the following experiments. One more model parameter p is introduced for ℓ_pMKL, v is introduced for SGMKL and θ is introduced for SM1MKL and SM2MKL. These parameters are set as follows:

(i) for ℓ_pMKL, $p \in \{1, 32/31, 16/15, 8/7, 4/3, 2, 3, \infty\}$;

(ii) for SGMKL, v is in the range of $\{0, 0.1, 0.2, \ldots, 1\}$;

(iii) for SM1MKL, θ is set to be $\frac{1}{\nu M}$, where ν is a ratio parameter from $\{1/M, 0.1, 0.2, \ldots, 1\}$;

(iv) for SM2MKL, θ is in the range of $\{10^{-5}, \ldots, 10^4, 10^5\}$.

Then all the algorithms have multiple sets of parameters, and the optimal parameters are determined by using five-fold cross validation on the training set.

3.5.2 Experiments on benchmark data sets

We first evaluate our proposed algorithms on some benchmark data sets. The experiments are conducted on seven publicly available data sets [6,7], which are *Heart*, *Diabetes*, *Australian*, *Ionosphere*, *Ringnorm*, *Banana* and *FlareSolar*.

3.5.2.1 Experimental settings

For the construction of base kernels on these benchmark data sets, we follow the method in [164] by designing the base kernels in the following manner:

(i) Gaussian kernels using 10 different bandwidth parameters from $\{2^{-3}, 2^{-2}, \ldots, 2^6\}$ by using all the variables and each single variable;

[4]http://appsrv.cse.cuhk.edu.hk/∼hqyang/doku.php?id=gmkl
[5]http://www.mosek.com/
[6]http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets
[7]http://www.fml.tuebingen.mpg.de/Members/raetsch/benchmark

Table 3.1: The performance evaluation (Mean Classification Accuracy (%) ± standard deviation) for different algorithms on the benchmark data sets. The number in the parenthesis shows the rank of each algorithm in terms of the mean classification accuracy.

	AveKernel	SimpleMKL	ℓ_2MKL	ℓ_pMKL	SGMKL	SM1MKL	SM2MKL
Heart	81.60±2.76 (7)	81.98±3.20 (5)	82.47±3.23 (1)	81.85±3.08 (6)	82.10±3.15 (3)	81.60±4.21 (4)	82.47±3.28 (1)
Diabetes	75.22±4.02 (7)	75.30±3.35 (6)	75.91±2.83 (4)	75.61±2.71 (5)	76.00±2.92 (3)	76.35±2.79 (1)	76.26±2.94 (2)
Australian	85.94±2.24 (2)	85.12±1.82 (4)	85.07±1.84 (6)	85.27±1.77 (3)	84.78±1.80 (7)	86.23±1.94 (1)	85.12±1.82 (4)
Ionosphere	90.29±4.01 (7)	91.81±1.92 (3)	91.71±2.70 (4)	91.33±2.64 (5)	91.90±2.39 (2)	91.33±2.82 (5)	92.10±2.38 (1)
Ringnorm	95.42±2.01 (7)	98.25±1.07 (1)	96.67±1.47 (6)	97.67±1.10 (4)	97.67±1.35 (4)	98.00±1.12 (2)	97.75±1.36 (3)
Banana	73.00±5.79 (7)	90.08±2.95 (1)	88.58±2.72 (6)	89.50±2.43 (4)	89.50±2.43 (4)	89.75±2.39 (3)	90.08±2.68 (1)
FlareSolar	67.04±3.45 (7)	67.59±3.99 (6)	68.64±2.96 (1)	68.59±2.93 (2)	67.94±3.32 (5)	68.59±2.93 (2)	68.59±2.93 (2)
Average Rank	6.28	3.71	4.00	4.14	4.00	2.57	2.00

Table 3.2: The training time evaluation (mean CPU time (Second) ± standard deviation) for different algorithms on the benchmark data sets. The number in the parenthesis shows the rank of each algorithm in terms of the mean CPU time.

	AveKernel	SimpleMKL	ℓ_2MKL	ℓ_pMKL	SGMKL	SM1MKL	SM2MKL
Heart	0.1938±0.039 (1)	19.32±11.71 (6)	9.023±1.957 (5)	4.273±2.533 (3)	129.9±92.55 (7)	1.928±3.517 (2)	8.245±4.498 (4)
Diabetes	1.075±0.5016 (1)	410.8±89.47 (6)	154.5±17.50 (4)	83.62±63.18 (3)	1206±694.8 (7)	39.41±22.79 (2)	240.6±88.93 (5)
Australian	1.134±0.144 (1)	194.6±25.60 (6)	159.6±76.61 (5)	82.94±63.39 (4)	897.3±535.5 (7)	23.58±18.77 (3)	22.96±3.542 (2)
Ionosphere	0.5656±0.2193 (1)	146.2±48.85 (6)	40.92±12.64 (5)	26.16±24.25 (3)	618.8±529.0 (7)	14.58±11.60 (2)	31.90±20.41 (4)
Ringnorm	0.606±0.292 (1)	297.3±10.4.7 (6)	69.18±40.48 (2)	277.5±288.4 (5)	1035±547.8 (7)	165.6±142.8 (3)	239.7±36.16 (4)
Banana	0.1734±0.1051 (1)	15.07±5.127(4)	15.71±1.656 (5)	49.74±37.16 (7)	16.85±6.792 (6)	11.56±5.480 (2)	15.23±17.55 (3)
FlareSolar	0.7609±0.2173(1)	2243±6244 (7)	152.5±24.54 (4)	382.7±414.4 (5)	649.5±253.7 (6)	81.03±129.8 (3)	58.69±38.95 (2)
Average Rank	1.00	5.86	4.28	4.28	6.71	2.43	3.43

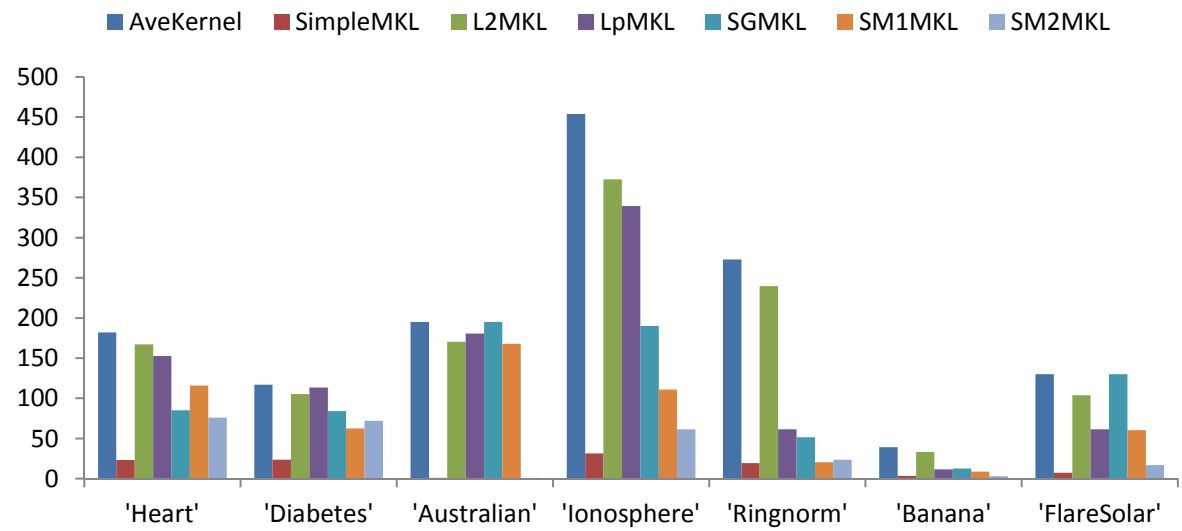


Figure 3.1: The average number of selected base kernels for each of the methods on the benchmark data set.

(ii) polynomial kernels with the degree from $\{1, 2, 3\}$ by using all the variables and each single variable.

We randomly partition the data set into two parts, namely 70% for training and the rest 30% for testing. For each partition, all the dimensions of samples in the training set are normalized to have zero mean and unit variance, while the samples in the test set are normalized accordingly. The experiments are then repeated for 10 times, and the mean accuracy and the standard deviation on the test set are reported for comparison.

3.5.2.2 Experimental results

Table 3.1 shows the performance comparison of different algorithms. We observe the effectiveness of our proposed MKL formulations when compared with the other MKL formulations. The average rank for each algorithm is calculated in the last row in Table 3.1. The average rank of SM2MKL is 2.00, and the average rank of SM1MKL is 2.57. So, SM1MKL and SM2MKL achieve similar performances. SimpleMKL follows SM1MKL and achieves the third position. In terms of the rank, SGMKL and ℓ_2MKL are a bit worse than SimpleMKL, and AveKernel is the worst. The results show that AveKernel and ℓ_pMKL cannot outperform ℓ_1MKL, probably because of redundant base kernels constructed in this setting. In terms of the loss functions defined on the kernel

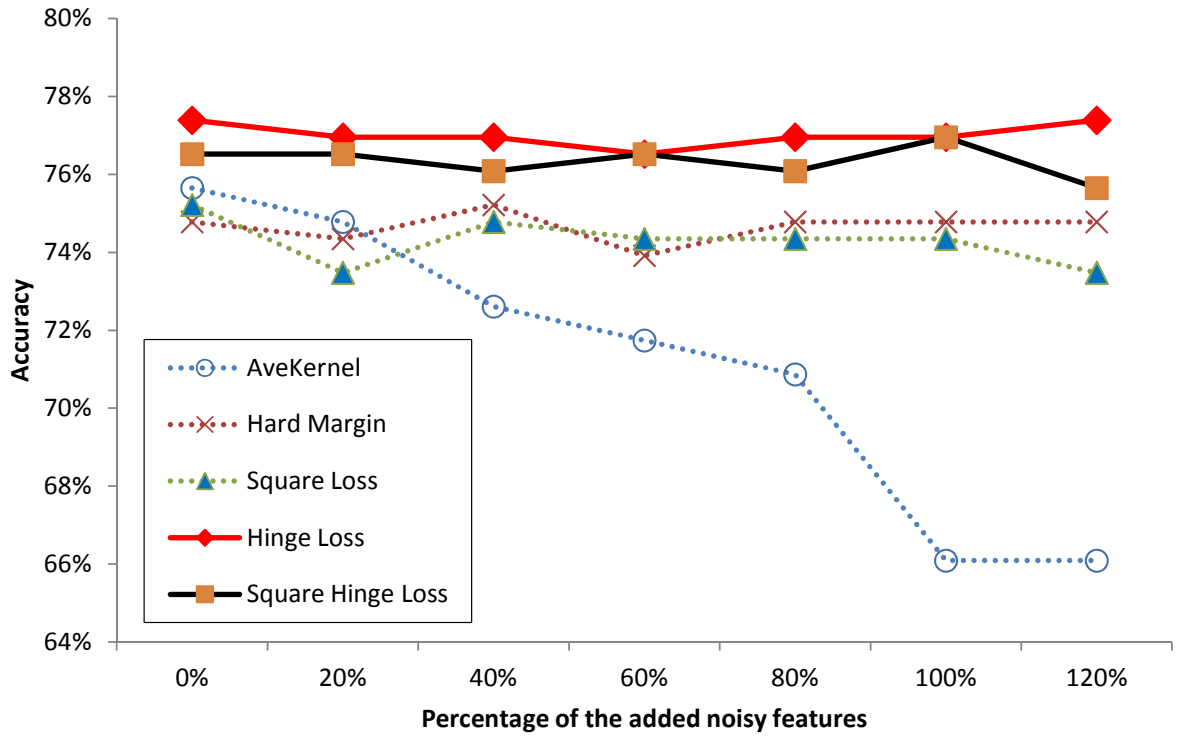


Figure 3.2: The performances of MKL when using different loss functions on kernel slack variables with respect to the level of noisy features for "Diabetes".

slack variables, the square loss is usually more sensitive to the outliers than (square) hinge loss, thus the generalization ability of ℓ_2MKL (ℓ_pMKL) may be limited when compared with the hinge loss soft margin MKL (SM1MKL) and square hinge loss soft margin MKL (SM2MKL).

The average numbers of selected base kernels for different MKL formulations are shown in Figure 3.1. We observe that SimpleMKL (ℓ_1MKL) selects the smallest number of base kernels on most of the data sets, and ℓ_2MKL selects almost all the base kernels, leading to dense solutions. The AveKernel selects all the base kernels. SGMKL, SM1MKL, and SM2MKL obtain sparser solutions when compared to ℓ_pMKL, which demonstrates that whether the solution is sparse or non-sparse should not be the main factor for the effectiveness of MKL methods.

Table 3.2 shows the mean CPU time costs for training each of the model on the training set. The average rank for each algorithm is also listed in the last row of the table. We can observe that generally AveKernel using the single average kernel is the fastest

Table 3.3: performance evaluation for different algorithms on the YouTube data set in terms of the mean Average Precision (MAP %), the mean number of selected kernels (MNK) and the mean training CPU time (MTT) over 11 concepts on the test set.

	AveKernel	SimpleMKL	ℓ_2MKL	ℓ_pMKL	SGMKL	SM1MKL	SM2MKL
MAP (%)	88.39	87.47	88.66	89.21	89.20	89.26	89.09
MNK	20	3.09	20	12.91	8.64	9.09	8.54
MTT (Second)	1.04	57.77	36.78	123.8	191.8	36.48	45.37

since SVM model is trained only once for prediction. For MKL algorithms, SGMKL and SimpleMKL are comparable to each other but they are less efficient when compared with other methods. When compared with the SimpleMKL and SGMKL, the ℓ_pMKL is more efficient due to the analytical solution for the kernel combination coefficients [222] [99]. For SM1MKL and SM2MKL, the training is very efficient thanks to the analytical updating rule for SM1MKL and the efficient simplex projection procedure for SM2MKL. Moreover, SM2MKL is much faster than SGMKL due to the utilization of the simplex projection method in our optimization process.

3.5.3 Measuring the impact of noisy base kernels for different MKL algorithms

From the soft margin point of view, we also analyze the characteristics for the MKL methods by using the regularization on the kernel slack variables. Specifically, some MKL formulations are more sensitive to noisy base kernels. To verify it, we compare AveKernel with other MKL methods using different loss functions on the kernel slack variables, including ℓ_1MKL (Hard Margin), ℓ_2MKL (Square Loss), SM1MKL (Hinge Loss) and SM2MKL (Square Hinge Loss). We use the first round of experiment for "Diabetes" from the benchmark data set to show the results of different algorithms when using noisy base kernels. The feature vector is augmented with $r*d$ dimensions of random generated features, where d is the dimension of the original feature vector and r is the percentage of the augmented noisy features in the range of $\{0, 0.2, 0.4, \cdots, 1.2\}$.

Figure 3.2 shows the accuracy of different MKL methods when using different levels of the noisy features for "Diabetes". We can clearly observe that AveKernel can achieve good results when the base kernels are clean. But when there are more noisy base kernels, the performance of AveKernel becomes much worse than the other algorithms. Moreover,

Table 3.4: performance evaluation for different algorithms on the Video Event data set in terms of the mean Average Precision (MAP %), the mean number of selected kernels (MNK) and the mean training CPU time (MTT) over 6 events on the test set.

	AveKernel	SimpleMKL	ℓ_2MKL	ℓ_pMKL	SGMKL	SM1MKL	SM2MKL
MAP (%)	44.33	47.14	53.34	53.49	53.81	54.84	53.98
MNK	80.00	3.63	68.00	60.50	60.53	53.83	61.77
MTT (Second)	2.297	542.9	261.1	396.4	1639	410.1	367.1

in this experiment, the hinge loss for the kernel slack variables is the most robust loss function when there are strong noisy base kernels.

3.5.4 Experiments on YouTube for Action Recognition

In computer vision applications, many features can be extracted for the image or video data sets, and the best results are usually obtained by fusing multiple types of features. However, some features may only be suitable for some specific applications and may even be harmful for other applications. Thus how to fuse or combine different features is an important problem for computer vision applications. In the following, we will show the effectiveness of MKL algorithms for Action Recognition [136].

3.5.4.1 Experimental setting

We evaluate different MKL algorithms on the YouTube data set, which contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The data set contains a total number of 1168 video sequences. We follow the pre-defined partitions as in [136], where the whole data set is partitioned to 25 folds. In order to compare the generalization ability of the different MKL formulations, we further choose 20 folds for training and use the remaining five folds for testing. The 20 training folds are also used to determine the parameters for all the algorithms.

Four types of features, namely Trajectory, HOG, HOF, MBH [198], are extracted from each of the video sequences. Then the base kernels are constructed from each of the four types of features by using the χ^2-kernel. The kernel mapping function is given as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma D(\mathbf{x}_i, \mathbf{x}_j)\big)$, where $D(\mathbf{x}_i, \mathbf{x}_j)$ is the χ^2 distance between any two videos

for each type of features, and $\gamma = \frac{1}{A}4^{n-1}$ with A being the mean value of the χ^2 distances between all the training samples. The kernel parameter n is from $\{-1, -0.5, \ldots, 1\}$, thus a total number of 20 base kernels are used in the experiment.

For the performance evaluation, we use the non-interpolated Average Precision (AP), which has been widely used as the performance metric for image and video retrieval applications. It corresponds to the multi-point average precision values of a precision-recall curve and incorporates the effect of recall. Mean AP (MAP) means the mean of APs over all the 11 semantic action concepts.

3.5.4.2 Experimental results

We report the MAP, the mean number of selected kernels (MNK) and the mean training CPU time (MTT) in Table 3.3 on this data set. The results are based on the mean of the 11 evaluated concepts. We can observe that the MAP of SimpleMKL is 87.47% and it is worse than AveKernel (88.39%), which indicates that ℓ_1MKL (SimpleMKL) may throw away some useful base kernels due to the hard margin property. We also observe that all the soft margin formulations ℓ_2MKL, ℓ_pMKL, SGMKL, SM1MKL and SM2MKL achieve better results when compared with AveKernel and ℓ_1MKL (SimpleMKL) and SM1MKL is the best in terms of MAP.

As shown in Table 3.3, we also observe that AveKernel and ℓ_2MKL select all the 20 base kernels, and ℓ_1MKL selects the smallest number of base kernels, (*i.e.*, 3.09 base kernels on average). SM1MKL, SM2MKL and SGMKL select fewer base kernels than AveKernel and ℓ_pMKL. Again, we conclude that whether the solution is sparse or non-sparse is not the key factor for the effectiveness of the MKL methods even though our new formulations can obtain sparser solutions compared with ℓ_pMKL.

We also find that the training time of AveKernel is much faster than other MKL algorithms, and SGMKL and SimpleMKL are slower when compared with other MKL algorithms like ℓ_2MKL, SM1MKL and SM2MKL, which have similar training time. ℓ_pMKL becomes slower in this experiment due to the smaller p value obtained from cross validation. Comparing the training time of SM2MKL and SGMKL, SM2MKL is much faster due to the efficient simplex projection method proposed under our soft margin framework. In general, our new SM1MKL outperforms other MKL learning algorithms in terms of both efficiency and effectiveness on this data set.

3.5.5 Experiments on Event6 for Video Event Recognition

3.5.5.1 Experimental setting

We evaluate different algorithms on another real world Event6 data set [58]. This data set contains 1101 videos, in which 924 videos are used as the training set and the remaining 177 are used as the test set. Six events (*i.e.*,"wedding", "birthday", "picnic", "parade", "show" and "sports") are used for performance evaluation. Two types of local features (*i.e.*,"STIP", "SIFT") are extracted from each of the video sequences, and then K-means is used to build the visual vocabularies for each of the local features. The spatial pyramid is also used to construct the final feature vector, in which two levels are used. Thus, four types of distances from two types of features and two pyramid levels are calculated as suggested in [58].

For a given distance $\mathbf{D}$, four types of kernels are used as the base kernels: Gaussian kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma D^2(\mathbf{x}_i, \mathbf{x}_j)\big)$), Laplacian kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\sqrt{\gamma} D(\mathbf{x}_i, \mathbf{x}_j)\big)$), inverse square distance (ISD) kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\gamma D^2(\mathbf{x}_i, \mathbf{x}_j)+1}$) and inverse distance (ID) kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{\gamma} D(\mathbf{x}_i, \mathbf{x}_j)+1}$), where $D(\mathbf{x}_i, \mathbf{x}_j)$ denotes the distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$. We set $\gamma = 4^{n-1}\gamma_0$, where $n \in \{-2, -1, \cdots, 2\}$ and $\gamma_0 = 1/A$ with A being the mean value of square distances between all the training samples, thus a total number of 80 base kernels are constructed from the four types of distances. Please refer to [58] for more details of the features and the kernels.

3.5.5.2 Experimental results

We report the MAP, the mean number of the selected base kernels (MNK) and the mean training CPU time (MTT) over all the 6 events. The MAP for AveKernel is only 44.33%, which is the worst on this data set. A possible explanation is the poor performance of the STIP features as shown in [58]. ℓ_1MKL (SimpleMKL) can improve the MAP to 47.14%, and ℓ_pMKL can further improve the performance to 53.49%. SM2MKL and SGMKL achieve comparable performances. However, our newly proposed SM1MKL achieves the best MAP 54.84%. We observe that AveKernel can be much worse when the base kernels are noisy. While SimpleMKL can discard the noisy base kernels, it may also discard some useful base kernels due to the hard margin property. Although ℓ_pMKL improves the performance, the generalized square loss is usually more sensitive to the outliers than

the (square) hinge loss, thus it can not achieve the best result. The hinge loss for the kernel slack variables should be the most robust one on this data set, thus SM1MKL achieves the best results when compared with other algorithms.

In terms of the MNK, AveKernel selects all the base kernels, and ℓ_2MKL still selects as more as possible base kernels, and ℓ_pMKL selects fewer base kernels due to a smaller value p determined from cross-validation. SGMKL, SM1MKL and SM2MKL can also select fewer base kernels when compared with AveKernel and ℓ_2MKL. As for the training time, our new algorithms are still very efficient. SM1MKL is faster than SGMKL and SimpleMKL and it is comparable to ℓ_pMKL. Again, we observe that the utilization of simplex projection method for SM2MKL significantly improves the efficiency, so SM2MKL is much faster than SGMKL, which again demonstrates it is beneficial to use our soft margin MKL framework to develop new efficient optimization method for improving the efficiency of square hinge loss soft margin MKL.

3.6 Summary

In this chapter, we have proposed a novel soft margin framework for Multiple Kernel Learning by introducing the kernel slack variables for kernel learning. Based on the formulation, we then propose the hinge loss soft margin MKL, the square hinge loss soft margin MKL and the square loss soft margin MKL. We additionally discover their connections with previous MKL methods and compare different MKL formulations in terms of the robustness of loss functions defined on the kernel slack variables. Comprehensive experiments have been conducted on the benchmark data set and the YouTube and Event6 data sets from computer vision applications. The experimental results demonstrate the effectiveness of our proposed framework.

In the future, we plan to analyze the theoretical bounds for the proposed soft margin MKLs and study their extensions to multi-class settings as well as investigate how to extend our MKL techniques for solving the more general ambiguity problem in [126, 218].

Chapter 4

Input-Output Kernel Learning for Learning with Ambiguity

Data ambiguities exist in many data mining and machine learning applications such as text categorization and image retrieval. For instance, it is generally beneficial to utilize the ambiguous unlabeled documents to learn a more robust classifier for text categorization under the semi-supervised learning setting. To handle general data ambiguities, we present a unified kernel learning framework named Input-Output Kernel Learning (IOKL). Based on our framework, we further propose a novel soft margin group sparse Multiple Kernel Learning (MKL) formulation by introducing a *group kernel slack variable* to each group of base input-output kernels. Moreover, an efficient block-wise coordinate descent algorithm with an analytical solution for the kernel combination coefficients is developed to solve the proposed formulation. We conduct comprehensive experiments on benchmark datasets for both semi-supervised learning and multiple instance learning tasks, and also apply our IOKL framework to a computer vision application called text-based image retrieval on the NUS-WIDE dataset. Promising results demonstrate the effectiveness of our proposed IOKL framework.

4.1 Introduction

The pioneering work for kernel learning was proposed by [115] to train the SVM classifier and learn the kernel matrix simultaneously, which is known as Multiple Kernel Learning (MKL). Since the objective function proposed in [115] has a simplex constraint for the

kernel coefficients, it is also known as ℓ_1MKL. While the developments of efficient algorithms for ℓ_1MKL have been a major research topic in the literature [7, 115, 164, 182, 221], recently [42] and [99] showed that ℓ_1MKL cannot achieve better prediction performance compared even with simple baselines for some real world applications. To address this problem, the non-sparse MKL [42, 99] was proposed.

The traditional MKL formulations are proposed for supervised classification problems, where the input base kernels and the labels of training samples are provided. The target is to learn a classifier as well as the optimal combination of input base kernels in a supervised manner. However, in many real-world applications, we often need to cope with data with uncertain labels or with an uncertain representation, which is uniformly referred to as *ambiguity* in this chapter. For instance, for text categorization under the semi-supervised learning setting [96], the unlabeled document with unknown labels may be helpful for learning a more robust classifier. Moreover, in text-based image retrieval [128], the training images collected from the photo-sharing websites (*e.g.*, Flickr.com or Photosig.com) are associated with loose labels. To tackle those data ambiguities, many learning strategies such as Semi-Supervised Learning (SSL) [96] and Multi-Instance Learning (MIL) [3] have been proposed.

Recently, MKL optimization techniques have been successfully applied to solve learning problems with ambiguity, such as bag-based MIL [130], instance-based MIL [128], SSL [131] and multi-view ambiguous learning [126]. In these works, their objective functions which are formulated in the form of a mixed integer programming (MIP) problem are relaxed into a reduced problem which shares a similar objective function as the ℓ_1MKL formulation. The empirical results in these works demonstrate the effectiveness of the MKL techniques for solving different learning problems with ambiguity. However, they assume that only one predefined input base kernel is provided beforehand, which may limit their generalization performance.

To address the ambiguity problem with multiple input base kernels, in this chapter, we formulate the general data ambiguities as a unified kernel learning problem. Specifically, by introducing the so-called *input-output kernels*, we propose a novel kernel learning framework, namely *Input-Output Kernel Learning (IOKL)*, which not only learns the optimal kernel but also handles data ambiguities. The major contributions of this chapter are summarized below:

(i) Unlike previous works for MKL in supervised learning settings without considering any uncertainty, our proposed IOKL framework simultaneously learns a robust classifier and the optimal kernel for the more challenging case in which there are data ambiguities either from unknown output labels or uncertainties associated with input data. Therefore, our kernel learning framework is applicable to more general learning scenarios such as multi-instance learning and semi-supervised learning.

(ii) To learn a more robust classifier, we propose a novel soft margin group sparse MKL formulation by introducing a new *group kernel slack variable* to each group of base input-output kernels. Moreover, a block-wise coordinate descent algorithm with an analytical solution for the kernel combination coefficients is developed to solve the new formulation efficiently.

(iii) We conduct comprehensive experiments on the benchmark datasets for both semi-supervised learning and multiple instance learning tasks, and also apply IOKL to a computer vision application (i.e., text-based image retrieval) on the challenging NUS-WIDE dataset. Promising results demonstrate the effectiveness of our proposed IOKL framework.

4.2 Learning with Ambiguity

4.2.1 Related works

In traditional **Multiple Kernel Learning** (MKL) methods [42, 57, 59, 99, 115, 225], the input base kernels and the labels for the training samples are given. Then the classifier is trained under a supervised learning setting where no uncertainty exists for either sample labels or the input data. However, in many real world applications, we often need to cope with data with uncertain labels or uncertainty associated with the input data. To this end, learning strategies such as the Semi-Supervised Learning [96] and Multi-Instance Learning [3] are designed to handle those data ambiguities.

Many **Semi-supervised Learning** (SSL) methods such as TSVM [96], LDS [32], LapSVM [12], LapRLS [12], LapREMR [36] and meanS3svm [131] have been proposed to utilize the unlabeled data for training the classifier. In addition, the **Multi-Instance**

Learning (MIL) methods including Non-SVM-based methods (*i.e.,* DD [144], EM-DD [238]) graph-based methods (*i.e.,* MIGraph [244], miGraph [244], HSR-MIL [120]), similarity-based method (*i.e.,* SMILE [212]) and SVM-based methods (*i.e.,* MI-SVM [3], mi-SVM [3], MI-Kernel [73], sMIL [23], MIL-CPB [128]) have been proposed recently.

In this chapter, we uniformly refer to such uncertainty in the data as ambiguity and divide it into two categories. The first type of uncertainty is due to the lack of label information, which is referred to as **output ambiguity**. For instance, in semi-supervised learning [96], the label information is not available for unlabeled training samples. Another type of uncertainty comes from the uncertainty associated with input data, such as the bag-based MIL [3, 73, 130], in which only the bag labels are given with the representative instance in each bag being unknown. Usually, an indicator variable is introduced for each instance to parameterize the bag representation of instances. We refer to this type of uncertainty as **input ambiguity**.

For clarity of presentation, we specify $\forall\, i$,$\forall\, m$ and $\forall\, t$ as meaning the value of i from 1 to n, the value of m from 1 to M, and the value of t from 1 to T, respectively.

4.2.2 Input-Output Kernel with Ambiguity

In this section, we define the *input-output kernel*, based on which we show several examples for utilizing MKL techniques to handle data ambiguities with only one predefined input base kernel. Then, in Section 4.2.3, we propose the Input-Output Kernel Learning (IOKL) framework by considering multiple input base kernels for handling general data ambiguities.

Suppose we are given a set of n input data $\{\mathbf{x}_i|_{i=1}^n\}$, and denote the possible output label vector as $\mathbf{y} = [\mathrm{y}_1, \ldots, \mathrm{y}_n]'$ with $\mathrm{y}_i \in \{+1, -1\}$, $\forall i$. We have the following definition:

Definition 4.7 *Given $\{\mathbf{x}_i|_{i=1}^n\}$ with $\mathbf{x}_i$ being the input data and the corresponding output label $\mathrm{y}_i \in \{+1, -1\}$, we define the **input-output kernel** as:*

$$\mathbf{K}^{IO} = \mathbf{K}^I \odot \mathbf{K}^O, \tag{4.1}$$

*where $\mathbf{K}^I \in \mathcal{R}^{n\times n}$ is the **input kernel** associated with the kernel function k, and $\mathbf{K}^{\mathrm{O}} = \mathbf{y}\mathbf{y}' \in \mathcal{R}^{n\times n}$ is the **output kernel** with $\mathbf{y} = [\mathrm{y}_1, \ldots, \mathrm{y}_n]'$.*

Example 1 (Output Ambiguity): This type of ambiguity comes from uncertain output labels, such as semi-supervised learning [96] and instance-based MIL [128]. The method in [128] formulates instance-based MIL as a MIP problem, and then further relaxes it as a ℓ_1MKL problem in the form of $\min_{\boldsymbol{\mu}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \left(-\frac{1}{2}\boldsymbol{\alpha}' \left(\sum_{t:\mathbf{y}^t\in\mathcal{Y}} \mu_t \mathbf{K} \odot (\mathbf{y}^t\mathbf{y}^{t'})\right)\boldsymbol{\alpha}\right)$, where $\mathcal{Y}$ is the feasible set of the instance label vector $\mathbf{y}$ and $\mathbf{y}^t$ is the t^{th} candidate label vector under the MIL constraints [3, 128], $\boldsymbol{\mu} \in \mathcal{R}^{|\mathcal{Y}|}$ is a coefficient vector, and $\boldsymbol{\alpha} \in \mathcal{R}^n$ is the SVM dual vector. This relaxed problem can be deemed as optimizing the linear combination of $|\mathcal{Y}|$ base input-output kernels $\mathbf{K}_t^{IO}$ constructed from Definition 4.7, namely, $\mathbf{K}^I = \mathbf{K} \in \mathcal{R}^{n\times n}$ with $\mathbf{K}(i,j) = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K}_t^O = \mathbf{y}^t\mathbf{y}^{t'}$. The $|\mathcal{Y}|$ base input-output kernels are obtained due to output ambiguity. The mapping function for $\mathbf{K}_t^{IO}$ is $\tilde{\varphi}_t(\mathbf{x}_i) = \mathrm{y}_i^t\varphi(\mathbf{x}_i)$ with $\varphi(\cdot)$ being the mapping function for k. The semi-supervised learning shares the same form of objective function except that the feasible set $\mathcal{Y}$ is based on the balance constraint as in [96].

Example 2 (Input Ambiguity): For bag-based MIL, the input data is composed of n bags, and each bag $\mathbf{x}_i$ consists of n_i instances $\{\mathbf{x}_i^j|_{j=1}^{n_i}\}$ with known bag label y_i but unknown bag representation w.r.t. the instances inside each bag. With $N = \sum_{i=1}^n n_i$, the kernel $\mathbf{K} \in \mathcal{R}^{N\times N}$ associated with a kernel function k w.r.t. the instances is given. The method in [130] formulates this problem as a mixed integer programming problem, and also relaxes the problem into a ℓ_1MKL problem in form of

$$\min_{\boldsymbol{\mu}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} -\frac{1}{2}\boldsymbol{\alpha}' \left(\sum_{t:\boldsymbol{\delta}^t\in\Delta} \mu_t(\mathbf{y}\mathbf{y}') \odot \mathrm{conv}\left(\mathbf{K} \odot (\boldsymbol{\delta}^t\boldsymbol{\delta}^{t'})\right)\right)\boldsymbol{\alpha},$$

where $conv(\cdot)$ is the convolution operator [73] for mapping the kernel matrix from instance level to bag level, $\boldsymbol{\delta} \in \mathcal{R}^N$ is an indicator vector with its element $\delta_i^j \in \{0, 1\}$ which is associated with $\mathbf{x}_i^j$ (i.e., $\delta_i^j = 1$ if $\mathbf{x}_i^j$ is used to represent the i^{th} bag), and Δ is the feasible set for $\boldsymbol{\delta}$ under bag-based MIL constraints [73, 130]. Then we can have $|\Delta|$ input-output kernels $\mathbf{K}_t^{IO}$ with $\mathbf{K}_t^I = \mathrm{conv}\left(\mathbf{K} \odot (\boldsymbol{\delta}^t\boldsymbol{\delta}^{t'})\right)$ and $\mathbf{K}^O = \mathbf{y}\mathbf{y}'$. The mapping function for each input-output kernel is $\tilde{\varphi}_t(\mathbf{x}_i) = \mathrm{y}_i \sum_{j=1}^{n_j} \delta_i^{jt}\varphi(\mathbf{x}_i^j), \forall t$ with $\varphi(\cdot)$ being the mapping function for k.

In this chapter, we uniformly model the general data ambiguities (i.e., output ambiguity and input ambiguity) by using a vector $\mathbf{h}$ referred to as an *ambiguity candidate*.

Specifically, for output ambiguity, we have $\mathbf{h} = \mathbf{y}$, and for input ambiguity, we have $\mathbf{h} = \boldsymbol{\delta}$. Note that for any predefined k, each ambiguity candidate leads to one input-output kernel, thus the total number T of base input-output kernels is determined by the size of $\mathcal{Y}$ or Δ, i.e., $T = |\mathcal{Y}|$ or $T = |\Delta|$. In the following, we refer to $\mathcal{C} = \{\mathbf{h}^1, \ldots, \mathbf{h}^T\}$ as the ambiguity candidate set which contains all possible ambiguity candidates, and propose the new Input-Output Kernel Learning framework for handling the general data ambiguities with multiple input base kernels.

4.2.3 Input-Output Kernel Learning (IOKL)

Considering $\mathcal{C}$ as that in Section 4.2.2 and M input base kernels $\mathcal{K} = \{k_1, ..., k_M\}$ as that in the traditional MKL framework, we can construct a total number of $M \times T$ base input-output kernels. Let us denote the input-output kernel from k_m and $\mathbf{h}^t$ as $\mathbf{K}^{IO}_{m,t}$, $m = 1, \ldots, M$ and $t = 1, \ldots, T$. The mapping function $\tilde{\varphi}_{m,t}(\cdot)$ for $\mathbf{K}^{IO}_{m,t}$ can be obtained by instantiating the $\varphi(\cdot)$ in Example 1 and 2 with $\varphi_m(\cdot)$. Inspired by the traditional MKL framework, we propose to learn the target classifier[1] $f(\mathbf{x}_i) = \sum_{t=1}^{T}\sum_{m=1}^{M} \tilde{\mathbf{w}}'_{m,t}\tilde{\varphi}_{m,t}(\mathbf{x}_i)$ with a linear combination of those input-output kernels. We then formulate the *Input-Output Kernel Learning (IOKL)* problem as the following kernel learning problem:

$$\begin{aligned} \min_{\mathbf{D}\in\mathcal{M},\tilde{\mathbf{w}}_{m,t},\rho,\xi_i} \quad & \frac{1}{2}\left(\sum_{t=1}^{T}\sum_{m=1}^{M}\frac{\|\tilde{\mathbf{w}}_{m,t}\|^2}{d_{m,t}} + C\sum_{i=1}^{n}\xi_i^2\right) - \rho \\ \text{s.t.} \quad & \sum_{t=1}^{T}\sum_{m=1}^{M}\tilde{\mathbf{w}}'_{m,t}\tilde{\varphi}_{m,t}(\mathbf{x}_i) \geq \rho - \xi_i, \forall\, i, \end{aligned} \tag{4.2}$$

where $\mathbf{D} \in \mathcal{R}^{M\times T}$ is the input-output kernel coefficient matrix with $\mathbf{D}(m,t) = d_{m,t}$ for $m = 1, \ldots, M, t = 1, \ldots, T$, and $\mathcal{M} = \{\mathbf{D}|\Omega(\mathbf{D}) \leq 1, \mathbf{D} \geq \mathbf{0}\}$ is the feasible set for input-output kernel coefficient matrix with $\Omega(\mathbf{D})$ being the general regularization term for $\mathbf{D}$.

Note that our IOKL is based on the input-output kernels by considering the general data ambiguities, while the existing MKL methods [186], [99] only learn the optimal kernel under the supervised setting. We take ν-SVM [170] with square hinge loss as an example in this chapter, but other SVM formulations can be incorporated similarly.

[1]The bias b can be incorporated by augmenting 1 as the additional feature in $\varphi_m(\cdot), \forall m$.

4.3 Soft Margin Group Sparse Regularization for IOKL

4.3.1 Regularization for IOKL

For traditional supervised MKL, the regularization for the kernel coefficients can be ℓ_1-norm [115, 164], ℓ_2-norm [42] and ℓ_p-norm [99]. In addition, Composite Kernel Learning (CKL) [186] proposed a generic $\ell_{p,q}$-norm for the input base kernels. However, all these regularization terms are for input base kernels without considering the ambiguity.

In general, any regularization from previous works can be readily adopted for our IOKL framework. Considering the ambiguity problem, we have two intuitions:

(i) **Non-sparse regularization for input base kernels**: The input base kernels are possibly based on different features from professional knowledge (*e.g.*, feature design in computer vision applications). Thus, complementary and orthogonal information [42, 99] from input base kernels should be preserved.

(ii) **Sparse regularization for ambiguity candidates**: The underlying authentic ambiguity candidate $\mathbf{h}$ only has few correct choices (*e.g.*, the authentic labels of unlabeled samples for semi-supervised learning only have one correct choice according to the ground-truth labels). Thus, the ambiguity candidates should be enforced to be sparse.

To preserve the non-sparseness for input base kernels and also enforce sparseness for ambiguity candidates, we employ the group sparse $\ell_{2,1}$-norm regularization [235] for our IOKL in (4.2) as:

$$\Omega(\mathbf{D}) = \sum_{t=1}^{T} \sqrt{\sum_{m=1}^{M} d_{m,t}^2}\,, \tag{4.3}$$

where the ℓ_2-norm is used for input base kernels and ℓ_1-norm is used for ambiguity candidates.

Note that, different from [186] that proposed a generic group structure to input base kernels for MKL under the traditional supervised learning setting, our $\ell_{2,1}$-norm structure is specifically enforced on the base input-output kernels by considering the general data ambiguities, thus our work significantly differs from [186]. Although the generic group structure for input base kernels from [186] is more general and can be incorporated into

our IOKL, we only utilize the non-sparse ℓ_2-norm [99] for the input base kernels due to the aforementioned intuitions. We will also validate these intuitions for designing the regularization term on the real world computer vision data set in Section 4.5.1.

4.3.2 A Hard Margin Perspective for Group Sparse MKL

Substituting the group sparse $\ell_{2,1}$-norm in (4.3) back into (4.2), we can get the primal form of the group sparse MKL. To further discover the properties of this group sparse MKL, we go a step further to derive its dual form, from which we give a novel "hard margin" interpretation for MKL. The dual form of (4.2) with regularization in (4.3) can be obtained in the following proposition:

Proposition 8 *The dual form of the MKL problem in (4.2) with* $\Omega(\mathbf{D})$ *defined in (4.3) is:*

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\lambda},\gamma} \quad & -\frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2 - \gamma \\ s.t. \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}_{m,t}^{IO}\boldsymbol{\alpha} \leq \lambda_{m,t}, \forall\, m, \forall\, t, \\ & \boldsymbol{\alpha} \geq 0, \mathbf{1}'\boldsymbol{\alpha} = 1, \\ & \sqrt{\sum_{m=1}^{M}\lambda_{m,t}^2} = \gamma, t = 1, \ldots, T, \end{aligned} \tag{4.4}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]'$, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}'_{\cdot,1}, \ldots, \boldsymbol{\lambda}'_{\cdot,T}]'$ *(with* $\boldsymbol{\lambda}_{\cdot,t} = [\lambda_{1,t}, \ldots, \lambda_{M,t}]', \forall\, t$*) and* γ *are the Lagrangian multipliers.*

Proof: We firstly rewrite the problem in (4.2) as:

$$\begin{aligned} \min_{\mathbf{D}\geq 0, \mathbf{z}, \tilde{\mathbf{w}}_{m,t}, \rho, \xi_i} \quad & \frac{1}{2}\left(\sum_{m,t}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}} + C\sum_{i=1}^{n}\xi_i^2\right) - \rho \\ \text{s.t.} \quad & \sum_{t,m}\tilde{\mathbf{w}}'_{m,t}\tilde{\varphi}_{m,t}(\mathbf{x}_i) \geq \rho - \xi_i, \forall\, i, \\ & d_{m,t} = z_{m,t}, \forall\, t, \forall\, m, \\ & \sum_{t=1}^{T}\sqrt{\sum_{m=1}^{M} z_{m,t}^2} \leq 1, \end{aligned} \tag{4.5}$$

where $z_{m,t}$ is an intermediate variable introduced for ease of derivation. Then the Lagrangian can be written as: $\mathcal{L} = \frac{1}{2}\left(\sum_{m,t}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}} + C\sum_{i=1}^{n}\xi_i^2\right) - \rho + \gamma\left(\sum_t\sqrt{\sum_m z_{m,t}^2} - 1\right) + \sum_{m,t}\lambda_{m,t}(d_{m,t} - z_{m,t}) - \sum_{i=1}^{n}\alpha_i\left(\sum_{m,t}\tilde{\mathbf{w}}'_{m,t}\tilde{\varphi}_{m,t}(\mathbf{x}_i) - \rho + \xi_i\right) - \sum_{m,t}d_{m,t}\eta_{m,t}$, where $\gamma \geq 0, \alpha_i \geq 0, \eta_{m,t} \geq 0$ and $\lambda_{m,t}$ are the Lagrangian multipliers introduced from the constraints in (4.5).

By setting the derivatives of $\mathcal{L}$ with respect to the primal variables $\tilde{\mathbf{w}}_{m,t}, \rho, \xi_i, d_{m,t}, z_{m,t}$ to be zeros, we have $\frac{\tilde{\mathbf{w}}_{m,t}}{d_{m,t}} = \sum_{i=1}^{n}\alpha_i\tilde{\varphi}_{m,t}(\mathbf{x}_i)$,$\sum_{i=1}^{n}\alpha_i = 1$, $C\xi_i = \alpha_i$, $-\frac{1}{2}\frac{\|\tilde{\mathbf{w}}_{m,t}\|^2}{d_{m,t}^2} - \eta_{m,t} + \lambda_{m,t} = 0$ and

$$\lambda_{m,t} = \gamma\frac{z_{m,t}}{\sqrt{\sum_{l=1}^{M}{z_{l,t}}^2}} \tag{4.6}$$

The equation in (4.6) gives

$$\sqrt{\sum_{m=1}^{M}\lambda_{m,t}^2} = \gamma\frac{\sqrt{\sum_{m=1}^{M}z_{m,t}^2}}{\sqrt{\sum_{l=1}^{M}{z_{l,t}}^2}} = \gamma,\ \forall t, \tag{4.7}$$

which are the equality constraints as in the last row of (4.4). The other constraints can be obtained similarly. Eliminating the primal variables in the Lagrangian gives the objective form as in (4.4). Thus we finish the proof. In the dual form, the group sparse regularization term reflects that the upper bounds $\lambda_{m,t}$'s of the quadratic terms are grouped into T groups accordingly. The constraint for each group of upper bounds is formulated as $||\boldsymbol{\lambda}_{\cdot,t}||_2 = \gamma$, which encodes the non-sparseness from the input base kernels inside the t^{th} group. However, we observe that the ℓ_2-norm $||\boldsymbol{\lambda}_{\cdot,t}||_2$ strictly equals the global *"margin"* γ, thus there is no "error" allowed from t^{th} group for the learning problem, which can be deemed as a *"hard margin"* property for each group of the input-output kernels.

4.3.3 Soft Margin Group Sparse MKL

To overcome the "hard margin" defect, in this section we propose a novel soft margin formulation to learn a classifier with better generalization ability. Specifically, we can introduce one slack variable ζ_t, namely a *group kernel slack variable*, to the t^{th} group for

$\forall t$, then the *soft margin group sparse MKL* can be formulated as:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\lambda},\gamma,\boldsymbol{\zeta}} \quad & \frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2+\gamma+\theta\sum_{t=1}^{T}\zeta_t \\ \text{s.t.} \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}_{m,t}^{IO}\boldsymbol{\alpha}\le\lambda_{m,t},\forall m,\forall t, \\ & \boldsymbol{\alpha}\ge 0,\mathbf{1}'\boldsymbol{\alpha}=1, \\ & \sqrt{\sum_{m=1}^{M}\lambda_{m,t}^2}=\gamma+\zeta_t,\zeta_t\ge 0,\forall\, t,\end{aligned} \tag{4.8}$$

where $\boldsymbol{\zeta}=[\zeta_1,\ldots,\zeta_T]'$ and θ is the soft margin regularization parameter for group kernel slack variables.

To efficiently solve this new objective function for MKL, we have the following proposition:

Proposition 9 *The primal form of the soft margin group sparse MKL problem as in (4.8) is shown as the following optimization problem:*

$$\begin{aligned}\min_{\mathbf{D}\ge\mathbf{0},\tilde{\mathbf{w}}_{t,m},\rho,\xi_i} \quad & \frac{1}{2}\left(\sum_{\forall m,\forall t}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}}+C\sum_{i=1}^{n}\xi_i^2\right)-\rho \\ s.t. \quad & \sum_{\forall m,\forall t}\tilde{\mathbf{w}}_{m,t}'\tilde{\varphi}_{m,t}(\mathbf{x}_i)\ge\rho-\xi_i,\forall i, \\ & \sum_{t=1}^{T}\sqrt{\sum_{m=1}^{M}d_{m,t}^2}\le 1, \\ & \sqrt{\sum_{m=1}^{M}d_{m,t}^2}\le\theta, t=1,\ldots,T.\end{aligned} \tag{4.9}$$

Proof: In order to get the dual form of (4.9), we rewrite the problem in (4.9) as:

$$\begin{aligned}\min_{\mathbf{D},\mathbf{z},\mathbf{e},\tilde{\mathbf{w}}_{m,t},b,\rho,\xi_i} \quad & \frac{1}{2}\left(\sum_{m,t}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}}+C\sum_{i=1}^{n}\xi_i^2\right)-\rho \\ \text{s.t.} \quad & \sum_{m,t}\tilde{\mathbf{w}}_{m,t}'\tilde{\varphi}_{m,t}(\mathbf{x}_i)\ge\rho-\xi_i,\forall i, \\ & d_{m,t}=z_{m,t},\quad d_{m,t}\ge 0,\quad \forall\, t,m, \\ & e_t=\sqrt{\sum_{m}z_{m,t}^2},\quad e_t\le\theta,\quad \forall\, t, \\ & \sum_{t}e_t\le 1,\end{aligned} \tag{4.10}$$

where $z_{m,t}$, e_t are the intermediate variables that would be beneficial for deriving the dual. The problem in (4.10) and the problem in (4.9) are equivalent by eliminating the intermediate variables $z_{m,t}$, e_t in (4.10).

Then the Lagrangian of (4.10) can be written as:

$$\begin{aligned}\mathcal{L}=&\frac{1}{2}\left(\sum_{m,t}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}}+C\sum_{i=1}^{n}\xi_i^2\right)-\rho+\sum_t\zeta_t\left(e_t-\theta\right)\\&-\sum_{i=1}^{n}\alpha_i\left(\sum_{m,t}\tilde{\mathbf{w}}'_{m,t}\tilde{\varphi}_{m,t}(\mathbf{x}_i)-\rho+\xi_i\right)\\&+\sum_{m,t}\lambda_{m,t}\left(d_{m,t}-z_{m,t}\right)-\sum_{m,t}d_{m,t}\eta_{m,t}\\&+\gamma\left(\sum_t e_t-1\right)-\sum_t\beta_t\left(e_t-\sqrt{\sum_m z_{m,t}^2}\right),\end{aligned}$$

where $\alpha_i \geq 0, \eta_{m,t} \geq 0, \gamma \geq 0, \zeta_t \geq 0, \beta_t$ and $\lambda_{m,t}$ are the Lagrangian multipliers introduced from the constraints in (4.10).

By setting the derivatives of $\mathcal{L}$ with respect to the primal variables $\tilde{\mathbf{w}}_{m,t}, \rho, \xi_i, d_{m,t}, z_{m,t}, e_t$ to be zeros, we can get the KKT conditions similarly with the proof for Proposition 8. Using similar elimination techniques gives exactly the dual form as in (4.8). Thus we get the conclusion.

Note that by introducing the group kernel slack variables in the dual of the group sparse MKL in (4.4), we observe that this corresponds to having one more box constraint for the ℓ_2-norm of each group of coefficients, specifically $\sqrt{\sum_{m=1}^{M} d_{m,t}^2} \leq \theta$, $\forall t$. The new regularization parameter θ for group kernel slack variables places an upper bound on the ℓ_2-norm of the coefficients from each group, thus preventing strong values from any groups of base input-output kernels.

This kind of improvement is in analogous to the change from the hard margin SVM [19] to hinge loss soft margin SVM [45]. The soft margin SVM introduces one slack variable for each training instance, while our proposed soft margin group sparse MKL introduces one slack variable for each group of base input-output kernels. If $\theta \geq 1$, the soft margin case in (4.8) reduces to the hard margin case in (4.4). To distinguish the two types of IOKL, we refer to (4.2) with regularization in (4.3) and (4.9) as IOKL-HM and IOKL-SM, respectively.

Algorithm 3 : Cutting-plane algorithm for IOKL

1: Initialize $\mathbf{h}^1$,$\tau = 1$, and set $\mathcal{C}_m = \mathbf{h}^1$, $m = 1 \dots M$.
2: Get $\mathbf{K}_{m,t}^{IO}$ by using $\mathcal{C}_m$ and $\mathcal{K}$ according to Definition 4.7.
3: Get $\boldsymbol{\alpha}^\tau$ by solving the MKL problem as in (4.8).
4: For $m = 1 \dots M$
   Get $\mathbf{h}^{\tau+1} = \arg\max_{\mathbf{y}\in\mathcal{Y}} (\boldsymbol{\alpha}^\tau)' (\mathbf{K}_m \odot (\mathbf{y}\mathbf{y}')) (\boldsymbol{\alpha}^\tau)$.
   Set $\mathcal{C}_m = \mathbf{h}^{\tau+1} \bigcup \mathcal{C}_m$.
5: End For
6: $\tau = \tau + 1$.
7: Repeat Steps 2 to 6 until convergence.

4.3.4 Cutting-plane Algorithm for IOKL

From Definition 4.7, we observe that the number of all possible candidates for the ambiguity candidate $\mathbf{h}$ could be exponential with the size of $\mathbf{h}$, which makes it inefficient to train a classifier with MKL. Fortunately, we can employ the cutting-plane algorithm to iteratively select a small number of the most violated input-output kernels instead of using all of them. Taking semi-supervised learning as an example, the detailed cutting-plane algorithm is listed in Algorithm 3.

Specifically, taking semi-supervised learning as an example, according to the quadratic constraints in (4.8), the most violated input-output kernels can be constructed iteratively with $\mathbf{h}^{\tau+1}$ obtained by solving the following problem:

$$\mathbf{h}^{\tau+1} = \arg\max_{\mathbf{y}\in\mathcal{Y}} (\boldsymbol{\alpha}^\tau)' (\mathbf{K}_m \odot (\mathbf{y}\mathbf{y}')) (\boldsymbol{\alpha}^\tau), \forall m, \tag{4.11}$$

which can be optimized by either the enumeration method [128] or the approximation based sorting algorithm [130], [125].

By using the cutting-plane algorithm, the ambiguity candidate is added into $\mathcal{C}_m$ iteratively. Thus, the whole solution to IOKL in Algorithm 3 depends on solving the inner MKL problem as in (4.9) efficiently, which will be detailed in Section 4.4. In the sequel, we still denote the size of $\mathcal{C}_m$ inside each iteration as T.

4.4 Solution to Soft Margin Group Sparse MKL

The formulation in (4.9) is a convex optimization problem, therefore the global solution for (4.9) is guaranteed. To solve this problem, we follow the block-wise coordinate descent

procedure for ℓ_p-norm MKL [99, 222] and CKL [186], and optimize two subproblems w.r.t. the two sets of variables $\{\tilde{\mathbf{w}}_{m,t}, \rho, \xi_i\}$ and $\{\mathbf{D}\}$ alternately. Note that, due to the additional box constraints introduced from soft margin regularization for the group input-output kernels, the subproblem for updating $\mathbf{D}$ becomes much more difficult than the one in [99, 186, 222].

4.4.1 Updating SVM Variables with Fixed D

With a fixed $\mathbf{D}$, we write the dual of (4.9) by introducing the non-negative Lagrangian multipliers α_i's as:

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \; -\frac{\boldsymbol{\alpha}'\boldsymbol{\alpha}}{2C} - \frac{1}{2}\boldsymbol{\alpha}'\left(\sum_{\forall m,\forall t} d_{m,t}\mathbf{K}^{IO}_{m,t}\right)\boldsymbol{\alpha}. \tag{4.12}$$

which is a quadratic programming (QP) problem with $\mathcal{A} = \{\boldsymbol{\alpha}|\boldsymbol{\alpha}'\mathbf{1} = 1, \mathbf{0} \leq \boldsymbol{\alpha}\}$, and can be efficiently solved by any existing QP solvers. Then, the primal variables $\tilde{\mathbf{w}}_{m,t}, \rho, \xi_i$ can be recovered accordingly. For instance, the norm for $\tilde{\mathbf{w}}_{m,t}$ can be expressed as:

$$||\tilde{\mathbf{w}}_{m,t}|| = d_{m,t}\sqrt{\boldsymbol{\alpha}'\mathbf{K}^{IO}_{m,t}\boldsymbol{\alpha}}. \tag{4.13}$$

4.4.2 Updating D with Fixed SVM Variables

For updating $\mathbf{D}$ with fixed SVM variables, the subproblem can be equivalently formulated as:

$$\begin{aligned}\min_{\mathbf{D}\geq\mathbf{0},\mathbf{e}} \; & \frac{1}{2}\sum_{m,t}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}} \\ & e_t = \sqrt{\sum_{m=1}^{M} d_{m,t}^2}, \quad e_t \leq \theta, \quad \forall\, t, \\ & \sum_t e_t \leq 1,\end{aligned} \tag{4.14}$$

where $\mathbf{e} = [e_1, \ldots, e_T]'$ is an intermediate variable vector introduced for ease of optimization.

Because of the additional upper bound θ, the existing optimization techniques [99, 186, 222] cannot be directly utilized. Inspired by [172] for simplex projection, we introduce

Algorithm 4 : Optimization procedure for solving ω

Calculate $a_t = \left(\sum_{l=1}^{M} ||\tilde{\mathbf{w}}_{l,t}||^{4/3}\right)^{3/4}, \forall\, t.$
Sort a_t's such that $a_1 \geq a_2 \geq \cdots \geq a_T$.
$\omega = 0$.
while $\omega < T$ **do**
  if $\frac{(1-\omega\theta)a_{\omega+1}}{\sum_{s=\omega+1}^{T} a_s} < \theta$
    break;
  else
    $\omega = \omega + 1$.
  end
end while

a Lagrangian method to solve (4.14) analytically. Before we introduce our algorithm to solve (4.14), let us denote ω as the number of elements whose value strictly equals θ in the optimal solution for $\mathbf{e}$ in (4.14), and the closed-form solution for (4.14) is obtained as in the following:

Proposition 10 *The optimal solution for subproblem (4.14) is given as, for* $t = 1, \ldots, \omega$,

$$d_{m,t} = \theta \frac{\|\tilde{\mathbf{w}}_{m,t}\|^{2/3}}{\sqrt{\sum_{l=1}^{M} ||\tilde{\mathbf{w}}_{l,t}||^{4/3}}}, \forall m, \tag{4.15}$$

and for $t = \omega + 1, \ldots, T$,

$$d_{m,t} = (1 - \omega\theta) \frac{||\tilde{\mathbf{w}}_{m,t}||^{2/3} \left(\sum_{l=1}^{M} ||\tilde{\mathbf{w}}_{l,t}||^{4/3}\right)^{1/4}}{\sum_{t=\omega+1}^{T} \left(\sum_{l=1}^{M} ||\tilde{\mathbf{w}}_{l,t}||^{4/3}\right)^{3/4}}, \forall m. \tag{4.16}$$

Proof: The Lagrangian for (4.14) is:

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} \sum_{m,t} \frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}} - \sum_{m,t} d_{m,t}\eta_{m,t} + \sum_{t} \zeta_t \left(e_t - \theta\right) \\ & + \gamma \left(\sum_{t} e_t - 1\right) - \sum_{t} \beta_t \left(e_t - \sqrt{\sum_{m} d_{m,t}^2}\right), \end{aligned} \tag{4.17}$$

where $\eta_{m,t} \geq 0, \gamma \geq 0, \zeta_t \geq 0$ and β_t are Lagrangian multipliers introduced for the constraints.

By setting the derivatives of the Lagrangian as in (4.17) with respect to the primal variables $d_{m,t}, e_t$ to be zeros, we have the following equations:

$$-\frac{1}{2}\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{d_{m,t}^2} + \beta_t \frac{d_{m,t}}{\sqrt{\sum_l d_{l,t}^2}} = \eta_{m,t}, \tag{4.18}$$

$$\gamma + \zeta_t - \beta_t = 0, \tag{4.19}$$

and the complementary KKT conditions give $\eta_{m,t} d_{m,t} = 0$, $\beta_t \left(e_t - \sqrt{\sum_m d_{m,t}^2}\right) = 0$ and

$$\zeta_t \left(e_t - \theta\right) = 0. \tag{4.20}$$

Since $\|\tilde{\mathbf{w}}_{m,t}\|^2 > 0$, thus $d_{m,t} > 0$, according to the previous KKT conditions, $\eta_{m,t} = 0$. According to (4.18) with $\eta_{m,t} = 0$, we have $\frac{1}{2}\frac{\|\tilde{\mathbf{w}}_{m,t}\|^2}{d_{m,t}^2} = \beta_t \frac{d_{m,t}}{\sqrt{\sum_l d_{l,t}^2}}$, which further gives $\frac{||\tilde{\mathbf{w}}_{m,t}||^2}{||\tilde{\mathbf{w}}_{l,t}||^2} = \frac{d_{m,t}^3}{d_{l,t}^3}$ for $\forall m, l$, then $e_t = \sqrt{\sum_l d_{l,t}^2} = d_{m,t}\sqrt{\sum_l \frac{d_{l,t}^2}{d_{m,t}^2}} = d_{m,t}\sqrt{\sum_l \frac{||\tilde{\mathbf{w}}_{l,t}||^{4/3}}{||\tilde{\mathbf{w}}_{m,t}||^{4/3}}}$, thus

$$e_t = \frac{d_{m,t}}{||\tilde{\mathbf{w}}_{m,t}||^{2/3}} \sqrt{\sum_l ||\tilde{\mathbf{w}}_{l,t}||^{4/3}}. \tag{4.21}$$

In the following, we will discuss the solutions for $d_{m,t} > 0$ based on the value of e_t.

If $e_t = \theta$ for any group t: Due to $e_t = \theta$ and (4.21), the solution for $d_{m,t}$ is obtained as that in (4.15).

If $e_t < \theta$ for any group t: We can observe from (4.20) that $\zeta_t = 0$, and this further gives $\gamma = \beta_t$ according to (4.19). With (4.18) and $\eta_{m,t} = 0$ as well as (4.21), we further have $d_{m,t}^3 = \frac{\sqrt{\sum_l d_{m,t}^2}}{2\beta_t}||\tilde{\mathbf{w}}_{m,t}||^2 = \frac{e_t}{2\gamma}||\tilde{\mathbf{w}}_{m,t}||^2 = \frac{d_{m,t}\sqrt{\sum_l ||\tilde{\mathbf{w}}_{l,t}||^{4/3}}}{2\gamma}||\tilde{\mathbf{w}}_{m,t}||^{4/3}$, thus we have,

$$d_{m,t} = \frac{||\tilde{\mathbf{w}}_{m,t}||^{2/3} \left(\sum_{l=1}^M ||\tilde{\mathbf{w}}_{l,t}||^{4/3}\right)^{1/4}}{\sqrt{2\gamma}}, \tag{4.22}$$

and then $e_t = \frac{1}{\sqrt{2\gamma}} \left(\sum_{l=1}^M ||\tilde{\mathbf{w}}_{l,t}||^{4/3}\right)^{3/4}$ by substituting (4.22) back into (4.21).

The formulations in (4.15) and (4.22) show that if we know γ and whether e_t equals to θ, the optimal solution for $\mathbf{D}$ can be obtained accordingly. Thus the remaining key problem is to get γ and to determine whether e_t equals to θ.

Suppose that ω, the number of elements whose value strictly equals θ in the optimal solution for $\mathbf{e}$, is given, we will show how to obtain the optimal γ. WLOG, we assume that e_t have been sorted such that $e_1 \geq e_2, \ldots, \geq e_T$,

$$\sum_{t=1}^T e_t = \omega\theta + \frac{1}{\sqrt{2\gamma}} \sum_{t=\omega+1}^T \left(\sum_{l=1}^M ||\tilde{\mathbf{w}}_{l,t}||^{4/3}\right)^{3/4}. \tag{4.23}$$

It can be similarly proved as in [99] that the constraint $\sum_{t=1}^{T} e_t \leq 1$ always holds as the equality constraint, thus γ can be obtained as the function of ω as,

$$\sqrt{2\gamma} = \frac{\sum_{t=\omega+1}^{T} \left(\sum_{l=1}^{M} ||\tilde{\mathbf{w}}_{l,t}||^{4/3} \right)^{3/4}}{1 - \omega\theta}, \tag{4.24}$$

then together with (4.22), for groups that $e_t < \theta$, one gets the solution in (4.16). Thus we finish the proof.

To determine ω, the number of the elements in $\mathbf{e}$ with value strictly equal to θ, we have the following lemma:

Lemma 4.3 *Let $\mathbf{D}^\star$ and $\mathbf{e}^\star$ be the optimal solution to (4.14), and suppose that $a_1 \geq a_2 \geq \cdots \geq a_T$ with $a_t = \left(\sum_{l=1}^{M} ||\tilde{\mathbf{w}}_{l,t}||^{4/3} \right)^{3/4}$ for $t = 1, \ldots, T$. Then ω, the number of elements whose value strictly equal θ in $\mathbf{e}^\star$, is*

$$\min \left\{ p \in \{0, 1, \cdots, T-1\} \middle| \frac{(1 - p\theta) a_{p+1}}{\sum_{s=p+1}^{T} a_s} < \theta \right\}.$$

Thus the optimization for ω is simply a sorting algorithm as shown in Algorithm 4.

Suppose that the indices from 1 to T has been reordered according to a_t as that in Algorithm 4. To determine the group that strictly has $e_t = \theta$, we have the following Lemma:

Lemma 4.4 *Let $\mathbf{e}^\star$ be the optimal solution to the problem (4.14), and suppose $a_p > a_q$ for any two given indices $p,q \in \{1, \cdots, T\}$. If $e_q^\star = \theta$, we have $e_p^\star = \theta$.*

The proofs of Lemma 4.3 and 4.4 are omitted here due to space limitation.

4.4.3 Overall Optimization Procedure for MKL

The whole optimization procedure for solving the soft margin group sparse MKL in (4.9) is detailed in Algorithm 5. Taking semi-supervised learning as an example, after obtaining the optimized $\mathbf{D}$ and $\boldsymbol{\alpha}$ with $\mathcal{C}_m = \{\mathbf{y}^{m,1}, \ldots, \mathbf{y}^{m,T}\}$, the learnt classifier is expressed as:

$$f(\mathbf{x}) = \sum_{i:\ \alpha_i \neq 0}^{n} \alpha_i \Big(\sum_{m,t: d_{m,t} \neq 0} d_{m,t} \mathrm{y}_i^{m,t} k_m(\mathbf{x}, \mathbf{x}_i) \Big).$$

Algorithm 5 : The block-wise coordinate descent algorithm for solving the soft margin group sparse MKL

1: Initialize $\mathbf{D}^1$.
2: $r = 1$
3: **while** the stop criterion is not satisfied **do**
4: Get $\boldsymbol{\alpha}^r$ by solving the subproblem (4.12) using the standard QP solver with $\mathbf{D}^r$.
5: Calculate $\|\tilde{\mathbf{w}}_{m,t}\|$ according to (4.13) and update $\mathbf{D}^{r+1}$ by solving (4.14).
6: $r = r + 1$.
7: **end while**

4.4.4 Computational Complexity for IOKL

Now we analyze the computational complexity for our proposed IOKL. The computational complexity of IOKL in Algorithm 3 depends on the cutting-plane strategy as well as the inner group sparse MKL problem as shown in Algorithm 5. Specifically, in Algorithm 3, the 2-nd step takes about $MO(n^2)$ time complexity to construct the input-output kernels, while the 3-rd step takes $O(MKL)$ time complexity to solve the group MKL problem. In the 4-th step, it takes $O(n\log(n))$ time complexity to infer the most violated ambiguity candidate. In total, it takes $O(IOKL) = \Gamma(M \times O(n^2) + O(MKL))$ with Γ being the number of iterations.

For the complexity of the group sparse MKL as shown in Algorithm 5, it shares the similar form with that of previous analyze for MKL in Chapter 3. However, due to the group structure for updating the kernel combination coefficients, the complexity of it could be given as $O(MKL) = R \times O(n^{2.3} + MTn + M\log(M))$ with R being the total number of iterations for Algorithm 5.

4.5 Experiments

4.5.1 Text-based Image Retrieval on NUS-WIDE Dataset

In this section, we show the experimental results of our IOKL framework for a computer vision application (i.e., text-based image retrieval [56]) on the NUS-WIDE dataset [39]. This dataset contains 269,648 images collected from *Flickr.com* and annotations for 81 semantic concepts. Following [39], the dataset is partitioned into a training set consisting of 161,789 images and a test set with 107,859 images.

Table 4.1: MAP (%) of the different MIL methods over 81 concepts on the NUS-WIDE dataset.

Method	SIL-SVM	mi-SVM	sMIL	MIL-CPB	IOKL-SM
MAP	57.54	58.63	59.71	61.49	**64.36**

As in [39, 128], three types of global visual features (*i.e.*, Grid Color Moment (225 dim), Wavelet Texture (128 dim) and Edge Direction Histogram (73 dim)) are extracted for each of the images. The three types of visual features are then concatenated into a 426-dimensional feature vector, and PCA is further used to project the feature vector into a 119-dimensional visual vector, preserving 90% of the energy. Also, a 200-dimensional term-frequency feature is extracted as the texture feature, and is concatenated with the 119-dimensional global visual feature. Also, we extract the local SIFT features [139], and quantize the SIFT features with codebook size of 1024 to form a 21504-dimensional LLC feature vector following [199].

The aforementioned two types of features are used to construct the input base kernels. For each type of features, we utilize the Gaussian kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma D^2(\mathbf{x}_i, \mathbf{x}_j)\big)$), where $D(\mathbf{x}_i, \mathbf{x}_j)$ denotes the Euclidean distance between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. We set $\gamma = 2^n\gamma_0$, where $n \in \{-1, -0.5, \ldots, 1\}$ and $\gamma_0 = 1/A$ with A being the mean value of the square distances between all the training samples. Thus 10 input base kernels are used.

We use 25 positive bags and 25 negative bags with each bag consisting of 15 instances to train one-versus-all classifiers for all 81 concepts. For performance evaluation, we use the non-interpolated Average Precision (AP), which has been widely used as the standard performance metric for image retrieval applications. Mean AP (MAP) represents the mean of APs over all the 81 concepts from the dataset.

The effectiveness of IOKL for MIL: We firstly show the results of IOKL and some representative MIL methods in Table 4.1, which include SIL-SVM [3], mi-SVM [3], sMIL [23] and MIL-CPB [128]. MIL-CPB can be regarded as a special case of our method by using the single input kernel as in [128] with $\theta \geq 1$. These results clearly show the effectiveness of our proposed framework for MIL with application to text-based image retrieval on NUS-WIDE.

Table 4.2: MAPs (%) of our IOKL using different regularization settings on the NUS-WIDE dataset.

Method	$\ell_{A,1}$	$\ell_{A,2}$	$\ell_{1,1}$	$\ell_{2,2}$	$\ell_{1,2}$	$\ell_{2,1}$	SM$\ell_{2,1}$
MAP	61.84	60.78	61.04	60.02	55.95	62.82	**64.36**

The effectiveness of $\ell_{2,1}$-norm regularization: To verify the non-sparseness for input base kernels and sparseness for ambiguity candidates, we compare different regularization settings in Table 4.2 as $\ell_{i,j}$, where $i = A, 1, 2$ represent averaging, ℓ_1-norm and ℓ_2-norm for input base kernels, respectively, and $j = 1, 2$ represent ℓ_1-norm and ℓ_2-norm for ambiguity candidates, respectively. Also, SM$\ell_{2,1}$ is $\ell_{2,1}$ with soft margin regularization.

Table 4.2 shows MAPs of different regularization settings for the IOKL framework. We can observe that enforcing sparseness for input base kernels leads to poor performances (*e.g.*, 61.04% for $\ell_{1,1}$ compared with 62.82% for $\ell_{2,1}$), which is consistent with most observations of traditional MKL. Moreover, enforcing sparseness for ambiguity candidates improves the performance (62.82% for $\ell_{2,1}$ compared with 60.02% for $\ell_{2,2}$). Also, improper utilization of group structure such as in the $\ell_{1,2}$ case degenerates the performance greatly. These results clearly demonstrate the benefits of preserving non-sparseness for input base kernels and enforcing sparseness for ambiguity candidates with the $\ell_{2,1}$-norm.

The effectiveness of soft margin regularization: As discussed previously, the soft margin case reduces to the hard margin case for large value of θ. We show the influence of the soft margin regularization parameter θ in Figure 4.1. The IOKL-HM is IOKL with $\ell_{2,1}$-norm regularization, and IOKL-SM is IOKL by using soft margin $\ell_{2,1}$-norm regularization. We observe that θ influences the final performance greatly, and IOKL-SM can achieve the best 64.36% in MAP. Therefore, our proposed soft margin regularization can effectively learn a more robust classifier.

The complexity of IOKL: The complexity of the IOKL framework depends on the number of input base kernels, the number of iterations for cutting-plane method and the regularization strategy for the input-output kernel coefficients. In this part, the first concept "airport" from the NUS-WIDE data set is taken as an example, and the training time under different regularization settings for IOKL is reported. The training CPU time

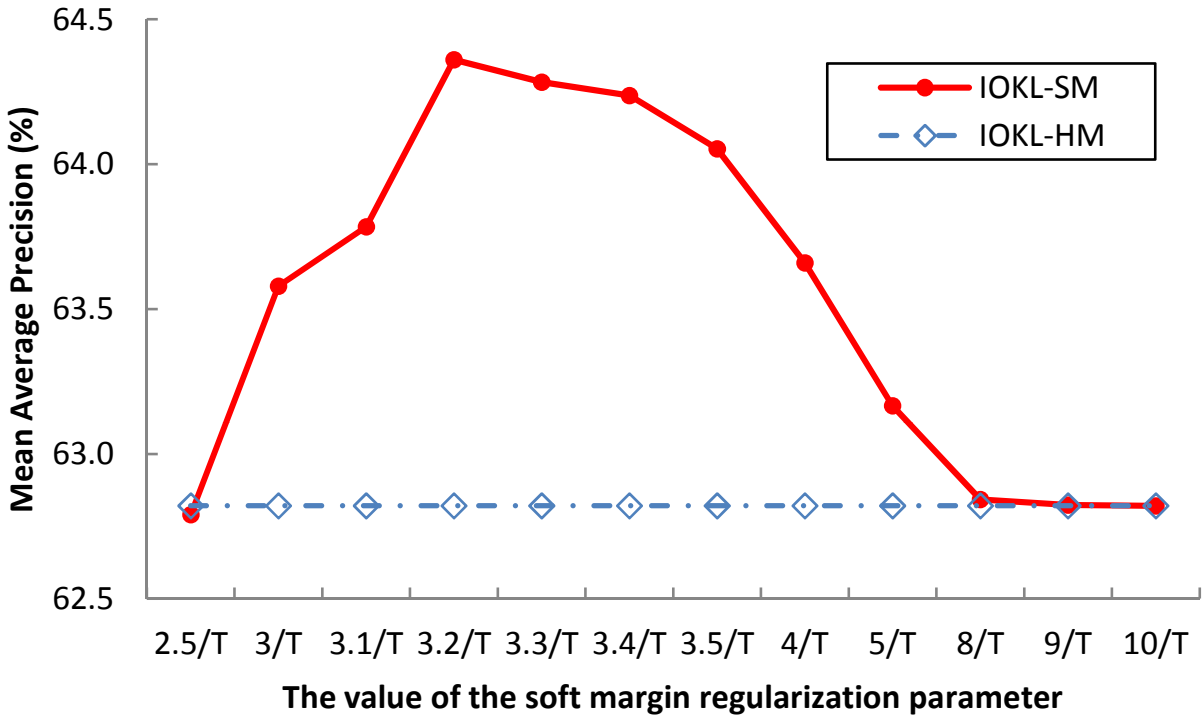


Figure 4.1: The MAP (%) over 81 concepts of our proposed IOKL-SM with respect to the regularization parameter θ on the NUS-WIDE dataset. Note that T in x-axis is the size of $\mathcal{C}_m$ in Algorithm 3.

Table 4.3: The number of input base kernels (#IK), training CPU time (CPU time), the number of selected input-output kernels (#IOK) and the number of selected output kernels (#OK) of our IOKL under different regularization settings for concept "airport".

Method	$\ell_{A,1}$	$\ell_{A,2}$	$\ell_{1,1}$	$\ell_{2,2}$	$\ell_{1,2}$	$\ell_{2,1}$	SM$\ell_{2,1}$
#IK	1	1	10	10	10	10	10
CPU time	133.42	346.09	8239.8	2380.2	26003	1282.5	591.72
#IOK	12	29	69	310	156	200	120
#OK	12	29	19	31	31	20	12

Table 4.4: Testing accuracy (%) on semi-supervised learning benchmark datasets

#l	Method	g241c	g241d	Text	Digit1	USPS	BCI	AveRank
10	SVM	52.66 (9)	53.34 (7)	54.63 (8)	69.40 (9)	79.97 (7)	50.15 (9)	8.17
	TSVM [96]	75.29 (2)	49.92 (8)	**68.79** (1)	82.23 (6)	74.80 (9)	50.85 (6)	5.33
	LDS [32]	71.15 (3)	49.37 (9)	63.85 (6)	84.37 (4)	**82.43** (1)	50.73 (8)	5.17
	LapSVM [12]	53.79 (7)	54.85 (3)	62.72 (7)	91.03 (2)	80.95 (3)	50.75 (7)	4.83
	LapRLS [12]	56.05 (6)	54.32 (4)	66.32 (5)	**94.56** (1)	81.01 (2)	51.03 (5)	3.83
	meanS3vm [131]	65.48 (4)	**58.94** (1)	66.91 (4)	83.00 (5)	77.84 (8)	52.07 (3)	4.17
	ℓ_2MKL [99]	52.16 (8)	53.67 (5)	54.62 (9)	73.79 (8)	80.76 (4)	52.16 (2)	6.00
	IOKL-HM	64.86 (5)	53.93 (6)	67.70 (3)	82.18 (7)	80.46 (5)	51.69 (4)	5.00
	IOKL-SM	**80.66** (1)	56.32 (2)	68.53 (2)	85.88 (3)	80.46 (5)	**52.48** (1)	**2.33**
100	SVM	76.89 (6)	75.36 (7)	73.55 (8)	94.47 (7)	90.25 (8)	65.69 (5)	6.83
	TSVM [96]	81.54 (3)	77.58 (3)	75.48 (7)	93.85 (9)	90.23 (9)	66.75 (4)	5.83
	LDS [32]	81.96 (2)	76.26 (5)	76.85 (3)	96.54 (3)	95.04 (3)	56.03 (9)	4.17
	LapSVM [12]	76.18 (8)	73.64 (8)	76.14 (6)	96.87 (2)	95.30 (2)	67.61 (3)	4.83
	LapRLS [12]	75.64 (9)	73.54 (9)	76.43 (5)	**97.08** (1)	**95.32** (1)	68.64 (2)	4.50
	meanS3vm [131]	80.25 (4)	77.58 (3)	76.60 (4)	95.91 (4)	93.17 (4)	**71.44** (1)	**3.33**
	ℓ_2MKL [99]	76.71 (7)	75.38 (6)	72.77 (9)	94.15 (8)	91.15 (5)	64.56 (7)	7.00
	IOKL-HM	79.86 (5)	**78.39** (1)	**77.86** (1)	94.79 (6)	90.78 (6)	64.33 (8)	4.50
	IOKL-SM	**83.62** (1)	**78.39** (1)	**77.86** (1)	94.88 (5)	90.78 (6)	65.06 (6)	**3.33**

with an IBM workstation (2.79GHz CPU with 32GB RAM) and Matlab implementation is reported in Table 4.3. The number of input base kernels (#IK), the number of selected base input-output kernels (#IOK), the number of output kernels (#OK) are also included in the table. We can observe the efficiency of the soft margin regularization (SM$\ell_{2,1}$) even compared with the single average base kernel (*i.e.*, $\ell_{A,1}$, $\ell_{A,2}$) using state-of-the-art MKL optimization techniques [99, 186, 222]. Also, the ℓ_1-norm selects fewer kernels, and the ℓ_2-norm selects more kernels, either in #IOK or #OK. Moreover, pursuing sparseness for input base kernels (*e.g.*, $\ell_{1,1}, \ell_{1,2}$), although it leads to a sparser solution, results in more training time and degenerated performance.

Table 4.5: Testing accuracy (%) on multiple instance classification benchmark datasets

Method	Musk1	Musk2	Elephant	Fox	Tiger
DD [144]	88.0	84.0	N/A	N/A	N/A
EM-DD [238]	84.8	84.9	78.3	56.1	72.1
MI-Kernel [73]	88.0	89.3	84.3	60.3	84.2
mi-SVM [3]	87.4	83.6	82.0	58.2	78.9
MI-SVM [3]	77.9	84.3	81.4	59.4	84.0
miGraph [244]	88.9	**90.3**	86.8	61.6	86.0
MIGraph [244]	**90.0**	90.0	85.1	61.2	81.9
Bag-KI-SVM [130]	88.0	82.0	84.5	60.5	85.0
IOKL-HM	86.9	87.2	87.0	60.0	85.0
IOKL-SM	88.0	88.2	**88.0**	**63.5**	**86.5**

4.5.2 Semi-Supervised Learning Benchmark Datasets

We evaluate our proposed IOKL on six semi-supervised learning benchmark datasets[2], including *g241c, g241d, Text, Digit1, USPS* and *BCI*. We follow two standard settings, one using 10 labeled samples and the other using 100 labeled samples. The experiments are repeated 12 rounds following the provided partitions, and the average testing accuracy on the unlabeled data is used as the performance measure.

We utilize four types of kernel functions to construct the input base kernels: Gaussian kernel (RBF) (*i.e.*,$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma D^2(\mathbf{x}_i, \mathbf{x}_j)\big)$), Laplacian kernel (Lap) (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\sqrt{\gamma} D(\mathbf{x}_i, \mathbf{x}_j)\big)$), inverse square distance (ISD) kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\gamma D^2(\mathbf{x}_i, \mathbf{x}_j)+1}$) and inverse distance (ID) kernel (*i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{\gamma} D(\mathbf{x}_i, \mathbf{x}_j)+1}$), where $D(\mathbf{x}_i, \mathbf{x}_j)$ denotes the Euclidean distance between sample $\mathbf{x}_i$ and $\mathbf{x}_j$, and γ is the kernel parameter.

We set $\gamma = 1/A$ with A being the mean value of the square distances between all the training samples, thus we have four input base kernels in total. The SVM regularization parameters for the labeled samples and unlabeled samples are set to be 100 and in the range $\{0.1, 1\}$ respectively, and the soft margin regularization parameter θ is set to be in the range $\{1.2/T, 1.4/T, \ldots, 6/T\}$ with T being the size of $\mathcal{C}_m$ in Algorithm 3 for each iteration. The balance constraint for the unlabeled data is set to be the ground truth value following [30].

[2]http://olivier.chapelle.cc/ssl-book/benchmarks.html

The final performance is reported in Table 4.4. We compare our learning framework with supervised MKL using labeled data only (*i.e.*, ℓ_2MKL), semi-supervised learning using the group sparse MKL (*i.e.*, IOKL-HM) and semi-supervised learning using the soft margin group sparse MKL (*i.e.*, IOKL-SM). We also include results reported by other SVM-type methods from the literature for comparison, including TSVM [96], LDS [32], LapSVM [12], LapRLS [12] and meanS3svm [131].

We can observe from Table 4.4 that the proposed IOKL achieves very competitive results for semi-supervised learning. We also report the average rank of the different algorithms. Note that when the number of labeled data is 10, the IOKL-SM achieves a much better result in the average rank compared with other methods; when the number of labeled data is 100, the difference between different algorithms becomes small. This may come from the fact that the less the labeled data is used, the more uncertainty is associated with the output labels. Moreover, comparing the IOKL-SM with IOKL-HM under all settings, we can observe the effectiveness of our proposed soft margin regularization.

4.5.3 Multi-Instance Learning Benchmark Datasets

We evaluate our proposed IOKL on five popular multiple instance classification task benchmark datasets[3], including *Musk1, Musk2, Elephant, Fox* and *Tiger*, which have been widely used in the literature. In the experiments, we utilize the same four types of kernel functions (*i.e.*, RBF, Lap, ISD, ID) with section 4.5.2.

We show the final performance by using the IOKL framework in Table 4.5. The results are all based on 10-fold cross validation accuracy following the common settings on these datasets. In the lower part of the table, we list results from Bag-KI-SVM [130] and IOKL-HM and IOKL-SM. The Bag-KI-SVM becomes a special case of our framework by using a Gaussian kernel with IOKL-HM.

The other representative MIL methods are also shown in the upper part of Table 4.5, including non-SVM-based methods (*i.e.,* DD [144], EM-DD [238]), graph-based methods (*i.e.,* MIGraph [244], miGraph [244]) and SVM-based methods (*i.e.,* MI-SVM [3], mi-SVM [3] and MI-Kernel [73]). Our IOKL framework achieves very competitive results on

[3]www.cs.columbia.edu/∼andrews/mil/datasets.html

these benchmark datasets. More importantly, comparing IOKL-SM with IOKL-HM, we again observe the effectiveness of our proposed soft margin regularization.

4.6 Summary

In this chapter, we have proposed an Input-Output Kernel Learning framework for handling general data ambiguities. By introducing the concept of *input-output kernel*, the methodology from traditional MKLs designed for supervised learning only is applicable for handling general data ambiguity problems such as SSL and MIL. To learn a more robust classifier, we further introduce a novel soft margin group sparse MKL formulation. In addition, a block-wise coordinate descent algorithm with an analytical solution for the kernel coefficients is developed to solve the new MKL formulation efficiently. The promising experimental results on the challenging NUS-WIDE dataset for a computer vision application (i.e., text-based image retrieval), SSL benchmark datasets and MIL benchmark datasets demonstrate the effectiveness of our proposed IOKL framework. In the future, we would like to extend our IOKL framework to solve more ambiguity problems such as clustering [133] and relative outlier detection [124], [123].

Chapter 5

Distance Metric Learning using Privileged Information

In this chapter, we propose a novel approach to improve face verification and person re-identification in RGB images by leveraging a set of RGB-D data, in which we have additional depth images in the training data captured by using depth cameras such as Kinect. Specifically, we extract visual features and depth features from the RGB images and depth images, respectively. As the depth features are only available in the training data, we treat the depth features as privileged information, and we formulate this task as a distance metric learning with privileged information problem. Unlike traditional face verification and person re-identification tasks which only use visual features, we further employ the extra depth features in the training data to improve the learning of distance metric in the training process. Based on the recent information-theoretic metric learning (ITML) method, we propose a new formulation called Information-theoretic Metric Learning with Privileged Information (ITML+) for this task. We also present an efficient algorithm based on the cyclical projection method for solving the proposed ITML+ formulation. Extensive experiments on the challenge faces datasets EUROCOM and CurtinFaces for face verification as well as the BIWI RGBD-ID dataset for person re-identification demonstrate the effectiveness of our proposed approach.

5.1 Introduction

Face verification and person re-identification are two important problems in computer vision, which have attracted attentions from many researchers in the last two decades [2,

26, 38, 77, 78, 80]. In both tasks, a few pairs of training images (*i.e.*, faces images or the images containing the whole head and body areas) are provided together with side information (*i.e.*, we only know whether each pair of images is from the same or different subjects instead of the names of those subjects in the images). The target of both tasks is to decide whether two test images are from the same subject or not.

Given only side information, one can learn a Mahalanobis distance metric for face verification or person re-identification. After that, the distance between a pair of testing images is used to decide whether they are from the same subject or different subjects [25, 77, 78, 80, 152, 207]. However, most of those existing works for face verification and person re-identification are based on the RGB images only. On the other hand, with the advance of new depth cameras such as Kinect, one can easily capture depth information together with RGB images when collecting training data for computer vision tasks. A few labeled RGB-D datasets were recently released to the public. Compared with RGB images, depth information is more robust to illumination changes, complex background, etc., thus it can provide useful information for many vision tasks, such as face recognition [119], gender classification [90], and object recognition [112]. However, those works require depth information and RGB information in both the training and test stages, which limits them for a broader range of applications, where the testing images (*i.e.*, the images captured by conventional surveillance cameras) do not contain depth information.

In this chapter, we propose a new scheme for recognizing RGB images by learning from a set of weakly labeled RGB-D training data, and our method can be used for face verification and person re-identification. As shown in Fig. 5.1, in our work, the training data consists of a few pairs of RGB images and the corresponding depth images together with side information, which are referred to as weakly labeled RGB-D training data. Our goal is to decide whether a pair of RGB testing images come from the same subject or not. In the training process, we firstly extract the visual features and depth features from the RGB images and depth images, respectively. Then we learn a robust Mahalanobis distance metric in the visual feature space by using both visual and depth features. In the testing process, we use the learnt Mahalanobis distance metric to determine whether a pair of RGB images are from the same subject or not only based on their visual features.

To learn the Mahalanobis distance metric under the new learning scheme, we propose a novel distance metric learning method called Information-theoretic Metric Learning

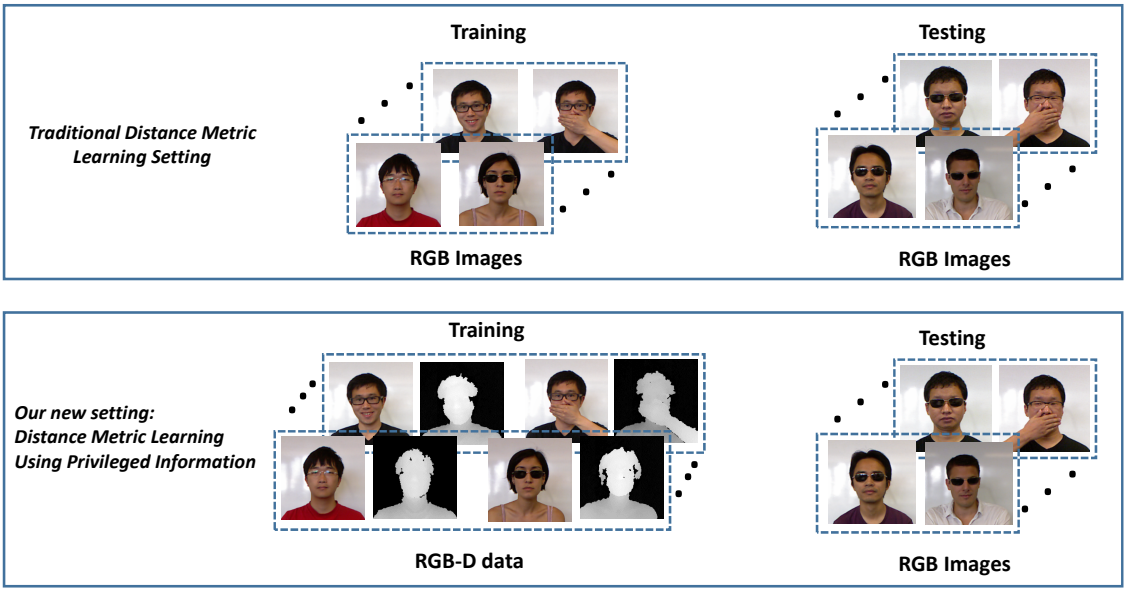


Figure 5.1: The comparison of the traditional distance metric learning setting and our new distance metric learning setting using privileged information.

with Privileged Information (ITML+) by formulating a new objective function based on the existing work ITML [52]. To effectively utilize the additional depth features in the training data, inspired by the recent work SVM+ [192], we model the loss term for each pair of visual training samples (*i.e.*, the training samples with visual features) by using the corresponding pair of depth training samples (*i.e.*, the training samples with depth features). In this way, the distance between two visual training samples can be affected by their corresponding depth training samples. An efficient cyclic projection method with analytical solution is also proposed to solve the new optimization problem. We conduct extensive experiments on the real-world EUROCOM and CurtinFaces datasets as well as BIWI RGBD-ID dataset and we demonstrate the effectiveness of our proposed ITML+ algorithm for improving the face verification and person re-identification performances in RGB images by utilizing the additional depth images.

This chapter is organized as follows. In section 5.2, we briefly review the related works. The proposed ITML+ algorithm is presented in Section 5.3 and its solution is provided in Section 5.4. In Section 5.5, we report the experimental results as well as the detailed analysis. Finally, the conclusion is given in Section 5.6.

5.2 Related Work

Our work is related to the distance metric learning methods and the recent works on learning using privileged information as well as the existing works on face verification and person re-identification.

5.2.1 Distance Metric Learning

Our work is related to the distance metric learning works [13, 52, 80, 107, 204, 214, 232]. The early work for the Mahalanobis distance metric learning in [214] formulates the distance metric learning problem as a convex optimization problem that maximizes the sum of distances between dissimilar pairs while minimizing the sum of distances between similar pairs. A projected gradient descent method was proposed to solve the proposed objective function, but the SVD operation on the distance metric $\mathbf{M}$ makes the algorithm only applicable to small scale problems. Following [214], a large number of methods were proposed in literature (see the surveys [13, 107, 232] for comprehensive reviews of different metric learning methods). The two representative methods for distance metric learning are the Large Margin Nearest Neighbors (LMNN) method [204] and the Information-theoretic Metric Learning (ITML) [52] method.

The LMNN [204] method was proposed for the nearest neighbor classifier by constraining the data in a local way, *i.e.*, the k nearest neighbors of any training instance from the same class should be closer to each other, while the instances from other classes should be kept away by a margin. The constraints are thus given in a triplet form, which requires two samples from the same class and one additional sample from the other class. Thus the explicit class label information is usually required for each sample in the training set to obtain such constraints. The ITML method [52] is based on the pairwise constraints, which assumes that the positive pairs are from the same class and the negative pairs are from different classes without requiring knowing the class label for each sample in the training set. The work in [52] introduced the LogDet divergence based regularization to the distance metric, and such regularization makes the optimization simpler and more efficient.

Different from the existing distance metric learning methods [52, 80, 204, 214], our proposed method for distance metric learning using privileged information aims to learn

a robust distance metric by further exploiting additional privileged information (*i.e.*, the depth features) in training data. There are also several multi-modal distance metric learning methods [146, 213], where multiple types of features are assumed to be available for both training and testing data. In these methods, the final decision is made based on all types of features, the learnt models are not suitable to the learning setting in our work. In the recent methods [49, 140], the distance metric is learnt for each type of features, and we can directly apply the learnt distance metric corresponding to the RGB images in the testing stage. However, our experiments show that those methods are also worse when compared with our proposed ITML+.

5.2.2 Learning Using Privileged Information

The recently proposed Learning Using Privileged Information (LUPI) method [161, 174] used privileged information to improve SVM for the supervised binary classification tasks. In SVM+ [174], privileged information is utilized to construct the correcting function to control the losses in the objective function. Given a set of n training data $\{\mathbf{x}_i\}|_{i=1}^n$ with $\mathbf{x}_i \in \mathcal{R}^h \subset \mathcal{X}$, where h is the feature dimension of each sample, we refer to $\mathcal{X}$ as the *decision space* as suggested in [161] because the final decision is based on the features of the testing samples in the space $\mathcal{X}$. Except for the training data in the decision space $\mathcal{X}$, the additional privileged feature $\{\mathbf{z}_i\}|_{i=1}^n$ with $\mathbf{z}_i \in \mathcal{R}^g \subset \mathcal{Z}$ in the *correcting space* $\mathcal{Z}$ [161] is only available for the training set, but it is not available for the test set. In [174], the task is to utilize the training data $\{\mathbf{x}_i, \mathbf{z}_i\}|_{i=1}^n$ as well as their labels $\{\mathrm{y}_i\}|_{i=1}^n$ to train a classifier for classifying the test data $\{\mathbf{x}_i\}|_{i=n+1}^{n+m}$ under the SVM framework for the supervised binary classification problem. Specifically, the linear target classifier $f(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b$ is learnt on the decision space $\mathcal{X}$ only in order to classify the test data. At the same time, another function $\xi = \mathbf{v}'\mathbf{z} + \rho$ is learnt on the correcting space $\mathcal{Z}$ by modeling privileged information as the loss function. The objective function of SVM+ is proposed as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{v},b,\rho} \quad & \frac{1}{2}\left(||\mathbf{w}||^2 + \lambda||\mathbf{v}||^2\right) + C\sum_{i=1}^{l}(\mathbf{v}'\mathbf{z}_i + \rho) \\ \text{s.t.,} \quad & \mathrm{y}_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - (\mathbf{v}'\mathbf{z}_i + \rho), \forall i = 1, \ldots, l, \\ & \mathbf{v}'\mathbf{z}_i + \rho \geq 0, \ \forall i = 1, \ldots, l. \end{aligned}$$

The above formulation can be reformulated in the dual form as a standard Quadratic Programming (QP) problem, which can be solved efficiently by using any state-of-the-art QP solvers.

It is shown in [174] that the convergence rate of SVM based algorithms can be improved by using such a correcting function to incorporate privileged information. As different algorithm has different kinds of loss function and decision function, we cannot directly extend it to the other learning scenarios such as distance metric learning.

Following the LUPI method [192], the work in [68] extended [192] for the clustering problem, while the work in [174] extended it into the Ranking SVM for the ranking problem. The recent work in [69] proposed an extension of the learning scenario to distance metric leaning. However, their proposed method is a two-step approach to utilize privileged information [69]. They firstly trained a distance metric based on ITML using privileged information. Based on the distance metric learnt from privileged information, some pairs of training samples are removed. Then ITML is retrained again by using the remained pairs based on the main features. Instead of using a two-step approach, in this chapter we follow the existing work SVM+ [192], and propose a new formulation from the correcting function perspective to utilize privileged information for the distance metric learning problem.

5.2.3 Face Verification and Person Re-identification

Our work is related to the face verification works. Generally, the existing face verification methods can be categorized into feature based methods and distance metric learning based methods. The feature based methods [38, 111, 197, 206] developed better face descriptors. For example, in [38], an unsupervised learning approach is proposed to encode the micro-structures of a face image. In [111], the output of the attributes and simile classifiers are used as the mid-level features to represent a face image for the face verification task. In contrast, the distance metric learning based works [25, 80, 152, 207] developed new metric learning methods for the face verification task. Specifically, two face images from the same person are regarded as a similar pair, while two face images from different persons are regarded as a dissimilar pair. Based on the extracted low level visual features (*i.e.*, SITF [139], HOG [50], LBP [2]) for each of face images, the Mahalanobis

distance metric is learnt by using these low level visual features on the training samples, and the learnt distance metric is applied to a pair of test samples with the same type of low-level visual features. The distance metric learning methods have been successfully applied to the face verification task on the benchmark datasets such as LFW [87]. The ITML method [52] was proposed for distance metric learning by considering the pairwise constraints as side information, while the work in [80] proposed a discriminant metric learning method that takes advantages of all the pairs of samples in the dataset, and the work in [152] proposed a cosine similarity metric learning method.

Person re-identification is another related task by using images containing the whole head and body areas. Recently, many benchmark datasets have been released for the person re-identification task, and the most famous ones are VIPeR [78] and CAVIAR4REID [55]. Many methods for person re-identification have been proposed, which include features based methods [78], [79], [64], [117], [10], [241], [76], as well as learning based methods [242], [163], [104], [83], [82]. The reviews of the related works on person re-identification have been published in a recent book [77]. The feature based methods aim to develop better descriptors for the human body areas by using spatial temporal appearances [76], salience learning [241], etc. The learning based methods aim to develop more effective learning algorithms for person re-identification task, such as probabilistic relative distance comparison [242], rank SVM [163] and KISSME [104]. The distance metric learning methods such as LMNN and ITML have also been successfully used for the person re-identification task [77].

5.3 Distance Metric Learning with Privileged Information

In this section, we first introduce the problem setting of our face verification and person re-identification task. Then we review the objective function of ITML. After that, we develop the objective function of our new method Information-theoretic Metric Learning with Privileged Information (ITML+). We also present a variant of our ITML+ called *partial ITML+* for the case that only parts of training data contain privileged information.

5.3.1 Problem Statement

In our task, the training data is a few pairs of RGB-D images together with side information describing whether each pair of RGB-D images belong to the same subject or not. In the training process, we extract the visual features and depth features from the RGB images and depth images, respectively. Formally, let us denote the visual features as $\{\mathbf{x}_i\}|_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{R}^h$ is the visual feature vector extracted from the RGB image of the i^{th} training sample, and n is the number of training samples. Similarly, we denote the depth features as $\{\mathbf{z}_i\}|_{i=1}^n$, where $\mathbf{z}_i \in \mathcal{R}^g$ is the depth feature vector extracted from the depth image of the i^{th} sample. We also use $(\mathbf{x}_i, \mathbf{z}_i)$ to denote the i-th training sample.

We also have side information for the training data, namely we have a set of similar pairs $\mathcal{S}$ and a set of dissimilar pairs $\mathcal{D}$. For each similar pair $(i, j) \in \mathcal{S}$ (*resp.*, dissimilar pair $(i, j) \in \mathcal{D}$), the two corresponding training samples $(\mathbf{x}_i, \mathbf{z}_i)$ and $(\mathbf{x}_j, \mathbf{z}_j)$ are from the same subject (*resp.*, different subjects). Our goal is to learn a distance metric $\mathbf{M} \in \mathcal{R}^{h\times h}$ that can be used to classify a pair of test data that only contains the RGB images and does not have the depth images for the face verification and person re-identification task. In other words, based on the training RGB-D images $\{(\mathbf{x}_i, \mathbf{z}_i)\}|_{i=1}^n$ together with side information in $\mathcal{S}$ and $\mathcal{D}$, we aim to learn a Mahalanobis distance $d_{\mathbf{M}}(\cdot, \cdot)$ defined as,

$$d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j), \tag{5.1}$$

where we use the squared distance for the ease of representation in this chapter. Intuitively, we expect the learnt Mahalanobis distance $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ can output a large value if $(i, j) \in \mathcal{D}$, and a small value if $(i, j) \in \mathcal{S}$. In the testing process, we use the learnt Mahalanobis distance on each pair of test samples, and determine whether the two corresponding RGB images are from the same subject or not based on their Mahalanobis distance.

5.3.2 Information-theoretic Metric Learning (ITML)

The key idea of ITML is to learn the distance metric $\mathbf{M}$ by enforcing that the learnt distance $d_{\mathbf{M}}$ is high for dissimilar pairs of samples and low for similar pairs of samples. Specifically, they expect $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq u$ for a relatively small value u if $(i, j) \in \mathcal{S}$,

and $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq l$ for a sufficiently large l if $(i,j) \in \mathcal{D}$. However, for the real-world applications, a feasible solution may not exist after using those strict constraints. Thus, a slack variable ξ_{ij} is introduced for each constraint. Let us define $\boldsymbol{\xi} \in \mathcal{R}^{|\mathcal{D}|+|\mathcal{S}|}$ as the vector of slack variables, where each entry ξ_{ij} corresponds to one training pair (i,j). Then, the objective function of ITML [52] is formulated as follows,

$$\begin{aligned} \min_{\mathbf{M} \succeq 0, \xi_{ij}} \quad & D_{ld}(\mathbf{M}, \mathbf{M}^0) + \gamma L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) \\ \text{s.t.,} \quad & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq \xi_{ij}, \quad (i,j) \in \mathcal{S}, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq \xi_{ij}, \quad (i,j) \in \mathcal{D}, \end{aligned} \tag{5.2}$$

where $\boldsymbol{\xi}^0 \in \mathcal{R}^{|\mathcal{D}|+|\mathcal{S}|}$ is a vector with each entry $\xi^0_{ij} = \begin{cases} u & (i,j) \in \mathcal{S}, \\ l & (i,j) \in \mathcal{D} \end{cases}$, $L(\boldsymbol{\xi}, \boldsymbol{\xi}^0)$ is the loss term which measures the difference between $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^0$, $D_{ld}(\mathbf{M}, \mathbf{M}^0)$ is a regularizer based on Bregman divergence to avoid the trivial solution.

Given any strictly convex differentiable function $\phi(\cdot)$ over a convex set, the Bregman divergence [108] over two matrices $\mathbf{M}$ and $\mathbf{M}^0$ is defined as

$$D_\phi(\mathbf{M}, \mathbf{M}^0) = \phi(\mathbf{M}) - \phi(\mathbf{M}^0) - \operatorname{tr}\left((\mathbf{M} - \mathbf{M}^0)' \nabla \phi(\mathbf{M}^0)\right).$$

By using the Burg entropy function $\phi(\mathbf{M}) = -\text{logdet}\mathbf{M}$, we can define the LogDet divergence (or the Burg matrix divergence) [52, 108] as:

$$D_{ld}(\mathbf{M}, \mathbf{M}^0) = \operatorname{tr}(\mathbf{M}(\mathbf{M}^0)^{-1}) - \log\det(\mathbf{M}(\mathbf{M}^0)^{-1}) - h, \tag{5.3}$$

where h is the dimension of $\mathbf{M}$. $\mathbf{M}^0 \in \mathcal{R}^{h \times h}$ is a predefined matrix, which is often set to be the identity matrix $\mathbf{I}$. Moreover, the loss term $L(\boldsymbol{\xi}, \boldsymbol{\xi}^0)$ can be written as $L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) = D_{ld}(\text{diag}(\boldsymbol{\xi}), \text{diag}(\boldsymbol{\xi}^0))$, which is the LogDet divergence between two diagonal matrices. Thus, ITML aims to minimize the difference between the slack variables $\boldsymbol{\xi}$ and the ideal distances $\boldsymbol{\xi}^0$ as well as enforce the learnt Mahalanobis metric $\mathbf{M}$ close to the identity matrix to avoid the trivial solution.

5.3.3 Information-theoretic Metric Learning with Privileged Information (ITML+)

Recall in our task, we additionally have the depth features in the training data. As ITML only considers one type of features when learning the Mahalanobis distance metric, we

thus propose a new distance metric learning method called Information-theoretic Metric Learning with Privileged Information (ITML+) to learn a more robust Mahalanobis distance metric in the visual feature space by further utilizing the additional depth features in the training data.

Inspired by the SVM+ method [192], we consider to use the additional depth features to correct the loss of each pair of training samples in the visual feature space. Specifically, we replace the slack variable ξ_{ij} in (5.2) by using a slack function in the depth feature space, *i.e.*, $\xi_{ij} = d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j) = (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}(\mathbf{z}_i - \mathbf{z}_j)$, where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the depth features of training samples from the pair (i, j), and $\mathbf{P} \in \mathcal{R}^{g\times g}$ is a Mahalanobis distance metric in the depth feature space. In this way, the distance between the training samples from the pair (i, j) in the depth feature space can serve as the correcting guidance for the distance calculated by using the visual features. Accordingly, the objective function for our ITML+ is formulated as follows,

$$\begin{aligned} \min_{\mathbf{M}\succeq 0, \mathbf{P}\succeq 0} \; & \Omega(\mathbf{M}, \mathbf{P}) + \gamma \sum_{(i,j)\in\mathcal{S}\cup\mathcal{D}} \ell(d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \xi^0_{ij}) \\ \text{s.t.,} \; & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \quad (i, j) \in S, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \quad (i, j) \in \mathcal{D}, \end{aligned} \tag{5.4}$$

where $\Omega(\mathbf{M}, \mathbf{P}) = D_{ld}(\mathbf{M}, \mathbf{M}^0) + \lambda D_{ld}(\mathbf{P}, \mathbf{P}^0)$ is the regularization term by summing the LogDet divergence based regularization terms related to $\mathbf{M}$ and $\mathbf{P}$, γ and λ are two tradeoff parameters, $\mathbf{M}^0$ and $\mathbf{P}^0$ are two predefined matrices (we use the identity matrices), and $\ell(d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \xi^0_{ij}) = D_{ld}(d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \xi^0_{ij})$ is the LogDet divergence between $d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j)$ and ξ^0_{ij} as defined in (5.3).

Compared with the objective function of ITML in (5.2), the objective function of ITML+ in (5.4) additionally learns a Mahalanobis distance metric $\mathbf{P}$ in the depth feature space. We also replace the original slack variable ξ_{ij} in (5.2) with $d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j)$ for each pair (i, j). Accordingly, the constraints become $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \quad \forall (i, j) \in S$, and $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j)$ otherwise. Intuitively, for any pair (i, j) in the visual feature space, if the visual features $\mathbf{x}_i$ and $\mathbf{x}_j$ are corrupted by some noises (*e.g.*, illumination changes), the distance in the visual feature space $d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ may not be satisfied (*i.e.*, the distance is large if $(i, j) \in \mathcal{S}$ or small if $(i, j) \in \mathcal{D}$). Considering the depth features

are relatively robust to the illumination changes, the distance in the depth feature space $d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j)$ can be satisfied (*i.e.*, the distance is small if $(i,j) \in \mathcal{S}$ or large if $(i,j) \in \mathcal{D}$). In this case, by using the constraints in (5.4), the learnt Mahalanobis distance metric $\mathbf{M}$ in the visual feature space should be more robust to those noises. Specifically, our ITML+ will enforce that similar pairs become more similar while dissimilar pairs will become more dissimilar by using the distances in the depth feature space as the correcting guidance. The detailed analysis of the learnt distances by using both ITML and ITML+ are given in Fig. 5.4 in our experiments (see Section 5.5.5).

5.3.4 Partial ITML+

In real-world applications, not all the training data are always associated with depth information. To handle the situation where only a part of training data contains depth information, we further formulate a variant of our ITML+ method called *partial ITML+*. Specifically, let us denote the training set as the similar pair set $\mathcal{S}_p$ and dissimilar pair set $\mathcal{D}_p$ which only contain RGB information, then we can formulate our partial ITML+ as follows:

$$\begin{aligned} \min_{\mathbf{M}\succeq 0, \mathbf{P}\succeq 0, \xi_{ij}} \; & \Omega(\mathbf{M},\mathbf{P}) + \gamma L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) \\ \text{s.t.}, \; & d^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) \le d^2_{\mathbf{P}}(\mathbf{z}_i,\mathbf{z}_j), (i,j) \in \mathcal{S} - \mathcal{S}_p, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) \ge d^2_{\mathbf{P}}(\mathbf{z}_i,\mathbf{z}_j), (i,j) \in \mathcal{D} - \mathcal{D}_p, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) \le \xi_{ij}, \quad (i,j) \in \mathcal{S}_p, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) \ge \xi_{ij}, \quad (i,j) \in \mathcal{D}_p, \end{aligned} \tag{5.5}$$

where $L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) = \sum_{(i,j)\in(\mathcal{S}-\mathcal{S}_p)\cup(\mathcal{D}-\mathcal{D}_p)} \ell(d^2_{\mathbf{P}}(\mathbf{z}_i,\mathbf{z}_j), \xi^0_{ij}) + \sum_{(i,j)\in S_p \cup D_p} \ell(\xi_{ij}, \xi^0_{ij})$ is the loss term with $\ell(d^2_{\mathbf{P}}(\mathbf{z}_i,\mathbf{z}_j), \xi^0_{ij})$ (*resp.* $\ell(\xi_{ij}, \xi^0_{ij})$) being the LogDet divergence between $d^2_{\mathbf{P}}(\mathbf{z}_i,\mathbf{z}_j)$ (*resp.* ξ_{ij}) and ξ^0_{ij}, and $\Omega(\mathbf{M},\mathbf{P}) = D_{ld}(\mathbf{M},\mathbf{M}^0) + \lambda D_{ld}(\mathbf{P},\mathbf{P}^0)$ is defined similarly as in (5.4), and γ and λ are two tradeoff parameters.

In other words, for the pairs of training samples that have privileged information, we use the constraints from ITML+, and for the pairs of training samples that do not have privileged information, we still utilize the constraints from ITML. We can observe from (5.5) that (5.5) reduces to the ITML+ formulation in (5.4) if $\mathcal{S}_p = \varnothing, \mathcal{D}_p = \varnothing$, while

(5.5) reduces to the ITML in (5.2) if $\mathcal{S}_p = \mathcal{S}, \mathcal{D}_p = \mathcal{D}$. In this way, the proposed partial ITML+ in (5.5) can naturally bridge the ITML and ITML+ by varying the number of pairs of training samples with privileged information.

5.4 Solution to ITML+

In this section, we develop a new optimization algorithm for solving our ITML+ problem in (5.4) by using the cyclic projection method [21].

5.4.1 ITML+ with Explicit Correcting Function

The cyclic projection method cannot be directly applied to solve the new objective function in (5.4) for ITML+, because we have two variables $\mathbf{M}$ and $\mathbf{P}$ in the constraints. Let us introduce an intermediate variable ξ_{ij} for each constraint related to one pair (i, j), we then rewrite our ITML+ formulation in (5.4) to an equivalent form as follows,

$$\begin{aligned} \min_{\mathbf{M}\succeq 0, \mathbf{P}\succeq 0, \boldsymbol{\xi}} \; & D_{ld}(\mathbf{M}, \mathbf{M}^0) + \lambda D_{ld}(\mathbf{P}, \mathbf{P}^0) + \gamma L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) \\ \text{s.t.}, \; & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq \xi_{ij}, \quad (i,j) \in S, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq \xi_{ij}, \quad (i,j) \in \mathcal{D}, \\ & \xi_{ij} = d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \quad (i,j) \in \mathcal{S} \cup \mathcal{D}, \end{aligned} \tag{5.6}$$

where $L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) = D_{ld}(\text{diag}(\boldsymbol{\xi}), \text{diag}(\boldsymbol{\xi}^0))$ is the LogDet divergence between $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^0$ defined similarly as in (5.2). The equivalence between (5.6) and (5.4) can be easily verified by substituting the correcting function $\xi_{ij} = d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j)$ back into the objective function in (5.6).

Now we apply the cyclic projection method similarly to that in [52]. For the ease of presentation, we further unify the two inequalities in (5.6), and write the new objective function as follows,

$$\begin{aligned} \min_{\mathbf{M}\succeq 0, \mathbf{P}\succeq 0, \boldsymbol{\xi}} \; & D_{ld}(\mathbf{M}, \mathbf{M}^0) + \lambda D_{ld}(\mathbf{P}, \mathbf{P}^0) + \gamma L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) \\ \text{s.t.}, \; & \mathrm{y}_{ij} d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq \mathrm{y}_{ij} \xi_{ij}, \quad (i,j) \in \mathcal{S} \cup \mathcal{D}, \\ & \xi_{ij} = d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \quad (i,j) \in \mathcal{S} \cup \mathcal{D}, \end{aligned} \tag{5.7}$$

where $\mathrm{y}_{ij} = \begin{cases} 1 & (i,j) \in \mathcal{S} \\ -1 & (i,j) \in \mathcal{D} \end{cases}$, and other terms are as the same as those in (5.6).

It can be observed that the objective function in (5.7) is convex. Following the cyclic projection method [21], [52], we first initialize the solution to (5.7) as $(\mathbf{P}_0, \mathbf{M}_0)$. Then we iteratively pick up a pair of training samples (i, j), and update the current solution with Bregman projection such that the objective is minimized and the constraints w.r.t. this pair are also satisfied. The above process is repeated until all constraints are satisfied. We will detail Bregman projection in the next subsection.

5.4.2 Bregman Projection

Let us denote the solution at the t-th iteration as $(\mathbf{M}^t, \mathbf{P}^t)$. At the $(t+1)$-th iteration, we pick up a pair of training samples (i, j), then the new solution $(\mathbf{M}^{t+1}, \mathbf{P}^{t+1})$ can be obtained using Bregman projection by optimizing the following subproblem:

$$\min_{\mathbf{M}\succeq 0, \mathbf{P}\succeq 0, \xi_{ij}} D_{ld}(\mathbf{M}, \mathbf{M}^t) + \gamma \ell(\xi_{ij}, \xi_{ij}^t) + \lambda D_{ld}(\mathbf{P}, \mathbf{P}^t) \tag{5.8}$$

$$\text{s.t.,} \mathrm{y}_{ij} d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq \mathrm{y}_{ij} \xi_{ij}, \tag{5.9}$$

$$\xi_{ij} = d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j). \tag{5.10}$$

The above problem has analytical solutions for $\mathbf{M}, \mathbf{P}$ and ξ_{ij}, as shown in the following proposition:

Proposition 11 *The optimal solution $(\mathbf{M}, \mathbf{P}, \xi_{ij})$ to the problem in (5.8) can be obtained in closed form as follows,*

$$\mathbf{M}^{t+1} = \mathbf{M}^t - \frac{\mathrm{y}_{ij}\alpha_{ij}\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t}{1 + \mathrm{y}_{ij}\alpha_{ij} r}, \tag{5.11}$$

$$\mathbf{P}^{t+1} = \mathbf{P}^t + \frac{\beta_{ij}\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)(\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t}{\lambda - \beta_{ij} s}, \tag{5.12}$$

$$\xi_{ij}^{t+1} = \frac{\lambda s}{\lambda - s\beta_{ij}}, \tag{5.13}$$

where $r = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)$, $s = (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)$, and α_{ij}, β_{ij} are the dual variables that can be obtained analytically in Lemma 5.5.

Proof: By respectively introducing the Lagrangian multipliers $\alpha_{ij} \geq 0$ and β_{ij} for the constraints in (5.9) and (5.10), we obtain the Lagrangian of (5.8) as follows,

$$\begin{aligned} & \mathcal{L}(\mathbf{M}, \mathbf{P}, \xi_{ij}) \qquad (5.14) \\ = \; & D_{ld}(\mathbf{M}, \mathbf{M}^t) + \gamma \ell(\xi_{ij}, \xi_{ij}^t) + \lambda D_{ld}(\mathbf{P}, \mathbf{P}^t) \\ & + \alpha_{ij} \left(\mathrm{y}_{ij} d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) - \mathrm{y}_{ij} \xi_{ij} \right) + \beta_{ij} \left(\xi_{ij} - d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j) \right). \end{aligned}$$

By setting the derivatives of $\mathcal{L}$ with respect to $\mathbf{M}$ and $\mathbf{P}$ to zeros and denoting $\phi(\mathbf{M}) = -\log\left(\det(\mathbf{M})\right)$, we have,

$$\nabla\phi(\mathbf{M}) - \nabla\phi(\mathbf{M}^t) + \mathrm{y}_{ij}\alpha_{ij}\mathbf{A}_{ij} = 0, \qquad (5.15)$$

$$\lambda\nabla\phi(\mathbf{P}) - \lambda\nabla\phi(\mathbf{P}^t) - \beta_{ij}\mathbf{B}_{ij} = 0, \qquad (5.16)$$

where $\mathbf{A}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)'$, and $\mathbf{B}_{ij} = (\mathbf{z}_i - \mathbf{z}_j)(\mathbf{z}_i - \mathbf{z}_j)'$.

Given a matrix $\mathbf{M}$, we have $\frac{\partial \det(\mathbf{M})}{\partial \mathbf{M}} = \det(\mathbf{M})(\mathbf{M}^{-1})'$, which gives $\nabla\phi(\mathbf{M}) = \frac{\partial \phi(\mathbf{M})}{\partial \mathbf{M}} = -(\mathbf{M}^{-1})'$. Thus, we derive the updating rules for the solution at the $(t+1)$-th iteration from (5.15) and (5.16) as follows,

$$(\mathbf{M}^{t+1})^{-1} = (\mathbf{M}^t)^{-1} + \mathrm{y}_{ij}\alpha_{ij}\mathbf{A}_{ij}, \qquad (5.17)$$

$$\lambda(\mathbf{P}^{t+1})^{-1} = \lambda(\mathbf{P}^t)^{-1} - \beta_{ij}\mathbf{B}_{ij}, \qquad (5.18)$$

Next, we further simply the above equations by eliminating the matrix inverse operator. By using Sherman-Morrison inverse formula (*i.e.*, $(\mathbf{A}+\mathbf{u}\mathbf{v}')^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}'\mathbf{A}^{-1}}{1+\mathbf{v}'\mathbf{A}^{-1}\mathbf{u}}$) [91], we derive the equation in (5.17) as follows,

$$\begin{aligned} \mathbf{M}^{t+1} &= \left((\mathbf{M}^t)^{-1} + \mathrm{y}_{ij}\alpha_{ij}\mathbf{A}_{ij}\right)^{-1} \\ &= \left((\mathbf{M}^t)^{-1} + \mathrm{y}_{ij}\alpha_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)'\right)^{-1} \\ &= \mathbf{M}^t - \frac{\mathrm{y}_{ij}\alpha_{ij}\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t}{1 + \mathrm{y}_{ij}\alpha_{ij}(\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)}, \end{aligned} \qquad (5.19)$$

which is exactly the solution for $\mathbf{M}^{t+1}$ as in (5.11) by denoting $r = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)$.

Similarly, we apply the Sherman-Morrison inverse formula to (5.18) and we arrive at,

$$\mathbf{P}^{t+1} = \mathbf{P}^t + \frac{\beta_{ij}\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)(\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t}{\lambda - \beta_{ij}(\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)}, \qquad (5.20)$$

which is the solution for $\mathbf{P}^{t+1}$ as in (5.12) by denoting $s = (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)$.

Moreover, according to the equality constraint in (5.10), we have

$$\xi_{ij}^{t+1} = (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^{t+1}(\mathbf{z}_i - \mathbf{z}_j). \tag{5.21}$$

Substituting (5.20) into the above equation, we arrive at,

$$\begin{aligned}\xi_{ij}^{t+1} &= (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^{t+1}(\mathbf{z}_i - \mathbf{z}_j) \\ &= s + \frac{\beta_{ij}s^2}{\lambda - s\beta_{ij}} \\ &= \frac{\lambda s}{\lambda - s\beta_{ij}},\end{aligned} \tag{5.22}$$

which is exactly the solution for ξ_{ij}^{t+1} as in (5.13). Thus, we complete the proof.

5.4.3 Solutions for α_{ij} and β_{ij}

The remaining problem is to solve the two dual variables α_{ij} and β_{ij} in the updating rules in Proposition 11. Based on the KKT condition, we give the analytical solution to those two dual variables in the following Lemma 5.5.

Lemma 5.5 *The dual variables α_{ij} and β_{ij} can be obtained in closed form as follows,*

$$\alpha_{ij} = \max\left\{0, \frac{\left(\frac{\gamma}{\xi_{ij}^t} + \frac{\lambda}{s} - \frac{\lambda+\gamma}{r}\right)}{\mathrm{y}_{ij}\,(\lambda+\gamma+1)}\right\}, \tag{5.23}$$

$$\beta_{ij} = \frac{\lambda}{\lambda+\gamma}\left(\frac{\gamma}{s} - \frac{\gamma}{\xi_{ij}^t} + \mathrm{y}_{ij}\alpha_{ij}\right), \tag{5.24}$$

where $r = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)$, and $s = (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)$.

Proof: By setting the derivative of $\mathcal{L}$ in (5.14) with respect to ξ_{ij} to zero, we have,

$$\gamma\nabla\phi(\xi_{ij}) - \gamma\nabla\phi(\xi_{ij}^t) - \mathrm{y}_{ij}\alpha_{ij} + \beta_{ij} = 0. \tag{5.25}$$

Similar to the derivations of (5.17) and (5.18), we derive the solution of ξ_{ij}^{t+1} at the $(t+1)$-th iteration as follows,

$$\gamma(\xi_{ij}^{t+1})^{-1} = \gamma(\xi_{ij}^t)^{-1} - \alpha_{ij}\mathrm{y}_{ij} + \beta_{ij}. \tag{5.26}$$

Substituting (5.13) into (5.26), we arrive at,

$$\gamma\frac{\lambda - s\beta_{ij}}{\lambda s} = \frac{\gamma}{\xi_{ij}^t} - \alpha_{ij}\mathrm{y}_{ij} + \beta_{ij},$$

which further gives the solution for β_{ij} shown as in (5.24).

As α_{ij} is non-negative, the final solution for α_{ij} is either greater than or equal to zero. Specifically, according to the complementary KKT conditions, for the inequality constraints of (5.8), we have,

$$\alpha_{ij} : \begin{cases} \alpha_{ij} > 0 : \mathrm{y}_{ij}[(\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^{t+1}(\mathbf{x}_i - \mathbf{x}_j)] = \mathrm{y}_{ij}\xi_{ij}^{t+1}, \\ \alpha_{ij} = 0. \end{cases}$$

Thus, if $\alpha_{ij} > 0$, we must have $\xi_{ij}^{t+1} = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^{t+1}(\mathbf{x}_i - \mathbf{x}_j)$. Together with (5.11), we further obtain

$$\xi_{ij}^{t+1} = r - \frac{\mathrm{y}_{ij}\alpha_{ij}r^2}{1 + \mathrm{y}_{ij}\alpha_{ij}r} = \frac{r}{1 + r\mathrm{y}_{ij}\alpha_{ij}}. \tag{5.27}$$

Combining with (5.13), we eliminate ξ_{ij}^{t+1} and obtain

$$\frac{\lambda s}{\lambda - s\beta_{ij}} = \frac{r}{1 + r\mathrm{y}_{ij}\alpha_{ij}}, \tag{5.28}$$

which gives $\beta_{ij} = \frac{\lambda(r - s(1 + r\mathrm{y}_{ij}\alpha_{ij}))}{sr}$. By using (5.24), we further obtain the closed-form solution for α_{ij} as

$$\alpha_{ij} = \frac{\left(\frac{\gamma}{\xi_{ij}^t} + \frac{\lambda}{s} - \frac{\lambda+\gamma}{r}\right)}{\mathrm{y}_{ij}\left(\lambda + \gamma + 1\right)}. \tag{5.29}$$

Considering $\alpha_{ij} > 0$, thus we can obtain the closed form solution for α_{ij} as shown in (5.23). This completes the proof.

5.4.4 The Overall Optimization Procedure

The detailed optimization procedure is given as in Algorithm 6. We first initialize $t = 0$ and initialize the matrices $\mathbf{M}^0 = \mathbf{I}$ and $\mathbf{P}^0 = \mathbf{I}$, and also set $\boldsymbol{\xi}^0$ as $\xi_{ij}^0 = \begin{cases} u & (i,j) \in \mathcal{S}, \\ l & (i,j) \in \mathcal{D} \end{cases}$. Then we iteratively pick up a training pair (i, j) and update $\mathbf{M}^{t+1}$, $\mathbf{P}^{t+1}$ and ξ_{ij}^{t+1} according to Proposition 11. This process is repeated until the relative change of the norms

Algorithm 6 : Optimization procedure for ITML+

Set $t = 0$, $\mathbf{M}^0 = \mathbf{I}$, $\mathbf{P}^0 = \mathbf{I}$ and initialize $\boldsymbol{\xi}^0$.
repeat
  Pick a constraint $(i, j) \in \mathcal{S} \cup \mathcal{D}$.
  Calculate $r = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{M}^t(\mathbf{x}_i - \mathbf{x}_j)$ and $s = (\mathbf{z}_i - \mathbf{z}_j)'\mathbf{P}^t(\mathbf{z}_i - \mathbf{z}_j)$, $\forall\, t$.
  Obtain α_{ij} using (5.23) with r, s, and ξ_{ij}^t.
  Obtain β_{ij} using (5.24) with s, α_{ij} and ξ_{ij}^t.
  Update $\mathbf{M}^{t+1}$ using (5.11) with r, α_{ij} and $\mathbf{M}^t$.
  Update $\mathbf{P}^{t+1}$ using (5.12) with s, β_{ij} and $\mathbf{P}^t$.
  Calculate ξ_{ij}^{t+1} using (5.13) with s and β_{ij}.
  Set $t \leftarrow t + 1$.
until The stop criterion is reached.

of the vectors of dual variables α_{ij}'s and β_{ij}'s between two successive iterations is small than 10^{-3} or the maximum number of iterations (which is set as ten times of the number of training pairs) is reached.

Moreover, the semi-definite properties for both $\mathbf{M}$ and $\mathbf{P}$ are automatically satisfied during the updating procedure at each iteration of Algorithm 6. We also observe that all the variables have closed-form solutions at each iteration. Thus, our optimization process is efficient and shares the similar convergence property as ITML [52].

5.4.5 Solution to Partial ITML+

Similarly as in ITML+, we introduce the intermediate variables ξ_{ij}'s, and rewrite the objective function of partial ITML+ in (5.5) as follows,

$$\begin{aligned} \min_{\mathbf{M}\succeq 0, \mathbf{P}\succeq 0, \boldsymbol{\xi}} \; & D_{ld}(\mathbf{M}, \mathbf{M}^0) + \gamma L(\boldsymbol{\xi}, \boldsymbol{\xi}^0) + \lambda D_{ld}(\mathbf{P}, \mathbf{P}^0) \\ \text{s.t.,} \; & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq \xi_{ij}, (i, j) \in S, \\ & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq \xi_{ij}, (i, j) \in \mathcal{D}, \\ & \xi_{ij} = d^2_{\mathbf{P}}(\mathbf{z}_i, \mathbf{z}_j), \; (i, j) \in (\mathcal{S} - \mathcal{S}_p) \cup (\mathcal{D} - \mathcal{D}_p). \end{aligned} \tag{5.30}$$

Note for the partial ITML+ formulation in (5.30), a part of pairs are associated with the correcting function based on privileged information, while the other pairs are not associated with the correcting function. Recall when using the cyclical projection method, we update our solution by picking one training pair at each iteration. Therefore, the

subproblem at each iteration can be solved in two ways. For the training pair associated with privileged information, *i.e.*, $(i,j) \in (\mathcal{S}-\mathcal{S}_p)\cup(\mathcal{D}-\mathcal{D}_p)$, the corresponding subproblem is as the same as in (5.8), and we update the variables $\mathbf{M}$, $\mathbf{P}$ and ξ_{ij} according to Proposition 11. For the training pair without privileged information, *i.e.*, $(i,j) \in \mathcal{S}_p \cup \mathcal{D}_p$, the subproblem reduces to the same form of the subproblem in ITML [52], so we update $\mathbf{M}$ and ξ_{ij} according to the solution for the subproblem in ITML and keep $\mathbf{P}$ unchanged.

5.4.6 Computational Complexity

We now analyze the complexity of our proposed ITML+ method in Algorithm 6. In the 4-th step, the time complexity for calculating r and s are $O(h^2)$ and $O(g^2)$, respectively. The updates for α_{ij} and β_{ij} in the 5-th step and 6-th step only require $O(1)$ time complexity. In the 7-th step, the projection of $\mathbf{M}$ for each constraint requires $O(h^2)$ time complexity using the closed-form updating solution(see (5.11)), while the projection of $\mathbf{P}$ (see (5.12)) requires $O(g^2)$ time complexity in the 8-th step. As we have a total number of $|\mathcal{S}|+|\mathcal{D}|$ training pairs, the time complexity for passing the whole training pairs once is $(|\mathcal{S}|+|\mathcal{D}|)O(h^2+g^2)$. Compared with ITML, which has the time complexity of $(|\mathcal{S}|+|\mathcal{D}|)O(h^2)$ for scanning the whole training pairs once, our ITML+ is slightly more expensive, because we need to additionally optimize the distance metric $\mathbf{P}$ introduced by privileged information.

5.5 Experiments

In this section, we study the face verification and person re-identification problems in the RGB images by using weakly labeled RGB-D data and compare our proposed ITML+ algorithm with several baseline algorithms. We use two real world face datasets (*i.e.*, the EUROCOM Kinect Face dataset [90] and the CurtinFaces dataset [119]) for the face verification task, and use the BIWI RGBD-ID dataset for the person re-identification task.

5.5.1 Baseline Approaches

To the best of our knowledge, we are the first to study the face verification and person re-identification tasks in the RGB images by learning distance metric from weakly labeled

RGB-D data. As there is no existing work specifically designed for this task, we compare our ITML+ with the following baselines:

- L2 distance, the Euclidian distance without learning the distance metric (*i.e.*, $\mathbf{M} = \mathbf{I}$) is directly used in the testing stage, which is the baseline for all the distance metric learning methods;
- ITML [52], the distance metric is learnt based on the visual features from the RGB images only with side information from the training pairs;
- LMNN [204], the Large Margin Nearest Neighbor method, in which the distance metric is learnt only based on the visual features from the RGB images but with explicit label information to construct the constraints in each triplet;
- NRML [140], the Neighborhood repulsed metric learning method, which is based on the pairwise constraint. We learn the distance metric by only using the visual features from the RGB images and apply the learnt distance metric to the RGB images in the test set;
- MNRML [140], the multi-view version of NRML, which learns the distance metric from multi-view training data. We train MNRML using the visual and depth features respectively from RGB and depth images in the training set, and then apply the learnt distance metric corresponding to the visual features to the RGB images in the test set;
- PMML [49], a Pairwise-constrained Multiple Metric Learning method, which trains multiple distance metrics for multiple types of features, and the learnt distance metric corresponding to the RGB images is directly applied to the RGB images in the test set. It can reduce to ITML if only one type of feature is used;
- ITML-S [69], a two-step approach to utilize privileged information for distance metric learning, which firstly learns a distance metric by using ITML based on the depth features, and then removes some pairs that are identified as outliers. Finally, it trains a distance metric by using ITML again based on the visual features from the remaining pairs of training images.

Table 5.1: The performance evaluation for different algorithms on the EUROCOM Kinect Face dataset. Average Precision (AP) (%) as well as Area Under Curve (AUC) (%) on the test set are reported.

	L2 distance	ITML	LMNN	NRML	MNRML	PMML	ITML-S	ITML+
AP	57.70	83.54	82.93	72.74	65.46	82.92	83.52	**85.39**
AUC	69.22	92.41	92.29	82.91	80.27	92.57	92.39	**93.39**

5.5.2 Face Verification on the EUROCOM Dataset

The EUROCOM Kinect Face dataset[1] is collected by using the Microsoft Kinect, in which the subjects are captured with different facial expressions and under different lightening and occlusion conditions. There are 14 RGB-D face images (*i.e.*, 14 RGB images and 14 depth images) for each of 52 subjects (38 males, 14 females). So a total number of 728 RGB-D images are used for the experiments.

For all the face images in the dataset, we align and crop them into a fixed size of 120×105 pixels based on the positions of the two eyes. Then, each of the face image is divided into 8×7 non-overlapping sub-regions. The Gradient-LBP features [90] from each sub-region are extracted from both the RGB and depth images. Finally, the Gradient-LBP features from all the 56 sub-regions in each face image are concatenated to form a single 6888-dim feature vector. We refer to the Gradient-LBP features extracted from the RGB image and the depth image as GLBP-RGB and GLBP-DEPTH, respectively. Recall that as the face verification task in RGB images is more common, in our experiments the GLBP-RGB feature is used as the main feature, while GLBP-DEPTH feature is used as privileged information.

For evaluating our proposed ITML+ algorithm for the face verification task in RGB images, we partition the 52 subjects into a training set containing 26 subjects and a test set with the remaining 26 subjects. For our learning scenario (*i.e.*, learning using privileged information), we assume that the depth features are only available in the training data, and they are not available in the test data. A total number of 2366 positive pairs (or similar pairs) are constructed by using the samples from the same subjects in the training set, while another 7634 negative pairs (or dissimilar pairs) are sampled from the pairs constructed from different subjects in the training set. So the total number of

[1]Downloaded from http://rgb-d.eurecom.fr/

training pairs is 10000. The same strategy is utilized on the test set to generate another set of test pairs containing a total number of 10000 pairs for performance evaluation.

We compare our proposed ITML+ algorithm with the baseline method directly using L2 distance for face verification without distance metric learning, ITML [52], LMNN [204], NRML [140], ITML-S [69] as well as the multi-view distance metric learning methods MNRML [140] and PMML [49]. The ITML[2] method utilizes the GLBP-RGB features only for both the training and testing processes, while ITML-S, MNRML, PMML and our proposed ITML+ utilize both the GLBP-RGB features and the GLBP-DEPTH features for the training process, but only employ the GLBP-RGB features for the testing process. The LMNN[3] method uses GLBP-RGB features only for both the training and testing processes, but it additionally uses the explicit class label information in the training set for the construction of triplets in the algorithm. Namely, LMNN utilizes stronger label information than the other methods, which only employ side information rather than the class label information.

We set the common parameter γ for ITML, ITML-S, PMML and ITML+ in the range of $\{10^{-4}, 10^{-3.5}, 10^{-3}, \ldots, 10^{0}\}$ while the regularization parameter λ for ITML+ is set in the range of $\{10^{-2}, 10^{-1.5}, \ldots, 10^{2}\}$. Following [52], the predefined values l and u are set to be the 3-rd and 97-th percentages of the distances according to L2 distances between all pairs of samples within the training dataset. For LMNN, the tradeoff parameter is set in the range of $\{0.1, 0.2, \ldots, 1\}$, while the parameter for KNN is set in the range of $\{3, 5, \ldots, 11\}$. For NRML and MNRML, we set the p-norm parameter in their algorithm in the range of $\{1.1, 1.5, 2, 2.5, 10, 100, 1000\}$. For all the methods, in Table 5.1 we report their best results by using the optimal parameters on the test set. We perform PCA for both the GLBP-RGB features and the GLBP-DEPTH features as it is computationally inefficient to learn the distance metric with the original feature dimension. We fix the PCA dimension for both GLBP-RGB and GLBP-DEPTH features to be 150 in our experiments. Average Precision (AP) and Area Under Curve (AUC) are reported for performance evaluation.

The detailed experimental results are shown as in Table 5.1. From the experimental results, we observe that ITML and LMNN outperform the original L2 distance in terms

[2]Codes are from http://www.cs.utexas.edu/ pjain/itml/
[3]Codes are from http://www.cse.wustl.edu/ kilian/code/page21/page21.html

of both AP and AUC, which demonstrates that it is useful to learn the distance metrics for the face verification problem. Although stronger class label information is used for LMNN, it is worse than ITML, which shows ITML is more suitable for face verification task. Moreover, our ITML+ is much better than ITML that learns the distance metric only using the GLBP-RGB features, which demonstrates it is beneficial to use the depth features GLBP-DEPTH as privileged information to learn a more robust distance metric for the face verification task in RGB images.

Note that the recently proposed work ITML-S in [174] uses a two-step approach to utilize privileged information. Specifically, in the first stage, a distance metric is learnt by using the ITML algorithm based on privileged information (*i.e.*, the GLBP-DEPTH feature). Then, the training pairs are sorted based on the distances in the learnt distance metric space, and some pairs are removed from the training set. In the second stage, another distance metric is trained based on GLBP-RGB feature only. Note that ITML-S is slightly worse than ITML. A possible explanation is the two stage approach based on the pair removal strategy is not so effective to utilize privileged information. Therefore it is critical to better utilize privileged information. In contrast, our ITML+ algorithm learns the correcting distance metric and the decision distance metric in a unified framework, and it directly models the relationship between the main feature GLBP-RGB from RGB images and the privileged feature GLBP-DEPTH from depth images, thus it is much more effective than the naive two-step approach in [174].

We also compare ITML+ with the two multi-view methods PMML [49] and MNRML [140], which are initially proposed to fuse multiple views of features in the training set for the distance metric learning. The results in Table 5.1 show that they are only comparable or even worse when compared with their single-view counterparts ITML and NRML [140], respectively. A possible explanation is that the final goal of these methods is to learn good distance metrics when different types of features are available in both the training and test sets. Although we can still obtain a distance metric corresponding to the visual features from RGB images, this distance metric cannot work well when the depth features are not available in the test set as in our task.

Table 5.2: The performance evaluation for different algorithms on the CurtinFaces dataset. Average Precision (AP) (%) as well as the Area Under Curve (AUC) (%) on the test set are reported.

	L2 distance	ITML	LMNN	NRML	MNRML	PMML	ITML-S	ITML+
AP	61.02	73.14	70.01	65.81	62.70	71.18	67.09	**76.86**
AUC	58.18	72.14	68.29	63.76	59.74	70.26	69.14	**77.50**

5.5.3 Face Verification on the CurtinFaces Dataset

We also conduct the experiments on the CurtinFaces dataset[4], in which the RGB-D images are also collected by using the Microsoft Kinect. The CurtinFaces dataset consists of 52 people, and each people contains 95 RGB-D face images, thus we have a total number of 4940 RGB-D face images in the dataset. The images are captured with different facial expressions and under different illuminations and poses.

Again, we use the samples from the first 26 subjects as the training set, and the remaining 26 subjects as the test set. On the training set, we randomly generate a total number 15000 similar pairs that contain the faces from the same subject as well as another 15000 dissimilar pairs that contain the faces from different subjects. The same strategy is also utilized on the test set to generate another 30000 pairs including 15000 similar pairs and 15000 dissimilar pairs for performance evaluation. For all the face images in the training dataset, we use the same strategy as in the EUROCOM dataset to extract 6888-dim visual features and 6888-dim depth features from RGB images and depth images, respectively.

The same baselines and parameter settings are used to evaluate the performances of our proposed ITML+ algorithm. We fix the PCA dimensions of both the GLBP-DEPTH feature the GLBP-RGB feature to be 150, and we report the APs and AUCs of different algorithms in Table 5.2. On this dataset, all the distance learning methods are better than the L2 distance and ITML is still better than LMNN and NRML. We can observe from Table 5.2 that ITML+ achieves the best results and it also outperforms ITML, which demonstrates that it is beneficial to utilize extra privileged information on the training dataset to improve the distance metric learning results for the face verification task in the RGB images. Our ITML+ again outperforms the two-step approach ITML-S

[4]http://impca.curtin.edu.au/downloads/datasets.cfm

Table 5.3: The performance evaluation for different algorithms on the BIWI RGBD-ID dataset. The Rank-1 recognition rates (%) on the two test sets are reported.

	L2 distance	ITML	LMNN	NRML	MNRML	PMML	ITML-S	ITML+
Walking	34.59	47.18	36.28	37.22	34.41	41.54	41.73	**50.38**
Still	85.53	91.92	85.71	87.97	87.97	90.04	91.92	**95.49**

as well as PMML and MNRML in terms of both AP and AUC, which demonstrates the effectiveness of our proposed ITML+ method for utilizing privileged information in a unified framework.

5.5.4 Person Re-identification on the BIWI RGBD-ID Dataset

In this section, we conduct the experiments on the BIWI RGBD-ID dataset[5] for the person re-identification task in RGB images. The BIWI RGBD-ID dataset [150, 151] was also collected by using the Microsoft Kinect, and the dataset consists of a training set and two testing sets (*i.e.*, "Walking" and "Still"). The training set records the video sequences of 50 different subjects performing certain actions (*e.g.*, rotation, head movements, walking) in front of a Kinect sensor. The test set is collected from 28 subjects that appears in the training set, but on a different day and with a different dress. In the "Walking" setting, each of the 28 subjects performs the action walking in front of the Kinect, while in the "Still" setting, all subjects stand still in front of the Kinect with little movement. Both the RGB images and the depth images are recorded simultaneously. In order to perform the person re-identification task, for each subject in the training set we uniformly sample 20 shots of RGB-D images from each of the video sequences, and sample 20 shots of RGB images from each of the video sequences for each subject in the testing set. The whole head and human body appear in both the RGB and depth images are cropped and the background is removed. Thus, we obtain a total number of 1000 RGB-D images in the training set, and a total number of 560 RGB images in the two test sets.

After that, we extracted the RGB-D kernel descriptors (KDES) [17] as the features, which have shown promising results for a broad range of applications using the RGB-D images [17, 18]. Following [17], the Gradient KDES features are extracted from each

[5]http://robotics.dei.unipd.it/reid/index.php/downloads

of the RGB/depth images by using the codes[6] from [17]. Then the extracted kernel descriptors are aggregated into the object-level features, in which the codebook size is set to be 1000, and three levels of pyramids (*i.e.*, 1×1, 2×2 and 4×4 for RGB images and 1×1, 2×2 and 3×3 for depth images) are utilized for the spatial pooling. Finally, the feature vectors from each region of the pyramids are concatenated into a single feature vector (21,000-dim for RGB images and 14,000-dim for depth images). We extract the KDES features from both the RGB images and the depth images in the training set, and we only extract the KDES features from the RGB images in the test data set.

Before learning the distance metric, we also perform PCA on both the RGB features and the depth features to reduce the feature dimension to be 150, as in the experiments for face verification. On the training set, we construct a set of 9500 similar pairs, and we also sample a set of 9500 dissimilar pairs. We also use the features from the RGB images as the main features, and the features from the depth images as privileged information. The Rank-1 recognition rate is the typical evaluation criterion, which is the first point in the so-called Cumulative Matching Characteristic (CMC) curve [78], and it measures the mean person recognition rate when finding the correct person in the top-1 match. We train all the models of all the algorithms on the training set, and then apply the learnt distance metrics on the test set, and report the Rank-1 recognition rate on the two test settings in Table 5.3, respectively.

From the experimental results in Table 5.3, we observe that in terms of the Rank-1 recognition rate, all the distance metric learning algorithms are better than the baseline method (*i.e.*, L2 distance). Our proposed ITML+ is much better than ITML as well as other baseline methods, which again shows the effectiveness of our proposed ITML+ to utilize additional depth information in the training set. The two methods PMML and MNRML learn a unified decision function for fusing the distances from multiple views, the results are only comparable or even worse than their single-view counterparts (*i.e.*, ITML and NRML). A possible explanation is that these methods aim to learn good distance metrics when all types of features are available during both the training and testing processes. Although we can still obtain a distance metric corresponding to the visual features from the RGB images, this distance metric cannot work well in our task due to

[6]http://mobilerobotics.cs.washington.edu/projects/kdes/

the lack of the depth features in the test set. We observe that the recognition rates for the "Still" case are much better than those for the "Walking" case, which demonstrates that it is more difficult to perform the person re-identification task for the "Walking" case, because there are more variations in the test set when people are walking.

5.5.5 Detailed Performance Analysis

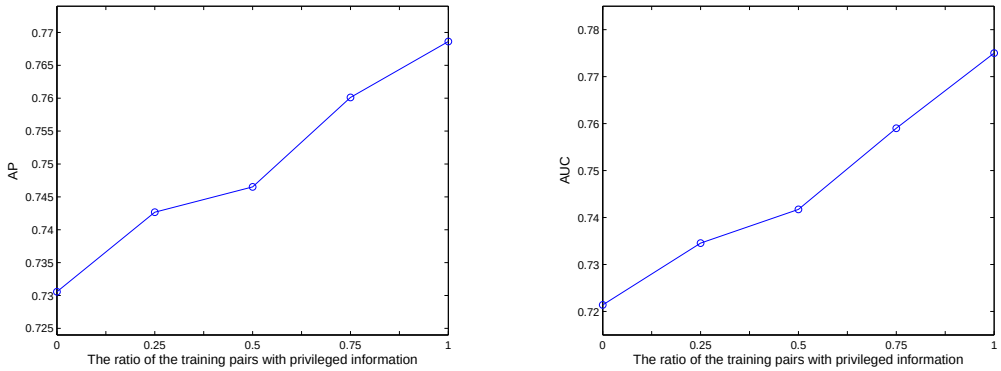
In this section, we conduct the experiments to analyze our proposed ITML+ algorithm. We firstly investigate partial ITML+ by using different percentages of training pairs with privileged information and then analyze the learnt distance metrics.

5.5.5.1 Evaluating partial ITML+ using different percentages of pairs with privileged information

In real world applications, privileged information may be hard to be obtained. So it is also possible that some training data are not associated with privileged information. We evaluate our partial ITML+ discussed in Section 5.3.4 by using different percentages of training pairs with privileged information.

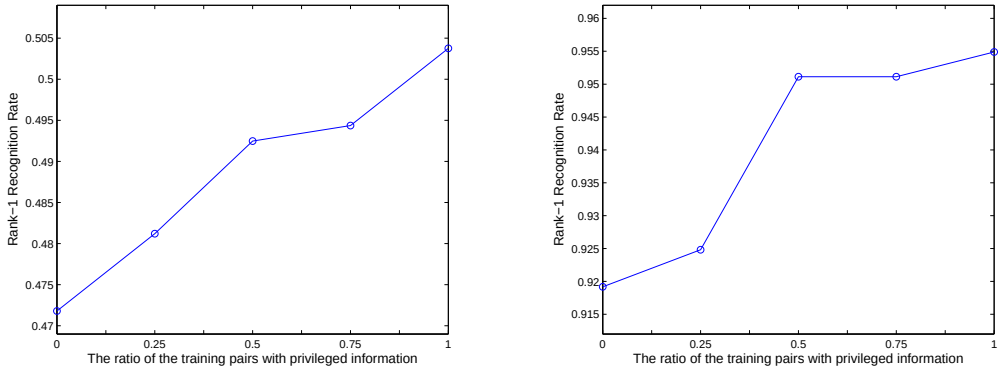
We take the CurtinFaces and BIWI RGBD-ID datasets as two examples, and use the partial ITML+ formulation to learn the distance metric by varying the percentage of the pairs with privileged information. We use the first 0%, 25%, 50%, 75%, and 100% of positive training pairs and negative training pairs with privileged information (*i.e.*, the GLBP-DEPTH features for CurtinFaces dataset and the KDES-DEPTH features for the BIWI RGBD-ID dataset) and the remaining 100%, 75%, 50%, 25% and 0% training samples are not with privileged information. Then we train our partial ITML+ model to learn a distance metric on the main features, which is used on the testing set for performance evaluation.

We report AP and AUC on the CurtinFaces dataset in Fig. 5.2 (a) and Fig. 5.2 (b), respectively. We also report the Rank-1 recognition rate on the BIWI RGBD-ID dataset for two test settings "Walking" and "Still" in Fig. 5.3 (a) and Fig. 5.3 (b), respectively. We can observe that the results are the same with those of ITML (*resp.*, ITML+) when the ratio is set to 0% (*resp.*, 100%). Note our partial ITML+ incorporates ITML and ITML+ as two special cases according to the formulation in (5.5). By varying the ratio



(a) AP (b) AUC

Figure 5.2: The results using different ratios of training pairs with privileged information on the CurtinFaces dataset.



(a) Rank-1 recognition rate for the test set "Walking"

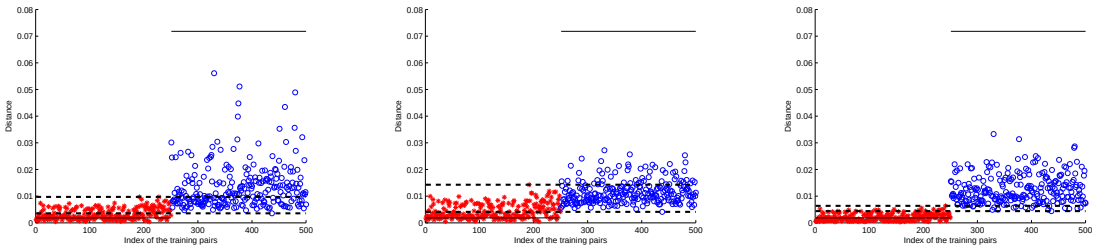
(b) Rank-1 recognition rate for the test set "Still"

Figure 5.3: The results using different ratios of training pairs with privileged information on the BIWI RGBD-ID dataset.

in the range of {0%, 25%, 50%, 75%, 100%}, we observe that the performances are improved when the ratio of training pairs with privileged information increases.

5.5.5.2 Analyzing the learnt distance metric

We take the BIWI RGBD-ID dataset as an example to analyze the learnt distance metric. Specifically, we analyze the distance metrics learnt by using ITML, ITML-S and ITML+ for classifying the first 250 positive training pairs as well as the first 250 negative training pairs in the following.



(a) ITML (b) ITML-S (c) ITML+

Figure 5.4: Illustration of the distances between 250 positive pairs of images and 250 negative pairs of images based on the distance metrics learnt by using ITML, ITML-S and our ITML+. The red star indicates the positive pair while the blue circle indicates the negative pair.

Note the KEDS-RGB feature is used as the main feature in the testing processes. We show the distances of these 500 pairs of RGB images based on the learnt distance metrics from ITML, ITML-S and ITML+ in Fig. 5.4(a), Fig. 5.4(b) and Fig. 5.4(c), respectively. In the two figures, the red star indicates the positive pair, while the blue circle indicates the negative pair. The two horizontal lines are the predefined parameters l (*i.e.*, $l = 1.9 \times 10^{-3}$) and u (*i.e.*, $u = 7.2 \times 10^{-2}$). As shown in Fig. 5.4(c), we can observe that the distances based on the learnt metrics by using the ITML+ algorithm for both the positive pairs and the negative pairs are better clustered when compared with the results obtained by using ITML as shown in Fig. 5.4(a) as well as using ITML-S as shown in Fig. 5.4(b). Moreover, the projections of the points along the y-axis in Fig. 5.4(c) seldomly have the overlaps (*i.e.*, the area between the two dashed lines) between the positive pairs and negative pairs, which is much better when compared with the results from ITML. As the ITML-S removes some training pairs, it is reasonable to observe that the data points are not even better classified than that of for ITML. Thus, we conclude that the distance metric learnt by using ITML+ is better, which demonstrates the effectiveness of our ITML+.

5.6 Summary

In this chapter, we have studied the face verification and person re-identification tasks in the RGB images by using the weakly labeled RGB-D data. We formulate a new

problem called distance metric learning with privileged information, where the distance metric is learnt with extra information which is available only in the training data but unavailable in the test data. We take the Information-theoretic Metric Learning (ITML) method as an example, and propose a new method called Information-theoretic Metric Learning with Privileged Information (ITML+) for distance metric learning by additionally using privileged information. An efficient cyclical projection method based on analytical solutions for updating all the variables is also developed to solve the new objective function in our proposed ITML+. The extensive experiments are conducted on the real-world EUROCOM, CurtinFaces and BIWI RGBD-ID datasets. The results demonstrate the effectiveness of our newly proposed ITML+ algorithm for learning the effective distance metric from weakly labeled RGB-D data for the face verification and person re-identification tasks in the RGB images.

Chapter 6

Conclusion and Future Work

With the development of more effective visual representations for computer vision tasks, the learning with multiple representations will receive increasing attention in the future. In this thesis, we have proposed several novel algorithms to the learning with multiple representations, and we also apply the proposed algorithms to a few computer vision applications. In this part, we conclude our proposed works and discuss the future work.

6.1 Conclusion

We conclude this thesis by summarizing the contributions for learning with multiple representation as follows:

- We have proposed a novel Soft Margin framework for Multiple Kernel Learning (SMMKL) based on the novel kernel slack variables introduced base kernels. Based on the hard margin perspective for traditional ℓ_1MKL, we then propose the hinge loss soft margin MKL, the square hinge loss soft margin MKL and the square loss soft margin MKL. The hinge loss soft margin MKL leads to a novel box constraint for MKL, while square hinge loss soft margin MKL and square loss soft margin MKL unifies the family of elastic-net MKL and the ℓ_2MKL from literature, respectively. We discover their connections with previous MKL methods and compare different MKL formulations in terms of the robustness of these different loss functions defined on the kernel slack variables. Comprehensive experiments have been conducted on the benchmark data sets, the YouTube and Event6 data sets from computer

vision applications. The experimental results demonstrate the effectiveness of our proposed framework.

- We have also proposed an Input-Output Kernel Learning (IOKL) framework for handling general data ambiguities with multiple representations. By introducing the concept of *input-output kernel*, the methodology from traditional MKLs designed for supervised learning only is applicable for handling general data ambiguity problems such as SSL, MIL and clustering with multiple data representations in a unified framework. To learn a more robust classifier, we further introduce a novel soft margin group sparse MKL formulation. In addition, a block-wise coordinate descent algorithm with an analytical solution for the kernel coefficients is developed to solve the new MKL formulation efficiently. The promising experimental results on the challenging NUS-WIDE dataset for a computer vision application (i.e., text-based image retrieval), SSL benchmark datasets and MIL benchmark datasets demonstrate the effectiveness of our proposed IOKL framework.

- We propose a new problem called distance metric learning with privileged information, where the distance metric is learnt with extra information which is available only in the training data but unavailable in the test data. We propose a novel method called Information-theoretic Metric Learning with Privileged Information (ITML+) for distance metric learning by additionally using the privileged information for the training data. An efficient cyclical projection method based on analytical solutions for all the variables is also developed to solve the new objective function. The extensive experiments are conducted on the real-world EUROCOM, CurtinFaces and BIWI RGBD-ID datasets. The results demonstrate the effectiveness of our newly proposed ITML+ algorithm for learning more effective distance metric from RGB-D data for face verification and person re-identification tasks in the RGB images.

6.2 Future Work

In this section, we discuss the possible extensions to our works.

6.2.1 Future Work for Soft Margin Multiple Kernel Learning

Our proposed Soft Margin Multiple Kernel Learning framework has provided a new perspective to multiple kernel learning problem. The generalization bound of our SMMKL framework could be further investigated for the completeness of theoretical analyse of our work. Moreover, the tool box with efficient implementations of the different types of MKL algorithms will be released to public as the proposed work makes it possible to fuse as many different types of visual representations as possible for visual classification tasks. It is possible to study the fusion of a large number of visual representations such as low-level handcrafted features, classifier-based features and even the deep representations from deep convolutional networks for more challenging real-world computer vision classification tasks.

6.2.2 Future Work for Input-output Kernel Learning

Our proposed Input-output Kernel Learning is applicable to more learning scenarios such as clustering [133], relative outlier detection [123], and multi-instance semi-supervised learning. It is also possible to apply our proposed group sparse soft margin regularization to the task of domain adaptation [34]. It is also an interesting topic to improve the speed of IOKL framework. Besides, more effective label inference procedure could be further explored. The different regularization strategies such as the multi-layer structure, hierarchical structure from the construction of the base input-output kernels could be explored to improve specific learning tasks.

6.2.3 Future Work for Learning with Privileged Information

The Learning with Privileged Information framework opens a wide area for both the machine learning algorithms and computer vision applications. Almost all the existing learning algorithms can be extended to this learning scenario. For the different learning scenarios, it is possible to explore the specific learning algorithms for utilizing the privileged information. For example, multiple kernel learning using privileged information could be further studied. Extending the learning setting to algorithms tackling weakly labeled data with either single representation or multiple representations is also an important direction. From the application point of view, the designing of the privileged

information could also be investigated for computer vision applications. For instance, the additional text description for the images from web could be obtained to help the traditional visual recognition tasks. It is also quite interesting to explore the domain adaptation problem using privileged information where source data not only have a different distribution with target data but also have additional privileged information.

References

[1] Steven P. Abney. Bootstrapping. In *ACL*, 2002.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[3] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[4] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950.

[5] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[6] Francis R. Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[7] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, 2004.

[8] Francis R. Bach, Romain Thibaux, and Michael I. Jordan. Computing regularization paths for learning multiple kernels. In *NIPS*, 2004.

[9] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[10] Slawomir Bak, Etienne Corvée, François Brémond, and Monique Thonnat. Person re-identification using spatial covariance regions of human body parts. In *AVSS*, 2010.

[11] Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2004.

[12] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[13] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.

[14] Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf, and Gunnar Rätsch. Support vector machines and kernels for computational biology. *PLoS Comput Biology*, 4(10), October 2008.

[15] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[16] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[17] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. In *NIPS*, 2010.

[18] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, 2011.

[19] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, 1992.

[20] U. Brefeld and T. Scheffer. Co-EM support vector learning. In *ICML*, 2004.

[21] Lev M. Bregman. Parallel optimization: Theory, algorithms, and applications. *Oxford University Press*, 1997.

[22] Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1354–1369, 2014.

[23] Razvan C. Bunescu and Raymond J. Mooney. Multiple instance learning for sparse positive bags. In *ICML*, 2007.

[24] Feng Cai and Vladimir Cherkassky. Generalized smo algorithm for svm-based multitask learning. *IEEE Transactions on Neural Networks and Learning Systems*, 23(6):997–1003, 2012.

[25] Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *ICCV*, 2013.

[26] Zhimin Cao, Qi Yin, Xiaoou Tang, and Jian Sun. Face recognition with learning-based descriptor. In *CVPR*, pages 2707–2714, 2010.

[27] Chih-Chung Chang and Chih-Jen Lin. Training nu-support vector classifiers: Theory and algorithms. *Neural Computation*, 13(9):2119–2147, 2001.

[28] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[29] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.

[30] Olivier Chapelle, Vikas Sindhwani, and S. Sathiya Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

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

[32] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. Society for Artificial Intelligence and Statistics, 2005.

[33] Lin Chen, Lixin Duan, Ivor W. Tsang, and Dong Xu. Efficient discriminative learning of class hierarchy for many class prediction. In *ACCV*, 2012.

[34] Lin Chen, Lixin Duan, and Dong Xu. Event recognition in videos by learning from heterogeneous web sources. In *CVPR*, 2013.

[35] Lin Chen, Wen Li, and Dong Xu. Recognizing rgb images by learning from rgb-d data. In *CVPR*, 2014.

[36] Lin Chen, Ivor W. Tsang, and Dong Xu. Laplacian embedded regression for scalable manifold regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 23(6):902–915, 2012.

[37] Lin Chen, Dong Xu, Ivor W. Tsang, and Jiebo Luo. Tag-based image retrieval improved by augmented features and group-based refinement. *IEEE Transactions on Multimedia*, 14(4):1057–1067, Auguest 2012.

[38] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[39] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao. Zheng. NUS-WIDE: A real-world web image database from national university of singapore. In *CIVR*, 2009.

[40] Ronan Collobert, Fabian H. Sinz, Jason Weston, and Léon Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

[41] Corinna Cortes, Marius Kloft, and Mehryar Mohri. Learning kernels using local rademacher complexity. In *NIPS*, 2013.

[42] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. In *UAI*, 2009.

[43] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *ICML*, 2010.

[44] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. In *ICML*, 2010.

[45] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[46] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[47] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[48] Nello Cristianini, John Shawe-Taylor, Andr Elisseeff, and Jaz S. Kandola. On kernel-target alignment. In *NIPS*, 2001.

[49] Zhen Cui, Wen Li, Dong Xu, Shiguang Shan, and Xilin Chen. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *CVPR*, 2013.

[50] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[51] Sanjoy Dasgupta, Michael L. Littman, and David McAllester. Pac generalization bounds for co-training. In *NIPS*, 2002.

[52] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[54] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[55] Michele Stoppa Loris Bazzani Dong Seon Cheng, Marco Cristani and Vittorio Murino. Custom pictorial structures for re-identification. In *BMVC*, 2011.

[56] Lixin Duan, Wen Li, Ivor Wai-Hung Tsang, and Dong Xu. Improving web image search by bag-based reranking. *IEEE Transactions on Image Processing*, 20(11):3280–3290, 2011.

[57] Lixin Duan, Ivor W. Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[58] Lixin Duan, Dong Xu, Ivor W. Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. In *CVPR*, 2010.

[59] Lixin Duan, Dong Xu, Ivor Wai-Hung Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2012.

[60] Lixin Duan, Yanwu Xu, Wen Li, Lin Chen, Damon Wing Kee Wong, Tien Yin Wong, and Jiang Liu. Incorporating privileged genetic information for fundus image based glaucoma detection. In *MICCAI*, 2014.

[61] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *ICML*, 2008.

[62] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[63] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[64] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010.

[65] Jason D. R. Farquhar, Hongying Meng, Sandor Szedmak, David R. Hardoon, and John Shawe-taylor. Two view learning: Svm-2k, theory and practice. In *ADVANCES in Neural Information Processing Systems*. MIT Press, 2006.

[66] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, April 2007.

[67] 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*, 32(9):1627–1645, 2010.

[68] Jan Feyereisl and Uwe Aickelin. Privileged information for data clustering. *Information Science*, 194:4–23, 2012.

[69] Shereen Fouad, Peter Tino, Somak Raychaudhury, and Petra Schneider. Incorporating privileged information through metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 24(7):1086–1098, 2013.

[70] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):92–104, 2013.

[71] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Sparse representation with kernels. *IEEE Transactions on Image Processing*, 22(2):423–434, 2013.

[72] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia, and Peilin Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *CVPR*, 2010.

[73] Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-instance kernels. In *ICML*, 2002.

[74] Dries Geebelen, Johan A. K. Suykens, and Joos Vandewalle. Reducing the number of support vectors of svm classifiers using the smoothed separable case approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):682–688, 2012.

[75] Peter V. Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

[76] Niloofar Gheissari, Thomas B. Sebastian, and Richard Hartley. Person reidentification using spatiotemporal appearance. In *CVPR*, 2006.

[77] S. Gong, M. Cristani, S. Yan, and C.C. (Eds.) Loy. *Person Re-identification*. Springer, 2014.

[78] Doug Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *IEEE International Workshop on Performance Evaluation for Tracking and Surveillance, Rio de Janeiro*, 2007.

[79] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008.

[80] Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009.

[81] David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, December 2004.

[82] Martin Hirzer, Csaba Beleznai, Peter M. Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *SCIA*, 2011.

[83] Martin Hirzer, Peter M. Roth, Martin Köstinger, and Horst Bischof. Relaxed pairwise learned metric for person re-identification. In *ECCV*, 2012.

[84] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 13(2):415–425, March 2002.

[85] Mingqing Hu, Yiqiang Chen, and James T. Kwok. Building sparse multiple-kernel svm classifiers. *IEEE Transactions on Neural Networks*, 20(5):827–839, 2009.

[86] Sujun Hua and Zhirong Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001.

[87] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments.

[88] Zakria Hussain and John Shawe-Taylor. Improved loss bounds for multiple kernel learning. In *AISTATS*, 2011.

[89] Zakria Hussain and John Shawe-Taylor. A note on improved loss bounds for multiple kernel learning. *CoRR*, abs/1106.6258, 2011.

[90] Tri Huynh, Rui Min, and Jean-Luc Dugelay. An efficient lbp-based descriptor for facial depth images applied to gender recognition using rgb-d face data. In *ACCV Workshops*, 2012.

[91] Sherman Jack and Morrison Winifred. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Annals of Mathematical Statistics*, 20, 1949.

[92] Shi Jianbo and Malik Jitendra. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[93] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel P. W. Ellis, and Alexander C. Loui. Consumer video understanding: a benchmark database and an evaluation of human and machine performance. In *ICMR*, 2011.

[94] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML*, 1998.

[95] Thorsten Joachims. Advances in kernel methods. chapter Making Large-scale Support Vector Machine Learning Practical. MIT Press, 1999.

[96] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[97] Thorsten Joachims. Training linear svms in linear time. In *SIGKDD*, 2006.

[98] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[99] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_{p}-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

[100] Marius Kloft and Gilles Blanchard. The local rademacher complexity of lp-norm multiple kernel learning. In *NIPS*, 2011.

[101] Marius Kloft and Gilles Blanchard. On the convergence rate of ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research*, 13:2465–2502, 2012.

[102] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, Pavel Laskov, Klaus-Robert Müller, and Alexander Zien. Efficient and accurate lp-norm multiple kernel learning. In *NIPS*, 2009.

[103] Marius Kloft, Ulrich Rückert, and Peter L. Bartlett. A unifying view of multiple kernel learning. In *ECML/PKDD*, 2010.

[104] Martin Köstinger, Martin Hirzer, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.

[105] Balaji Krishnapuram, David Williams, Ya Xue, Alex Hartemink, Lawrence Carin, and M. Figueiredo. On semi-supervised classification. In *NIPS*, 2004.

[106] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[107] Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013.

[108] Brian Kulis, Mátyás A. Sustik, and Inderjit S. Dhillon. Learning low-rank kernel matrices. In *ICML*, 2006.

[109] Abhishek Kumar and Hal Daumé III. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.

[110] Abhishek Kumar, Piyush Rai, and Hal Daumé III. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.

[111] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[112] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011.

[113] Kuan-Ting Lai, Felix X. Yu, Ming-Syan Chen, and Shih-Fu Chang. Video event detection by inferring temporal instance labels. In *CVPR*, 2014.

[114] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semi-definite programming. In *ICML*, 2002.

[115] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[116] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[117] Ryan Layne, Timothy M. Hospedales, and Shaogang Gong. Person re-identification by attributes. In *BMVC*, 2012.

[118] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.

[119] Billy Y. L. Li, Ajmal S. Mian, Wanquan Liu, and Aneesh Krishna. Using kinect for face recognition under varying poses, expressions, illumination and disguise. In *WACV*, 2013.

[120] Bing Li, Weihua Xiong, and Weiming Hu. Context-aware multi-instance learning based on hierarchical sparse representation. In *ICDM*, 2011.

[121] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.

[122] Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 37:1088–1098, 2007.

[123] Shukai Li and Ivor W. Tsang. Learning to locate relative outliers. *ACML*, 20, 2011.

[124] Shukai Li and Ivor W. Tsang. Maximum margin/volume outlier detection. In *ICTAI*, 2011.

[125] Wen Li, Lixin Duan, Ivor Wai-Hung Tsang, and Dong Xu. Batch mode adaptive multiple instance learning for computer vision tasks. In *CVPR*, 2012.

[126] Wen Li, Lixin Duan, Ivor Wai-Hung Tsang, and Dong Xu. Co-labeling: A new multi-view learning approach for ambiguous problems. In *ICDM*, 2012.

[127] Wen Li, Lixin Duan, Dong Xu, and Ivor W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, 2014.

[128] Wen Li, Lixin Duan, Dong Xu, and Ivor Wai-Hung Tsang. Text-based image retrieval using progressive multi-instance learning. In *ICCV*, 2011.

[129] Wen Li, Li Niu, and Dong Xu. Exploiting privileged information from web data for image categorization. In *ECCV*, 2014.

[130] Yu-Feng Li, James T. Kwok, Ivor W. Tsang, and Zhi-Hua Zhou. A convex method for locating regions of interest with multi-instance learning. In *ECML/PKDD (2)*, 2009.

[131] Yu-Feng Li, James T. Kwok, and Zhi-Hua Zhou. Semi-supervised learning using label mean. In *ICML*, 2009.

[132] Yu-Feng Li, Ivor W. Tsang, James T. Kwok, and Zhi-Hua Zhou. Convex and scalable weakly labeled svms. *Journal of Machine Learning Research*, 14:1391–1445, 2013.

[133] Yu-Feng Li, Ivor W. Tsang, James Tin-Yau Kwok, and Zhi-Hua Zhou. Tighter and convex maximum margin clustering. In *AISTATS*, 2009.

[134] C.-J. Lin. A formal analysis of stopping criteria of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 13(5):1045–1052, 2002.

[135] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh. Multiple kernel learning for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1147–1160, 2011.

[136] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009.

[137] Li Liu and Ling Shao. Learning discriminative representations from RGB-D video data. In *IJCAI*, 2013.

[138] Chris Longworth and Mark J. F. Gales. Combining derivative and parametric kernels for speaker verification. *IEEE Transactions on Audio, Speech & Language Processing*, 17(4):748–757, 2009.

[139] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[140] Jiwen Lu, Xiuzhuang Zhou, Yap-Peng Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):331–345, 2014.

[141] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

[142] Qi Mao and Ivor Wai-Hung Tsang. Efficient multitemplate learning for structured prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2):248–261, 2013.

[143] Qi Mao and Ivor Wai-Hung Tsang. A feature selection method for multivariate performance measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2051–2063, 2013.

[144] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, 1998.

[145] Brian McFee, Carolina Galleguillos, and Gert R. G. Lanckriet. Contextual object localization with multiple kernel nearest neighbor. *IEEE Transactions on Image Processing*, 20(2):570–585, 2011.

[146] Brian McFee and Gert R. G. Lanckriet. Learning multi-modal similarity. *Journal of Machine Learning Research*, 12:491–523, 2011.

[147] Stefano Melacci and Mikhail Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12:1149–1184, 2011.

[148] Charles A. Micchelli, Jean Morales, and Massimiliano Pontil. A family of penalty functions for structured sparsity. In *NIPS*, 2010.

[149] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[150] Matteo Munaro, Andrea Fossati, Alberto Basso, and Emanuele Menegatti Luc Van Gool. One-shot person re-identification with a consumer depth camera. *Book Chapter in "Person Re-Identification"*, 2014.

[151] Matteo Munaro, Tal Hassner, and Yaniv Taigman. 3d reconstruction of freely moving persons for re-identification with a depth sensor. In *ICRA*, 2014.

[152] Hieu V. Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *ACCV*, 2010.

[153] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, 2000.

[154] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001.

[155] Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[156] Francesco Orabona and Jie Luo. Ultra-fast optimization algorithm for sparse multi kernel learning. In *ICML*, 2011.

[157] Francesco Orabona, Jie Luo, and Barbara Caputo. Online-batch strongly convex multi kernel learning. In *CVPR*, 2010.

[158] Francesco Orabona, Jie Luo, and Barbara Caputo. Multi kernel learning with online-batch optimization. *Journal of Machine Learning Research*, 13:227–253, 2012.

[159] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, 2011.

[160] Dmitry Pechyony, Rauf Izmailov, Akshay Vashist, and Vladimir Vapnik. Smo-style algorithms for learning using privileged information. In *DMIN*, 2010.

[161] Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. In *NIPS*, 2010.

[162] John C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[163] Bryan Prosser, Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by support vector ranking. In *BMVC*, 2010.

[164] Alain Rakotomamonjy, Francis R. Bach, Stphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2512, 2008.

[165] Klaus robert Mller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12:181–201, 2001.

[166] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[167] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[168] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[169] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[170] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[171] Chun-Wei Seah, Ivor Wai-Hung Tsang, and Yew-Soon Ong. Healing sample selection bias by source classifier selection. In *ICDM*, 2011.

[172] Shai Shalev-Shwartz and Yoram Singer. Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, 7:1567–1599, 2006.

[173] Ling Shao, Li Liu, and Xuelong Li. Feature learning for image classification via multiobjective genetic programming. *IEEE Transactions on Neural Network and Learning System*, 25(7):1359–1371, 2014.

[174] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to rank using privileged information. In *ICCV*, 2013.

[175] John Shawe-Taylor. Kernel learning for novelty detection. In *NIPS 2008 Workshop Kernel Learning: Automatic Selection of Optimal Kernels*.

[176] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.

[177] V. Sindhwani and D. Rosenberg. An rkhs for multi-view learning and manifold co-regularization. In *ICML*, 2008.

[178] Vikas Sindhwani and Partha Niyogi. A co-regularized approach to semi-supervised learning with multiple views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.

[179] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[180] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[181] Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio De Bona, Alexander Binder, Christian Gehl, and Vojtech Franc. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 11:1799–1802, 2010.

[182] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[183] Nathan Srebro and Shai Ben-David. Learning bounds for support vector machines with learned kernels. In *COLT*, 2006.

[184] Taiji Suzuki and Masashi Sugiyama. Fast learning rate of multiple kernel learning: Trade-off between sparsity and smoothness. In *AISTATS*, 2012.

[185] Taiji Suzuki and Ryota Tomioka. Spicymkl: a fast algorithm for multiple kernel learning with thousands of kernels. *Machine Learning*, 85(1-2):77–108, 2011.

[186] Marie Szafranski, Yves Grandvalet, and Alain Rakotomamonjy. Composite kernel learning. *Machine Learning*, 79(1-2):73–103, 2010.

[187] Mingkui Tan, Ivor W. Tsang, and Li Wang. Towards ultrahigh dimensional feature selection for big data. *Journal of Machine Learning Research*, 15(1):1371–1429, 2014.

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

[189] ROBERT Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[190] Ivor W. Tsang and James T. Kwok. Efficient hyperkernel learning using second-order cone programming. *IEEE Transactions on Neural Networks*, 17(1):48–58, 2006.

[191] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, December 2005.

[192] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

[193] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[194] Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In *ICML*, 2009.

[195] Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[196] S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, and M. Varma. Multiple kernel learning and the smo algorithm. In *NIPS*, 2010.

[197] Ngoc-Son Vu and Alice Caplier. Enhanced patterns of oriented edge magnitudes for face recognition and image matching. *IEEE Transactions on Image Processing*, 21(3):1352–1365, 2012.

[198] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[199] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas S. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[200] Wei Wang and Zhi-Hua Zhou. Analyzing co-training style algorithms. In *ECML*, pages 454–465, 2007.

[201] Wei Wang and Zhi-Hua Zhou. A new analysis of co-training. In *ICML*, 2010.

[202] Wei Wang and Zhi-Hua Zhou. Co-training with insufficient views. In *ACML*, 2013.

[203] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[204] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[205] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, CALIFORNIA Institute of Technology, 2010.

[206] Lior Wolf, Tal Hassner, and Yaniv Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008.

[207] Lior Wolf, Tal Hassner, and Yaniv Taigman. Similarity scores based on background samples. In *ACCV*, 2009.

[208] John Wright, Allen Y. Yang, Arvind Ganesh, Shankar S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[209] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, December 2007.

[210] Xinxiao Wu, Dong Xu, Lixin Duan, and Jiebo Luo. Action recognition using context and appearance distribution features. In *CVPR*, 2011.

[211] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[212] Yanshan Xiao, Bo Liu, Longbing Cao, Jie Yin, and Xindong Wu. Smile: A similarity-based approach for multiple instance learning. In *ICDM*, 2010.

[213] Pengtao Xie and Eric P. Xing. Multi-modal distance metric learning. In *IJCAI*, 2013.

[214] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, 2002.

[215] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013.

[216] Dong Xu, Yi Huang, Zinan Zeng, and Xinxing Xu. Human gait recognition using patch distribution feature and locality-constrained group sparse representation. *IEEE Transactions on Image Processing*, 21(1):316–326, 2012.

[217] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *NIPS*, 2004.

[218] Xinxing Xu, Ivor W. Tsang, and Dong Xu. Handling ambiguity via input-output kernel learning. In *ICDM*, pages 725–734, 2012.

[219] Xinxing Xu, Ivor W. Tsang, and Dong Xu. Soft margin multiple kernel learning. *IEEE Transactions on Neural Networks and Learning Systems*, 24(5):749–761, 2013.

[220] Xinxing Xu, Dong Xu, and Ivor W. Tsang. Video concept detection using support vector machine with augmented features. *Pacific-Rim Symposium on Image and Video Technology*, 2010.

[221] Zenglin Xu, Rong Jin, Irwin King, and Michael R. Lyu. An extended level method for efficient multiple kernel learning. In *NIPS*, 2008.

[222] Zenglin Xu, Rong Jin, Haiqin Yang, Irwin King, and Michael R. Lyu. Simple and efficient multiple kernel learning by group lasso. In *ICML*, 2010.

[223] Fei Yan, Krystian Mikolajczyk, Mark Barnard, Hongping Cai, and Josef Kittler. ℓ_p norm multiple kernel fisher discriminant analysis for object and image categorisation. In *CVPR*, 2010.

[224] Shengye Yan, Xinxing Xu, and Qingshan Liu. Learning the object location, scale and view for image categorization with adapted classifier. *Information Sciences*, 281(10):661–673, 2014.

[225] Shengye Yan, Xinxing Xu, Dong Xu, Stephen Lin, and Xuelong Li. Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification. In *ECCV*, pages 473–487, 2012.

[226] Shengye Yan, Xinxing Xu, Dong Xu, Stephen Lin, and Xuelong Li. Image classification with densely sampled image windows and generalized adaptive multiple kernel learning. In *IEEE transactions on cybernetics*, 2014.

[227] Shuicheng Yan, Dong Xu, Benyu Zhang, HongJiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[228] Haiqin Yang, Zenglin Xu, Jieping Ye, Irwin King, and Michael R. Lyu. Efficient sparse generalized multiple kernel learning. *IEEE Transactions on Neural Networks*, 22(3):433–446, 2011.

[229] Jian-Bo Yang and Ivor W. Tsang. Hierarchical maximum margin learning for multi-class classification. In *UAI*, 2011.

[230] Jianchao Yang, JOHN Wright, Thomas S. Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[231] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[232] Liu Yang. Distance metric learning: A comprehensive survey, 2006.

[233] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *ICML*, 2009.

[234] Shipeng Yu, Balaji Krishnapuram, Romer Rosales, Harald Steck, and R. Bharat Rao. Bayesian co-training. In *NIPS*, 2007.

[235] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[236] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[237] Kai Zhang, Ivor W. Tsang, and James T. Kwok. Maximum margin clustering made practical. *IEEE Transactions on Neural Networks*, 20(4):583–596, 2009.

[238] Qi Zhang and Sally A. Goldman. Em-dd: An improved multiple-instance learning technique. In *NIPS*, 2002.

[239] Qi Zhang, Sally A. Goldman, Wei Yu, and Jason E. Fritts. Content-based image retrieval using multiple-instance learning. In *ICML*, 2002.

[240] Bin Zhao, James T. Kwok, and Changshui Zhang. Multiple kernel clustering. In *SDM*, 2009.

[241] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013.

[242] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):653–668, 2013.

[243] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[244] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML*, 2009.

[245] Fan Zhu and Ling Shao. Weakly-supervised cross-domain dictionary learning for visual recognition. *International Journal of Computer Vision*, pages 42–59, 2014.

[246] Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *ICML*, 2007.

[247] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

Appendix A

Appendix

A.1 Proof of Proposition 3

We can rewrite the problem (3.10) in the following form:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in A,\tau,\zeta_m} \quad & -\tau + \theta \sum_{m=1}^{M} \zeta_m \\ \text{s. t.} \quad & -\frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) \geq \tau - \zeta_m, \\ & \zeta_m \geq 0, m = 1, \ldots, M, \end{aligned} \tag{A.1}$$

where the domain for $\boldsymbol{\alpha}$ is $A = \{\boldsymbol{\alpha} | \boldsymbol{\alpha}'\mathbf{1} = 1, \boldsymbol{\alpha}'\mathbf{y} = 0, 0 \leq \boldsymbol{\alpha} \leq C\}$.

The Lagrangian of problem (A.1) is

$$\begin{aligned} \mathcal{L} = \quad & -\tau + \theta \sum_{m=1}^{M} \zeta_m - \sum_{m=1}^{M} \mathrm{z}_m \zeta_m \\ & + \sum_{m=1}^{M} \mu_m \Big(\frac{1}{2} (\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) + \tau - \zeta_m \Big), \end{aligned} \tag{A.2}$$

where $\mu_m \geq 0$ and $\mathrm{z}_m \geq 0$ are the non-negative Lagrangian multipliers for inequalities in (A.1). Setting the gradient of the Lagrangian with respect to the primal variables τ and ζ_m, we get the following

$$\sum_{m=1}^{M} \mu_m = 1, \tag{A.3}$$

$$\theta - \mu_m - \mathrm{z}_m = 0, m = 1, \ldots, M. \tag{A.4}$$

Substituting equation (A.3) and (A.4) back into the Lagrangian, the proof is completed.

A.2 Proof of Proposition 4

The objective function for hinge loss soft margin MKL can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in A,\tau,\zeta_m} \quad & -\tau + \frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^2 \\ \text{s. t.} \quad & -\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y}) \geq \tau - \zeta_m, \end{aligned} \tag{A.5}$$

where the domain for $\boldsymbol{\alpha}$ is $A = \{\boldsymbol{\alpha}|\boldsymbol{\alpha}'\mathbf{1} = 1, \boldsymbol{\alpha}'\mathbf{y} = 0, 0 \leq \boldsymbol{\alpha} \leq C\}$.

The Lagrangian of problem (A.5) is

$$\begin{aligned} \mathcal{L} = \quad & -\tau + \frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^2 \\ & + \sum_{m=1}^{M}\mu_m\Big(\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y}) + \tau - \zeta_m\Big), \end{aligned}$$

where $\mu_m \geq 0$'s are the nonnegative Lagrangian multipliers of the inequalities in (A.5). Setting the gradient of the Lagrangian with respect to the primal variables λ and ζ_m, we can get the following

$$\sum_{m=1}^{M}\mu_m = 1, \tag{A.6}$$

$$\theta\zeta_m - \mu_m = 0, m = 1, \ldots, M. \tag{A.7}$$

Substituting equation (A.6) and (A.7) back into the Lagrangian, the proof is completed.

Publication

Journal Publications

Published

- Shengye Yan, **Xinxing Xu**, Dong Xu, Stephen Lin, and Xuelong Li, "Image Classification with Densely Sampled Image Windows and Generalized Adaptive Multiple Kernel Learning," *IEEE Transactions on Cybernetics*, June 2014.

- Shengye Yan, **Xinxing Xu**, and Qingshan Liu, "Learning the object location, scale and view for image categorization with adapted classifier," *Information Sciences,* March, 2014.

- **Xinxing Xu**, Ivor W. Tsang and Dong Xu, "Soft Margin Multiple Kernel Learning," *IEEE Transactions on Neural Networks and Learning Systems* (**T-NNLS**), vol. 24, no. 5, pp. 749-761, May 2013.

- Dong Xu, Yi Huang, Zinan Zeng and **Xinxing Xu**, "Human Gait Recognition Using Patch Distribution Feature and Locality-Constrained Group Sparse Representation," *IEEE Transactions on Image Processing* (**T-IP**), vol. 21, no. 1, pp. 316-326, January 2012.

Under Review

- **Xinxing Xu**, Wen Li, Dong Xu and Ivor W. Tsang, "Co-Labeling for Multi-view Weakly Labeled Learning," submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (**T-PAMI**).

- Xinxiao Wu, **Xinxing Xu**, Dong Xu and Yunde Jia, "View-Invariant Action Recognition using ℓ_p-Norm Latent Multiple Kernel Learning," submitted to International Journal of Computer Vision (**IJCV**), under major revision.

- **Xinxing Xu**, Wen Li and Dong Xu, "Face Verification and Person Re-identification in RGB Images by Learning Distance Metric from Weakly Labeled RGB-D Data," submitted to IEEE Transactions on Neural Networks and Learning Systems (**T-NNLS**).

Conference Publications

- **Xinxing Xu**, Ivor W. Tsang, and Dong Xu, "Handling Ambiguity via Input-Output Kernel Learning," *in Proceedings of IEEE International Conference on Data Mining* (**ICDM**), Brussels, Belgium, December 2012, pp. 725-734.

- Shengye Yan, **Xinxing Xu**, Dong Xu, Stephen Lin and Xuelong Li, "Beyond Spatial Pyramids: A New Feature Extraction Framework with Dense Spatial Sampling for Image Classification," *in Proceedings of European Conference on Computer Vision* (**ECCV**), Firenze, Italy, October 2012.

- **Xinxing Xu**, Dong Xu, and Ivor W. Tsang, "Video Concept Detection Using Support Vector Machine with Augmented Features," *in Proceedings of the Fourth Pacific-Rim Symposium on Image and Video Technology* (**PSIVT**), Singapore, September 2010.