CLASSIFICATION WITH MULTI-IMPRECISE LABELS

By

SHENG ZOU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2021

To my Lord Jesus Christ

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisor, Dr. Alina Zare. She is an excellent teacher who teaches the knowledge in a very unambiguous fashion. She is also an experienced researcher who deeply understands the key problems in the research area and has insight into addressing them. I learned a lot from her about how to do research, how to be creative, and how to cooperate with other researchers, especially on interdisciplinary projects. I am grateful for her support for my Ph.D. work. This Ph.D. dissertation owes to her patient instruction and helpful suggestions. Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Paul Gader, Dr. Stephanie Bohlman, and Dr. Damon Woodard, for their insightful comments on my research. I also would like to thank my labmates for their real-time support in the lab. I thank Changzhe Jiao for his advice on my research and encouragement during the hard times of my study. Also, I thank Chao Chen, Xiaoxiao Du, and Guohao Yu who help me a lot at the early stage of my Ph.D. work. I thank Xiaolei Guo for her insightful questions and sincere encouragement. Lastly, I would like to thank my parents, who raise me and love me. They enlightened me and inspired me to explore the unknown.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, NOMENCLATURE, OR ABBREVIATIONS

ALC	Ambiguous label classification
APR	Axis-parallel rectangle
BCM	Beta compositional model
BS	Bag-space
DD	Diverse density
EDC	Evidential distance-based classifier
eFUMI	Extended functions of multiple instances
EM	Expectation maximization
EMDD	Expectation maximization diverse density
ES	Embedded-space
GMM	Gaussian mixture model
HSI	Hyperspectral image
IS	Instance-space
KNN	K-nearest neighbor
$\mathcal{L}$	Label matrix where each element $\ell_{i,j}^{m}$ denotes the labeling information (varies among different types of uncertain labels) on *instance level* or *bag level* for ith instance over mth class by jth annotator.
LDA	Latent Dirichlet allocation
LDL	Label distribution learning
LMM	Linear mixture model
MAP	Maximum a posteriori
MCMC	Markov Chain Mote Carlo

MESMA	Multiple endmember spectral mixture models
MH	Metropolis-Hasting
MI	Multiple instance
MI-ACE	Multiple instance adaptive cosine estimator
MI-HE	Multiple instance hybrid estimator
MIL	Multiple instance learning
MILES	Multiple instance learning via embedded instance selection
MIML	Multiple instance multi label
MI-SMF	Multiple instance spectral matched filter
MI-SVM	Multiple instance support vector machine
MLL	Multi-label learning
MRF	Markov random field
NCM	Normal Compositional model
NEON	National Ecological Observatory Network
PM-LDA	Partial membership latent Dirichlet allocation
PSO-EM	Particle swarm optimization expectation maximization
QP	Quadratic programming
RGB	Red, green and blue
SCM	Spatial compositional model
SDP	Semidefinite programming
SEM	Stochastic expectation maximization
SLL	Single-instance learning
S-PCUE	Sampling piecewise convex unmixing and endmember estimation

SVM Support vector machine

USGS United States Geological Survey

VCA Vertex component analysis

$\mathcal{W}$ Labeling reliability matrix where each element $w_{i,j}^{m}$ denotes the labeling reliability of jth annotator for labeling the ith instance as mth class

$\mathbf{X}$ Training set

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

CLASSIFICATION WITH MULTI-IMPRECISE LABELS

By

Sheng Zou

May 2021

Chair: Alina Zare
Cochair: Paul Gader
Major: Electrical and Computer Engineering

Imprecise labels or label uncertainty are common problems in many real supervised and semi-supervised learning problems. However, most of the state-of-the-art supervised learning methods in the literature rely on accurate labels. Accurate labels are often either expensive, time-consuming, or even impossible to obtain in many real applications. There are many approaches in the literature that address imprecise and uncertain labels of various types. However, not all types of label uncertainty and imprecision are addressed. Furthermore, the overwhelming majority of methods that address label imprecision and uncertainty generally address only one form of imprecision/uncertainty, even when many problems have more than one type of imprecise label coexisting in a problem.

Multiple instance (MI) label is one type of imprecise label. MI label is a shared label for a set of instances, or "bag", instead of one instance. For example, in tree species classification from remotely sensed hyperspectral imagery, precise species class labels for each pixel are often difficult or expensive to obtain, as stated above. In comparison, an imprecise species label can be much easier to be assigned to a region (a bag) in the hyperspectral image, which can be a tree polygon, generated by an ecologist or tree delineation algorithm. The imprecise species label indicates the existence of the labeled species in the region as a form of subpixel, a pixel, or several pixels. However, this MI assumption, assuming the existence of accurate bag labels, may not hold in some applications. For instance, tree species labeling requires expertise in tree species knowledge, and sometimes relatives of a tree species are difficult to tell apart, thus

the MI labels (species class label) for the bags (tree polygon) may be incorrect. In this case, another imprecise label, probabilistic label, can be introduced to incorporate the imprecision of MI labels. A probabilistic label is the confidence degree assigned to the original label as a way to show how probable the labeling by the ecologist is correct. In this case, a mixture of two types of imprecise labels, probabilistic labels and multiple instance labels, both present in the training set.

In this dissertation, different types of imprecise labels are defined and reviewed. A novel multi-imprecise label learning approach is proposed to address classification problems for real scenarios where probabilistic labels over the multiple instance bag-level labels present. In addition, the classes in the multiple instance problems are modeled by distributions (e.g. Gaussian Mixture Model) instead of learning prototypes to account for within-class variabilities. The classification approach proposed is applied to a variety of data sets with label uncertainty including hyperspectral data sets and medical imagery. Learning a distribution for each class of the imprecisely labeled data unveils the latent per-class distribution, not only improving the classification performance but also offering a way to visualize the variability of each class and deepen our understanding of the data, compared with other prototype-based or discriminative methods.

CHAPTER 1
INTRODUCTION

Imprecise labels, compared with accurate labels, are more realistic and cheaper to obtain in many machine learning applications, such as classification, regression and unmixing. For example, it is much easier to label image patches instead of hundreds of thousands of individual pixels in a training image, although there will be some pixels in each image patch that are different from the patch label class. A multiple instance learning based approach can often address this type of imprecision and learn a pixel level classifier, similar to or slightly worse than a classifier learned with accurate pixel level labels. Sometimes, imprecise labels offer a relatively suboptimal labeling mechanism, when the absolute accurate labeling for the training data is infeasible or requires many experts (which can be expensive!). For instance, for the classification of different tree species, very precise species labeling is difficult; for the classification of different topics in text, it is generally a hard task to choose only one topic class for a document since there are usually several topics involved with some degree. However, it is easier and more reasonable to associate a probability/confidence value to all the candidate species labels. Also, it is more accurate to use multi-labels with some memberships for all the candidate topics in the document. In both examples, the imprecise labels, allow for a more real and feasible labeling mechanism, compared to ideal and infeasible accurate labeling approaches.

Training data with imprecise labels have a variety of definitions based on the cause of the imprecision. Many types of imprecise labels have been tackled in the literature (Zhou, 2018). However, there are few studies where multiple types of imprecise labels are involved. Multiple types of imprecise labels are widely observed in classification in many different research areas, especially when crowdsourcing or labeling with a committee is used. For instance, a committee of doctors can individually label the possible lesion regions of a medical image. In this case, an average confidence vector is assigned to the candidate labels of the labeled region by fusing the labeling of all doctors. For the tree species labeling application discussed above, a probability/confidence value can be queried from experts or by exploiting neighboring species

information. For both examples, there are both probabilistic label and multiple instance label in the training label. Compared with only one type of imprecise label, this case is more real and allows for the possibility of incorporating as much information as possible to help the training procedure.

Most state-of-the-art multiple instance learning based approaches generally estimate a "concept" instance for each class. However, one instance is usually not adequate to cover the variability inside the class, especially for remote sensing applications. For instance, each tree species class usually has large spectral variability. Motivated by this issue, a multiple instance learning with distributions is proposed on the basis of modeling probabilistic labels over multiple instance labels. The goal is to model each class as a distribution to capture the intra-class variability.

1.1 Imprecise Label

Imprecise labels are a common problem in supervised and semi-supervised applications and, in particular, remote sensing applications. Imprecise labels (also known as label noise or label uncertainty) are often caused by the following reasons. First, the labeled samples can be composed of several classes and may be described using a fuzzy membership. For instance, the emotion classification of a face can be classified as both happy and surprised, with a membership of 70% on class "happy" and 30% on "surprised". Second, the annotator might be uncertain about the true class of the labeled pixel or object, so the annotator gives a label probability value about how confident they are that the label/object belongs to each class. Third, in some applications, the existence of a target class is certain in a region of the scene, but the accurate pixel-level spatial location of the target in this region is not. For instance, the polygons in Figure 1-1 are the regions all contain some wiregrass determined by the annotator, but the accurate pixel-level wiregrass labels are infeasible to obtain. Fourth, some real-value labels may be imprecise. For instance, the age of a person can be labeled as 26, or between 25 and 27. However, the real age is an irrational number and impossible to obtain. Lastly, the labels can simply be wrong because of less reliable, easy-to-get labeling from non-experts (e.g.

Amazon Mechanical Turk). These are some examples of imprecise labels. A more complete list of imprecise labels is reviewed in Chapter 2.

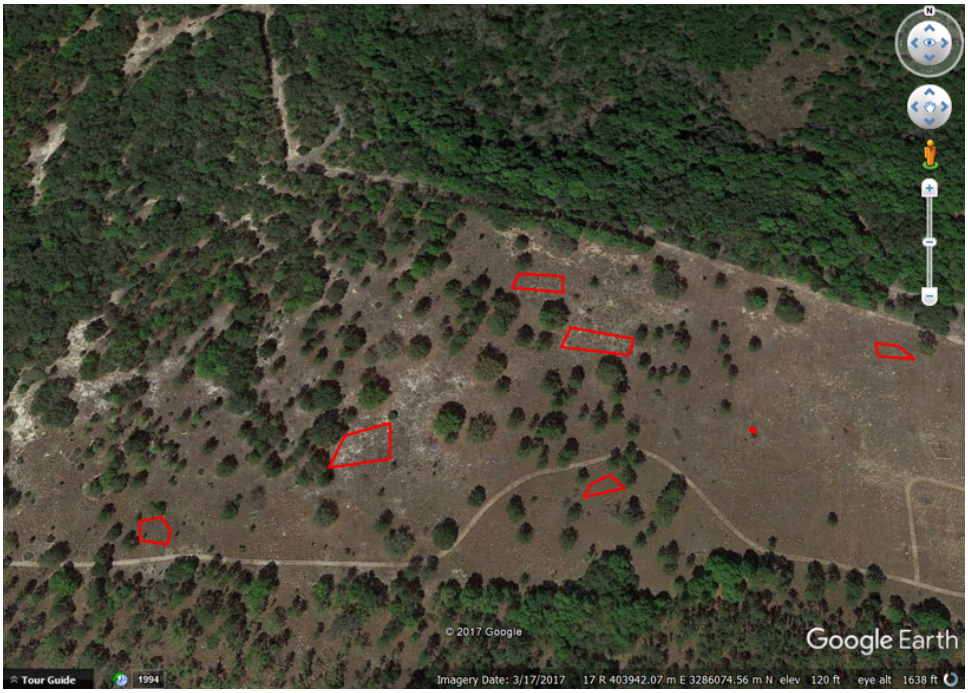


Figure 1-1. Wiregrass polygons in Ordway-Swisher Biological Station (OSBS)

There are many works in the literature addressing imprecise labels, such as looking for and correcting the imprecise labels, looking for and only using the precise labels, or building a noise-robust model. However, there are few works in the literature addressing more than one type of imprecise label, although multiple types of imprecise labels are commonly seen in many real applications. Two types of imprecise labels, the probability label over the multiple instance bag level label, are studied in this dissertation since the state-of-the-art methods in the literature can only classify on data with either multiple instance bag level labels or probabilistic labels.

1.2 Applications of Classification with Imprecise Labels

Imprecise labels widely exist for remote sensing applications. The first application that is studied in this thesis is tree species classification of hyperspectral imagery over Ordway-Swisher Biological Station (OSBS). Hyperspectral images are collected and the tree species labels are

annotated by National Ecological Observatory Network (NEON). Species labels are assigned to the centers of tree trucks. After some pre-processing steps, the centers can be expanded to regions (e.g. a square shown in Figure 1-2 or circle or polygon), based on the measured size of the tree crown or some other factors. The regions are modeled using bags in multiple instance learning (MIL). Then, the tree crown region is assigned with the species class. This is not accurate since there are many pixels in the region that are not the assigned class, such as sand, soil, neighboring species, and grass. In addition, these species labels are not 100% accurate since some species are too similar so it's easy to mislabel. Fortunately, some species are more probable to grow together, as we know from ecology. So a model can be proposed to infer the probability of the species label based on its neighboring species in future work. In this application, there are probabilistic labels over the bag-level labels.

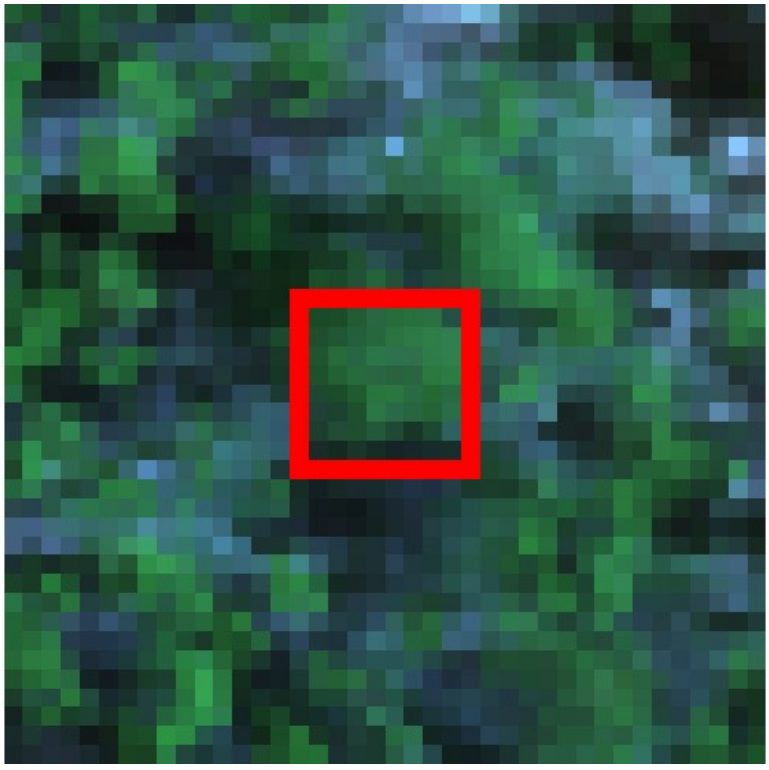


Figure 1-2. NEON tree crown image (extracted from HSI) and its MIL bag

The second application that is studied is very similar to the first application. It is also tree polygons outlined and labeled by experts from the University of Santa Barbara (UCSB). Each

polygon can be viewed as a bag in MIL. The bag label indicates that there are more than 70% of a certain class inside the polygon. Similar to NEON datasets, the probabilistic label of the MIL label can be inferred by the neighboring tree species.

The third application that is investigated is the classification of diabetic retinopathy, using RGB (red, green, and blue) images. The task using the DIARETDB1 dataset is to classify each color fundus image into a binary class of "with" or "without" diabetic retinopathy. Both diabetic retinopathy and the level of the disease can be diagnosed by checking its signs (features), which are microaneurysms, soft exudates, hard exudates, and hemorrhages. Four medical experts labeled the same color fundus images using polygons, circles, or ellipses, indicating the possible regions of each of the four signs, shown in Figure 1-3. Each polygon, circle, or ellipse can be viewed as a bag. Since some regions are labeled from more than one expert as one of the four signs, these regions have higher probabilities on their corresponding bag labels.

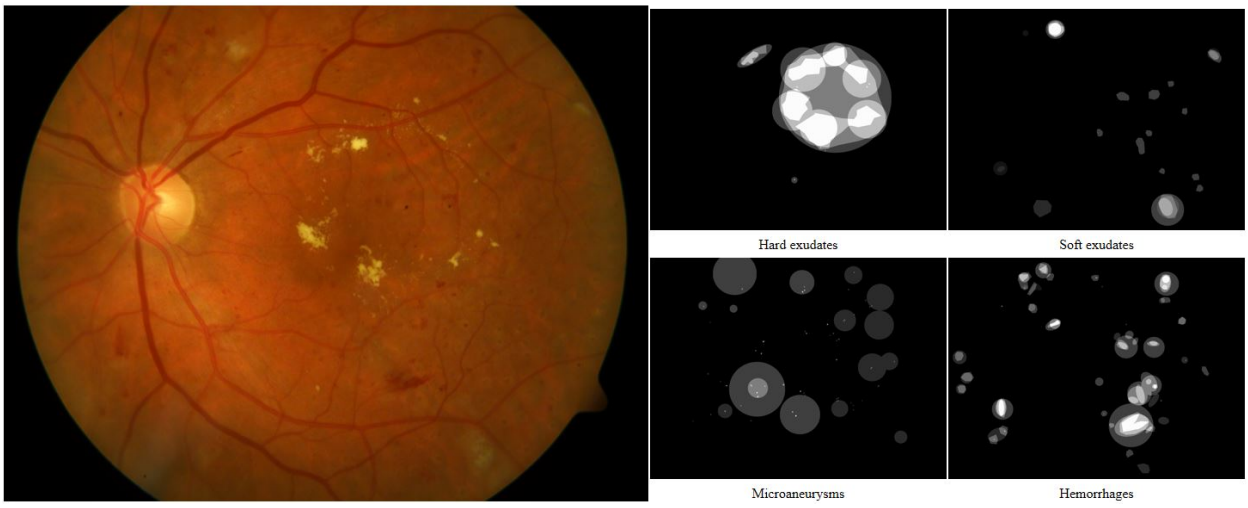


Figure 1-3. Fundus image and its MIL bags

CHAPTER 2
LITERATURE REVIEW

As stated in chapter 1, imprecise labels have many definitions based on the cause of the imprecision. Different types of imprecise labels have different mathematical forms and there are various approaches to tackle them. In this chapter, a complete review of different types of imprecise labels and how they are addressed by the state-of-the-art is provided. More importantly, it is found that there are few studies on imprecise labels with multiple types of imprecision involved. Hyperspectral applications, such as unmixing and classification, are some of the applications where multiple types of imprecise labels exist. After the review of imprecise labels, the spectral variability of the hyperspectral image is reviewed. Here, spectral variability is grouped into two categories in the literature, sets based and distribution based, which is the motivation for proposing the distribution-based multiple instance learning approach.

2.1 Existing Label Uncertainty Algorithm

Uncertain labels, also called noisy labels, or imprecise labels in the literature refer to the observed training labels that are not accurate or reliable in classification. For instance, incorrectly setting a negative label on a positive instance in binary classification(Frénay and Verleysen, 2014).

The main reason leading to the uncertain labels is that accurate training labels are usually expensive or time-consuming or impossible to obtain. For instance, in remote sensing applications, assume we run an SVM algorithm on an image to classify different objects in a scene. A standard approach needs a set of pixel-level training labels. This training set is usually hundreds of thousands of pixels that need to be labeled, which is extremely time-consuming and infeasible for many real applications. Therefore, many training labels are obtained by some cheap, easy-to-get non-expert labeling framework. These non-expert frameworks are usually less reliable or even randomly assign labels when they have no knowledge of the labeling problems, resulting in imprecise labels. The accuracy of labeling also relies on the information provided by the labeled source. For example, in the diagnosis (labeling) process on a patient,

if the patient provides imprecise answers related to the symptoms, the diagnosis result is not reliable. In other words, the information provided to the expert may not be sufficient for reliable labeling.

Not only the non-expert framework labeling process has labeling errors, but expert labeling is subjective in many applications. In medical applications, for example, the labeling of medical images to determine if a specific disease exists may vary among doctors with their subjective understanding. The disagreement among annotators can be characterized as a confidence value or probability value by voting on the labeling results from different annotators. The disagreement can not only exist among annotators but also exist in a single annotator. One annotator can provide a confidence vector for all possible labels when labeling training data.

Label uncertainty can also stem from the multi-label characteristic of the training data itself. In other words, compared with traditional supervised learning where one single instance is associated with one (and only one) true single label, a single instance is associated with multiple true labels. To name a few, in hyperspectral classification, a single pixel can be composed of several different materials; in text categorization, a document can contain several topics, such as policy, education simultaneously. Hence, a label set and an associated fuzzy membership set are obtained for this type of uncertain label. If only a crisp label is provided to this kind of training data, the labels are imprecise.

In some applications, only accurate high-level label information can be obtained because of the limited labeling source or difficulty of low-level labeling. For instance, in some semi-supervised hyperspectral unmixing or classification applications, only a high-level training label can be accurately obtained. To be more specific, the annotator is capable of giving an identical label to a group of pixels in the hyperspectral image indicating the possible existence of target pixel(s) in this group of pixels. However, the precise pixel-level labels are not provided, due to the reasons such as the high cost of pixel-level labeling and GPS error for a single pixel.

The uncertain labels can be grouped into four cases according to the six scenarios above, which are noisy labels, probabilistic labels, fuzzy and multi- labels, multiple instance labels, multiple instance multi-labels, and regression labels.

2.1.1 The Framework of Label Uncertainty

Let us assume that there are N training instances $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a D-dimensional real-valued feature vector. Let $\mathcal{W} = \{w_{i,j}^m\}_{i=1:N,j=1:K}^{m=1:M}$ be the labeling reliability matrix where $w_{i,j}^m \in [0,1]$ denotes the how much we trust ($w_{i,j}^m = 0$ denotes not trust; $w_{i,j}^m = 1$ denotes completely trust) on jth annotator for labeling the ith instance as mth class where K is the number of annotators and M is the number of classes. In literature, It is assumed that $w_{i,j}^m$ does not depend on the instance $\mathbf{x}$ to simplify the problem such that $w_{i,j}^m$ can be simplified to be w_j^m. However, the original form of labeling reliability, $w_{i,j}^m$, is more general and suitable for real applications. Note that even though the simplified form, w_j^m, is assumed, the original form can also assumed for all types of uncertain labels in this section for future work. There are several common assumptions on $\mathcal{W}$ shown in Equations 2–1, 2–2, 2–3 and 2–4:

$$w_j^m = 1, \forall j, m \tag{2–1}$$

where Equation 2–1 denotes that all annotators are fully trusted. Even though the annotators are assumed to 100% accurately on labeling, it does not mean there is no label uncertainty. Since the label information can be not only a label indicator but label confidence, fuzzy membership, label interval or a high-level label provided by the annotators, containing a certain level of uncertainty for the true underlying label(s).

$$w_1^m = w_2^m =, ..., w_K^m, \forall m \quad \text{and} \quad w_j^1 \neq w_j^2 \neq, ..., w_j^M, \forall j \tag{2–2}$$

where Equation 2–2 represents the case that annotators have the same labeling reliability but for an individual annotator, he or she has different labeling reliabilities on different classes.

$$w_1^m \neq w_2^m \neq, ..., w_K^m, \forall m \quad \text{and} \quad w_j^1 = w_j^2 =, ..., w_j^M, \forall j \tag{2–3}$$

where Equation 2–3 represents the case that different annotators have different labeling reliabilities but for an individual annotator, he or she has the same labeling reliability on all classes.

$$w_1^1 \neq \ldots w_j^m \neq \ldots w_K^M, \forall j, m \tag{2–4}$$

where Equation 2–4 denotes the most realistic case that different annotators have different labeling reliabilities on different classes.

Let $\mathcal{L} = \{\ell_{ij}^m\}_{m=1}^M$ denote the complete set of label information, where ℓ_{ij}^m represents the labeling information (varies among different types of uncertain labels) on instance level or bag level for ith instance over mth class by jth annotator. The label information assigned to the instance can be in instance level or bag level. The label assigned in instance level is based on the observation of the single instance only, but label assigned in bag level is based on the observation of a set of instances together. In other words, all instances in a bag have the same bag level label(s). Thus, the instance level label is generally more precise than the bag level label, in terms of each instance. In some scenarios, bag level labels are preferred, for example, when the instance level labels are expensive or difficult to obtain. For any uncertain labels with labeling reliability the satisfying Equation 2–1 and 2–2 can be simplified to single annotator labeling using majority voting on the label information, i.e. we use $\mathcal{L} = \{\ell_i^m\}_{m=1}^M$ to denote the label information set.

There are some common assumptions on $\mathcal{L}$ shown in Equation 2–5, 2–6, 2–7 2–9 and 2–10.

$$\ell_{ij}^m \in \{0, 1\} \quad \text{s.t.} \sum_{m=1:M} \ell_{ij}^m = 1, \forall i, j \tag{2–5}$$

where ℓ_{ij}^m denotes a single label indicator, indicating there is a single true label for each instance where the labeled class is the class with $\ell_{ij}^m = 1$.

$$\ell_{ij}^m \in \{0, 1\} \tag{2–6}$$

where ℓ_{ij}^{m} denotes a multi-label indicator, indicating there are multiple true labels for each instance where the labeled classes are the classes with $\ell_{ij}^{m} = 1$.

$$\ell_{ij}^{m} \in [0,1]_{Pro} \quad \text{s.t.} \sum_{m=1:M} \ell_{ij}^{m} = 1, \forall i, j \tag{2–7}$$

where ℓ_{ij}^{m} denotes a probability value, indicating the probability that the true label is mth class for ith instance provided by jth annotator.

$$\ell_{ij}^{m} \in [0,1]_{Pos} \tag{2–8}$$

where ℓ_{ij}^{m} denotes a possibilistic value, indicating the possibility that the true label is mth class for ith instance provided by jth annotator.

$$\ell_{ij}^{m} \in [0,1]_{Fuz} \quad \text{s.t} \sum_{m=1:M} \ell_{ij}^{m} = 1, \forall i, j \tag{2–9}$$

where ℓ_{ij}^{m} denotes a fuzzy membership value, indicating the degree that the mth class can fully describe the ith instance provided by jth annotator.

$$\ell_{ij}^{m} \in \mathbb{R} \tag{2–10}$$

where ℓ_{ij}^{m} denotes a real value, indicating an approximate label in regression problems.

$$\ell_{ij}^{m} \in [r_1, r_2], r_1, r_2 \in \mathbb{R}, \quad r_1 < r_2 \tag{2–11}$$

where ℓ_{ij}^{m} denotes a real-value set, a range of real values, indicating the possible range of label in regression problems.

2.1.2 Noisy Labels

Noisy labels represent the case that there is no other information associated with the inaccurate labels except for the label indicators (Frénay and Verleysen, 2014). Also, a traditional supervised learning framework is assumed that only one label is associated with each instance satisfying Equation 2–5.

2.1.2.1 Classification with equal weighted annotators

In this framework, the labeling reliability is assumed to be consistent over different annotators satisfying Equation 2–2. If $\mathcal{L}_i$ doesn't match the true label indicator of instance x_i, it is called a noisy label. For instance, incorrectly labeling an apple as orange in an apple-vs-orange binary classification problem. There are three main categories of approaches in the literature in terms of how to address noisy labels, which are label noise robust methods, data cleansing methods and label noise tolerant methods.

Label noise robust methods are the approaches that are naturally less sensitive to label noise than others. These methods have been shown to retain satisfactory performances, though the training data are corrupted by a certain level of wrong training labels. Some techniques embedding with a regularization term used for avoiding overfitting can address label noise more effectively (Teng, 2000, 2001, 2005). Data cleansing methods are the approaches that are capable of detecting the noisy labels followed by a correction step. In the correction step, the noisy labels can be removed, relabeled or the training data with noisy labels can be removed. Generally, the data cleansing method is applied, and a cleansed training dataset is generated before the learning algorithm. Similar to outlier detection or anomaly detection, some techniques are based on ad hoc measures where the training data are removed when the confidence of anomaly detection is over a certain threshold. For instance, Brodley et al. (1996) presented a method that can detect the mislabeled training data, similar to outlier detection, without any assumption on the learning approaches. The idea is to train a set of classifiers using a part of training data and then test on the remaining part of the training data. The instances for which the classifier disagrees most are relabeled using the predicted labels. Label noise tolerant methods are the methods that naturally learn the label noise during the learning procedure. Most of the label noise tolerant methods are probabilistic models, including Bayesian (Pérez et al., 2007) and frequentist methods (Eskin, 2000). For Bayesian approaches, Bayesian priors are widely used on the mislabeling probabilities (Joseph et al., 1995; Gaba and Winkler, 1992). The choice for the Bayesian priors can be Beta priors (Zhang et al., 2005) and

Dirichlet priors (Ruiz et al., 2008). An indicator variable α_i is defined by Rekaya et al. (2001), indicating the switched label for the associated instance x_i if $\alpha_i = 1$. Therefore, each indicator follows a Bernoulli distribution based on mislabeling rate. The mislabeling rate is assumed to follow a Beta prior. The frequentist methods consider the label noise as a stochastic process (Eskin, 2000).

2.1.2.2 Classification with multiple weighted annotators

In previous binary or multi-class classification examples, if there are multiple annotators, the reliability of each annotator is treated equally. However, this is too ideal and not always true in real applications. For example, the annotators in crowdsourcing usually have different knowledge backgrounds and thus, have different error rates when labeling. Even for the labeling by a committee of experts, the labeling accuracy may vary based on the level of expertise of each expert. In this section, the performance of the annotator is regarded as a variable and characterized using weights in Equation 2–4.

Raykar et al. (2010) presents a Bayesian model to characterize the performance of each annotator. Without loss of generality, a binary classification is considered as an example and Equation 2–5 is satisfied where $M = 2$. The class indicator values of class 1 and 2 assigned to an instance $\mathbf{x}$ are ℓ_j^1 and ℓ_j^2, respectively. The true class indicator values are defined as $\hat{\ell}_j^1$ and $\hat{\ell}_j^2$. Therefore, sensitivity and specificity are used to represent the probability that the annotator correctly labels the instance with the true label of 1 and 0, respectively. More formally, the sensitivity α^j for jth annotator is defined in Equation 2–12. Sensitivity is also called true positive rate.

$$\alpha^j := Pr\left(\ell_j^1 = 1 | \hat{\ell}_j^1 = 1\right) \tag{2–12}$$

The specificity β^j for jth annotator is defined in Equation 2–13. Specificity is also called false positive rate.

$$\beta^j := Pr\left(\ell_j^2 = 1 | \hat{\ell}_j^2 = 1\right) \tag{2–13}$$

It is assumed that α^j and β^j are both consistent over all instances for jth annotator. For multiclass classification, a more general form of weights, w_j^m can be used to replace α^j (can be viewed as w_j^1) and β^j (can be viewed as w_j^2). Therefore, under a Bayesian framework, priors can be imposed on the sensitivity and specificity to incorporate the labeling performance of each annotator. Since α^j and β^j represent a binary classification problem, beta priors are suggested by Raykar et al. (2010). Thus, the beta priors for sensitivity and specificity can be represented as

$$Pr\left(\alpha^j | a_1^j, a_2^j\right) = Beta\left(\alpha^j | a_1^j, a_2^j\right) \tag{2–14}$$

$$Pr\left(\beta^j | b_1^j, b_2^j\right) = Beta\left(\beta^j | b_1^j, b_2^j\right) \tag{2–15}$$

where $a_1^j, a_2^j, b_1^j, b_2^j$ are the hyperparameters for beta distributions. Then Raykar et al. (2010) proposed a model that sets the true labels as latent variables and jointly estimates the true labels and a classifier by maximum a posteriori (MAP) estimator optimized by EM algorithm.

2.1.3 Probabilistic Labels

Probabilistic labels are also called ambiguous labels. In this setting, an instance may be labeled in a non-unique way by a subset of classes, similar to multi-label classification (Hüllermeier and Beringer, 2006). A confidence or probability set is also provided by the annotator, which characterizes how confident or probable that the training data belongs to each class in the subset. The probability set can be a real value between 0 and 1, or even an indicator set denoting possible or impossible for each candidate class (Ambroise et al., 2001). But the existence of a unique correct classification is assumed, and the labels are regarded as candidates. The general process in the literature to address ambiguous labels refers to ambiguous label classification (ALC). There are many works in the literature addressing the probabilistic labels and can be roughly categorized into the following two scenarios: binary classification by equal weighted annotators, multiclass classification by equal weighted annotators.

2.1.3.1 Binary classification

Nguyen et al. (2014) proposed to associate soft-label information by annotator with training labels for binary classification. Soft-label information is the additional confidence value/level reflecting how strongly the annotator feels about the class labels. In this scenario, the additional confidence value/level is assumed to be provided by a single annotator or averaged by multiple annotators who are considered to have the same labeling ability and be fully trusted. More formally, Equation 2–1 is satisfied. Assuming the binary classification task is to label the fruit between apple (class 1) and orange (class 2). Generally, soft-label information can be represented by (1) a probability in Equation 2–7 where $M = 2$ for binary classification, for instance, the probability that the ith fruit is an apple is 0.7 ($\ell_i^1 = 0.7$ and $\ell_i^2 = 0.3$), or (2) an ordinal category, for example, the annotator 'strongly agree' that the fruit is an apple where the possible ordinal categories including 'strongly agree' , 'weakly agree', 'weakly disagree' and 'strongly disagree'.

Soft-label information proved in (Nguyen et al., 2014), can help to learn a classification model more efficiently than with binary labels only, when the number of training labels is limited. These algorithms leveraging soft-label information vary based on different types of soft-label information, i.e., probabilistic labels or categorical labels.

For probabilistic labels, soft-label information is defined as the probability value of the most probable class. More formally, $c_i = \max(\ell_i^1, \ell_i^2)$ for $\mathbf{x}_i$. One approach considers the learning procedure as discriminative linear regression which regresses the features directly to probabilities. Thus, assuming a linear regression mapping, $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w}$ are the weights of the model. The learning procedure is to minimize the following objective function:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{w}^T\mathbf{x}_i - c_i\right)^2 + Q(\mathbf{w}) \tag{2–16}$$

where $Q(\mathbf{w})$ is the regularization term preventing overfitting.

Another approach regards the learning procedure as logistic regression. Unlike the linear regression where the output is unbounded and inconsistent with probability, the logistic

regression naturally outputs the values in the range between 0 and 1. Therefore, similarly, the learning procedure is to minimize the following objective function:

$$\mathbf{w}^* = \arg\min_{\mathrm{w}} \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{w}^T \mathbf{x}_i - t\left(c_i\right)\right)^2 + Q\left(\mathbf{w}\right) \tag{2–17}$$

where $Q\left(\mathbf{w}\right)$ is the regularization term preventing overfitting and $t\left(c_i\right) = \ln \frac{c_i}{1-c_i}$ is the inverse of logistic function.

For categorical labels, the soft label information is no longer probabilities but several crisp levels. The authors suggested to use ordinal regression based on SVM approach. The main idea is to construct a regression function $f\left(\mathbf{x}\right) = \mathbf{w}^T\mathbf{x}$ mapping each $\mathbf{x}$ onto a real line such that instances in each categories are projected to a compact and well separated region. In SVM, in order to apply the categorical labels, a set of inequality constraints are encoded. More specifically, let $\mathbf{b} = b_1, b_2, ..., b_{r-1}$ represent the boundaries that separate r categories. For one instance $\mathbf{x}_i$ assigned to category c_j, it should also satisfy $b_1 < ... < b_{j-1} < f\left(\mathbf{x}\right) < b_j < b_{j+1} < ... < b_{r-1}$.

2.1.3.2 Multiclass classification

Multiclass classification is a general case of binary classification where there are more than two possible classes in the classification. In this case, the same scenario is assumed as a binary classification that the labels are provided by a single annotator or averaged by multiple equally treated annotators. Similar to binary classification, it also satisfies Equation 2–1 and Equation 2–7 where $M > 2$ for multiclass classification. Note that multiclass classification is different from multi-label classification where the former allows a unique class (label) for each instance but the latter allows multiple classes (labels) associated with each instance. Assuming the multiclass classification task is to label the fruit among apple, orange and banana. An instance x_i can be labeled as $\mathcal{L}_i = \{0.7, 0.2, 0.1\}$, denoting the probability values that this fruit is an apple, orange, or banana, respectively. Since there are more than two classes, a more general case can be assumed that no probability/confidence values are provided for each training instance. Instead, a subset of possible labels, where each candidate label has the same

probability, is provided by the annotator. For instance, an instance x_i can be labeled as either the apple or orange but not the banana. In other words, $\mathcal{L}_i = \{0.5, 0.5, 0\}$.

Jin and Ghahramani (2003) presents an approach that models the multiclass classification problems. It is capable of inferring the correct label among the set of candidate labels and achieve a performance close to the case of training with true labels. They proposed the Expectation Maximization (EM) and EM+prior methods to address the two cases defined above in terms of with or without probability/confidence values. The general idea is to assume a parameterized classifier with some unknown parameters to be estimated. A maximum likelihood criterion is used to estimate the parameters such that the target label has a high probability of being a member of the label set. To incorporate the prior information on the class labels, they generalize the likelihood function so that the estimated label distribution has low relative entropy with the prior on the class labels. The authors also prove that the EM+prior model can not only take advantage of the probability information provided to the candidate label set but is robust to a small amount of noise on the prior distribution over class labels.

2.1.4 Possibilistic Labels

Possibilistic labels are more relaxed imprecise labels, compared with probabilistic labels. In probabilistic labels, there is an underlying assumption that there must be at least one class that is completely possible, while for possibilistic labels, the instance may belong to none of the enumerated categories. Equation 2–8 is satisfied. The possibilistic labels can be obtained from a single annotator, who is asked to provide a real number between 0 and 1 as the degree of possibility that ith instance belongs to mth class, or obtained by multiple annotators, who is asked to provide a binary number for the possibility that ith instance belongs to mth class (1 for possible, 0 for impossible) and the degree of possibility is aggregated by voting. For either case, Equation 2–1 is satisfied. Denœux and Zouhal (2001) suggested using model and tackle the possibilistic labels within the framework of Evidence Theory. They introduced an evidential distance-based classifier (EDC) and generalized it to address the more general

possibilistic labels, where the degree of possibility (a real value between 0 and 1) instead of binary possibility (0 or 1) is associated with the class labels of training instances. The EDC method uses a belief function to model the class possibility of the unseen testing instance. Two approaches were proposed based on the belief function. One is to transform each possibility distribution into a consonant belief function. The other one is to use generalized belief structures with fuzzy focal elements.

2.1.5 Fuzzy Labels and Multi-labels

Fuzzy and multi-labels represent the scenario that the label for each instance is not unique. In other words, the candidate labels from the label set for each instance are not mutually exclusive. Multi-label learning (MLL) (Zhang and Zhou, 2014) assumes indiscriminate importance within the irrelevant label set while label distribution learning (LDL) (Geng, 2016) allows direct modeling of different importance of each label to the instance. The labels corresponding to MLL and LDL are called multi labels and fuzzy labels, respectively.

2.1.5.1 Multi-labels

For the case of multi-labels, the true underlying labels associated with a training instance are usually more than one. Multi-labels denote the possible labels that can represent a single instance together (Zhang and Zhou, 2014). It can handle some real applications that the traditional supervised learning method can't. For example, real-world instances may have multiple semantic meanings simultaneously. For image labeling, the labels for the natural scene can both have mountain and ocean as the labels. For the topic modeling of an article, the labels (topics) for the article can have politics, economic and student at the same time. Therefore, a single label associated with one instance does not fit perfectly for these applications. Multi-label learning (MLL) is proposed by Zhang and Zhou (2014), allowing multiple labels for a single instance. In MLL, annotators are assumed to have the same labeling reliability, satisfying Equation 2–2. To label an instance $\mathbf{x}_i$, assigning a binary indicator number $\ell_{ij}^m \in \{0, 1\}$ to mth label, referring to the existence of mth label for instance x_i. However, the

binary label indicator doesn't sum to one over the whole label set for an instance since there is more than one class for each instance. More formally, Equation 2–6 is satisfied.

2.1.5.2 Fuzzy labels

For fuzzy labels, the labels associated with a training instance are measured using a fuzzy membership set. Compared with ambiguous labels where this is only one true label for each instance, fuzzy labels allow for multiple labels simultaneously. The fuzzy membership, or called description degree (Geng, 2016), denotes the degree that each label describes one instance. For instance, assuming a training instance is a face emotion. The fuzzy labels associated with the facial emotion can be 70% happy and 30% surprised. Fuzzy labels can also be used for film rating. For example, the scores for a film can range from 0 to 10, labeled by audiences. Thus, there is no crisp score for this film but can be modeled using the score distribution by fuzzy labels. The general process in literature to address fuzzy labels refers to label distribution learning (LDL) (Geng, 2016). In LDL, annotators are assumed to have the same labeling reliability and fully trusted, satisfying Equation 2–1. To label an instance $\mathbf{x}_i$, assigning a real number ℓ_{ij}^m to mth label, referring to the degree to which jth label describe x_i. Let $\mathcal{L} = \{\ell_{ij}^m\}_{m=1}^M$ be the label distribution for instance x_i. It is assumed that the label set is complete, so there is a sum-to-one constraint $\sum_{m=1:M} \ell_{ij}^m = 1$ and non-negative constraint $\ell_{ij}^m \geqslant 0$ on the degree. In other words, Equation 2–9 is satisfied. Notice that the degree ℓ_{ij}^m is not the probability that the label of instance x_i is jth label, but the proportion that label accounts for a full description of instance x (Geng, 2016).

There are several approaches proposed in the literature to address the LDL problem. The first category is the problem transformation. A simple and straightforward method to transform the LDL problem into single-instance learning (SLL) problem. SLL denotes the traditional method that each instance is associated with only one label. To be more specific, assume that there are c labels in total. Each training instance $(\mathbf{x}_i, \mathcal{L}_i)$ can be transformed in to M single label instance $(\mathbf{x}_i, \ell_i^m)$ using the degree as the weight. Then the training dataset is resampled to the same size according to the weights, resulting in a $M \times N$ training set. At last, any

SLL approaches can be applied to the resampled training dataset. The second category is algorithm adaptation. Algorithm adaptation represents that some existing algorithms can be naturally adapted to fit the LDL problem, for example, the k-nearest neighbor (kNN) method. In details, the label distribution for mth label y_m of testing instance $\mathbf{x}$ can be represented as the mean value of the corresponding degree of its k nearest neighbors:

$$p\left(y_m|\mathbf{x}\right) = \frac{1}{k}\sum_{i\in N_k(\mathrm{x})} \ell_i^{y_m} \tag{2–18}$$

where $N_k(\mathbf{x})$ is the index set of the k nearest neighbors of $\mathbf{x}$ in the training set.

2.1.6 Multiple Instance (MI) Labels

Multiple Instance (MI) labels refer to some high level labels, for example, only bag level labels, instead of instance level labels are obtained. A bag represents a group of instances. In Multiple Instance Learning (MIL) (Maron and Lozano-Pérez, 1998), there are usually two types of bag labels, positive bag and negative bag. $\mathbf{X}_k = [\mathbf{x}_1, ..., \mathbf{x}_n] \subseteq \mathbf{X}$ is denoted as a kth bag, with a set of instances $\mathbf{x}_1, ..., \mathbf{x}_n$. Each bag is a subset of the entire training set. If one given bag label is negative, $\mathcal{L}_k = -$, then we know that all instances in a bag are negative, $\mathcal{L}_k = [\ell_1^-, ..., \ell_i^-, ..., \ell_n^-]$ where ℓ_i^- denotes an instance level label that truly belongs to the negative class. If it's positive, $\mathcal{L}_k = +$, then we know that at least one instance in the bag is labeled as positive, $\mathcal{L}_k = [\ell_1^+, ..., \ell_i^+, \ell_{i+1}^-, ..., \ell_n^-]$ where ℓ_i^+ denotes an instance level label that truly belongs to the positive class. Let the positive and negative class be class 1 and 2, respectively. For instances in positive bags, the underlying true instance level labels are unknown. Only bag level labels for an instance is obtained where $\ell_i^1 = 1$ denotes the bag level label is positive and $\ell_i^2 = 1$ denotes the bag level label is negative for $\mathbf{x}_i$. Generally, the bag level labeling reliability of each annotator is considered as fully trusted so that Equation 2–1 is satisfied. The bag level labels ℓ_i^m satisfies Equation 2–5 where $M = 2$.

MIL is proposed motivated by some real applications where the precise instance level labeling is expensive or infeasible. For example, in the target characterization of hyperspectral images, the target size is usually a couple of pixels or even sub-pixel, and the target location

is not precise because of the GPS error. Thus, the accurate pixel (instance) level labeling of the training target objects is usually infeasible. But a halo covering the target can be easily labeled as a positive bag, indicating the existence of the target inside the halo. Therefore, MIL based approaches can address the label uncertainty and infer the accurate pixel locations of the targets. Many MIL methods have been proposed, such as learning axis-parallel concepts (Dietterich et al., 1997), diverse density (Maron and Lozano-Pérez, 1998), extended Citation *kNN* (Wang and Zucker, 2000).

The goal of Multiple instance learning based classification is to train a model which can predict the bag-level class labels or instance-level labels of a testing bag. The algorithms in the literature can be categorized as three paradigms, Instance-Space (IS) paradigm, Bag-Space (BS) paradigm and Embedded-Space (ES) paradigm (Amores, 2013). The IS paradigm considers the instance-level, local discriminative information. It learns a discriminative classifier on the instance space that separates the underlying true positive instances from the true negative instances. For a testing bag, the bag-level classification is obtained by aggregating the instance-level classification information. The BS paradigm considers the bag-level, global discriminative information. It learns a discriminative classifier on the bag space and uses the information from the whole bags and classifies the whole bags. The ES paradigm is also another type of paradigm that considers the bag-level, global information. For ES based MIL algorithm, each bag is mapped to a feature vector capturing the information in the whole bag. Therefore, a discriminative classifier can be trained on the mapped embedded space. In other words, the MIL problem is transformed into a traditional supervised learning problem.

IS paradigm learns an instance-level classifier. One of the early works of IS paradigm is Axis-Parallel Rectangle (APR) (Dietterich et al., 1997). The goal of APR is to find an axis-parallel hyper-rectangle in the feature space. The APR is the minimum size hyper-rectangle of all possible hyper-rectangles that covers at least one instance from each positive bag and does not include any instance from negative bags. The most effective approach of the three solutions proposed by Dietterich et al. (1997) finding the optimal APR is called the 'inside-out'

approach. This approach estimates the smallest APR by growing the APR from a seed point in the feature space which covers at least one instance from each positive bag and no instances from negative bags. Similar to APR, Diverse Density (DD) (Maron and Lozano-Pérez, 1998) algorithm also follows IS paradigm. DD approach learns a concept point in feature space such that the concept point is as close to at least one instance from each positive bag as possible and far away from all instances from negative bags as possible. Diverse Density is defined as a measure of how close the positive bags and how far the negative bags are from the concept point. Therefore, the MI problem is to find such as concept point that maximizes the DD values. Gradient ascend approach is suggested by Maron and Lozano-Pérez (1998) to find the concept point with a strategy of starting with instances from each positive bag repetitively. EM-DD (Zhang and Goldman, 2002) is an EM version of the DD method. EM-DD assumes that there is a 'most representative' point in each bag capturing the label information of the bag. Since these 'most representative' points are unknown, they are estimated using EM-based approach, leading to EM-DD, a combination of EM-based approach with the DD method. EM-DD has a similar framework as $k-$means clustering. It starts with an initial concept estimated by the DD algorithm. In the E step, the current concept point is used to find the 'most representative' point for each bag. In the M step, a new concept point is estimated such that the DD is maximized and used to replace the current concept. EM-DD iterates the E and M steps until convergence. By finding the 'most representative' points, EM-DD converts the multiple instance problem to a single instance problem such that the computational complexity is reduced. Extended Functions of Multiple Instances (eFUMI) (Jiao and Zare, 2015) is another EM style MI algorithm that learns positive and negative concept points. Each instance is considered as a convex combination of positive and negative concepts. Multiple Instance Spectral Matched Filter (MI-SMF) and Multiple Instance Adaptive Cosine Estimator (MI-ACE) (Zare et al., 2017) maximize the ACE and SMF responses respectively by learning a discriminative positive concept. Instead of learning only one 'concept' point for the positive or negative bag, Multiple Instance Hybrid Estimator (MI-HE) (Jiao et al.,

2017) learns a set of 'concept' points to capture the variability within the bags. mi-SVM and MI-SVM are two SVM-based approaches aim at instance-level classification and bag-level classification, respectively (Andrews et al., 2003). For mi-SVM, all instances are accounted for estimating the margin. The margin is maximized with the constraint that at least one instance from each positive bag is in one halfspace and all instances in the negative bags are in the other halfspace. For MI-SVM, only the 'most representative' instances from all bags are used. The margin is defined by the 'most positive' instance from each positive bag and the 'least negative' instance from each negative bag. For all the IS paradigm based algorithm discussed above (ARP, DD, EM-DD and MI-SVM), the bag-level classifier can be the max rule, one of the aggregation rules used by many IS methods:

$$F(\mathbf{X}) = \max_{x \in X} f(\mathbf{x}) \tag{2–19}$$

where $\mathbf{x}$ denotes every instance in a bag $\mathbf{X}$ and $f(\mathbf{x})$ is a selected instance level classifier with an binary output (1 for positive and 0 for negative) for each instance $\mathbf{x}$.

The standard multiple instance problem assumes that the positive bag is defined if containing a class of instances that negative bags do not contain. However, there is a different multiple instance problem definition. For the alternative definition, instances from all classes can exist in both positive and negative bags. Each positive bag contains instances from more than one class while each negative bag contains instances from only a single class. Note that different negative bags may contain instances from a different class. For the alternative MI definition, the IS paradigm based methods have bad performances since it learns an instance-level classifier and both positive and negative bags may contain instances from every class. Thus, global, bag-level information becomes necessary for the alternative multiple instance definition.

BS paradigm learns a bag-level classifier. The most common approaches in the literature learn a distance function or a kernel function for pairwise bags. Since a bag is a set of instances in the feature space, distance or kernel metrics that compare two sets can be applied.

Related BS based MIL classification methods use minimal Hausdorff distance (Wang and Zucker, 2000), Earth Movers Distance (Zhang et al., 2007), the Chamfer distance (Belongie et al., 2002) and the kernel (Gärtner et al., 2002). Chamfer distance (Belongie et al., 2002) is one of the most widely used BS based approach. Let $\mathbf{X}$ and $\mathbf{Y}$ be two bags. Let $\mathbf{x} \in \mathbf{X}$ and $\mathbf{y} \in \mathbf{Y}$ be the corresponding instances in each bag. The Chamfer distance between bag $\mathbf{X}$ and $\mathbf{Y}$ is defined as:

$$D(\mathbf{X}, \mathbf{Y}) = \frac{1}{|\mathbf{X}|} \sum_{x \in X} \min_{y \in Y} \|(\mathbf{x} - \mathbf{y})\| + \frac{1}{|\mathbf{Y}|} \sum_{y \in Y} \min_{x \in X} \|(\mathbf{x} - \mathbf{y})\| \tag{2–20}$$

ES paradigm also considers the bag level information for multiple instance problem as BS paradigm. ES paradigm based approaches learn a mapping function which maps each bag to a feature vector. Then the bag-level classification is performed on the space of mapped feature vectors. The ES paradigm based methods can be categorized into two groups, non-vocabulary and vocabulary-based methods. For the non-vocabulary based methods, the instances in a bag are equally treated for mapping the bag to a feature vector. For the vocabulary based methods, the instances in a bag are unequally treated, for example, some prototypes are learned from the instances in the training set. Then the mapping of the bag is performed by comparing how similar between its instances and each of these prototypes.

The simplest non-vocabulary method is the Simple MI method proposed by Dong (2006). Simple MI maps each bag to the average of the instances inside the bag. Gärtner et al. (2002) proposed a max-min operator as the mapping function. Each bag is mapped to a 1-by-2d feature vector which is the concatenation of the max value of each dimension and min value of each dimension, where d is the dimensionality of the instances.

The idea of vocabulary methods is to discover what classes of instances present in the bag. A general vocabulary based approach contains three major steps: 1) building a vocabulary, i.e., by clustering into K clusters; 2) proposing a mapping function, which maps the bag into a 1-by-K feature vector, by comparing the instances in a bag with the concept (prototype) from each cluster; 3) classifying the feature vector in the embedded

$K-$dimensional space. Different vocabulary based approaches vary mainly on vocabulary generation and/or mapping function design.

The histogram-based bag-of-words method is a vocabulary based method (Nowak et al., 2006). First, the vocabulary is formed as K concepts by running a clustering method, i.e., k-means. These concepts are the corresponding average value of each cluster, a prototype for each class. Second, each instance in the testing bag is associated with its nearest concept from the K concepts. At last, each element of the mapped feature vector is the count of how many instances are assigned to each concept. The distance-based bag-of-words method is another vocabulary based method. For previous histogram-based methods, the Euclidean distance is used to find the nearest concept for each instance, i.e. the Multiple-Instance Learning via Embedded Instance Selection (MILES) method (Chen et al., 2006). However, some authors use Mahalanobis distance (Opelt et al., 2006) or Gaussian kernel (Serre et al., 2007). Another major difference between distance-based and histogram based methods is the mapping function. For distance-based methods, each element of the mapped feature vector is the minimum distance between instances in the bag and each concept. To be more specific, histogram based methods map the bag by checking how many instances fall into each class (a histogram), while distance based method map the bag by checking how far between the instances in the bag and each class (a distance).

2.1.7 Multiple Instance Multi-label (MIML) Labels

Multiple Instance Multi-label (MIML) is proposed by Zhou et al. (2012), which can be regarded as the combination of Multi-label learning and Multiple instance learning, aiming to solve the real complicated applications where the instance level label is not feasible, and the object has multiple semantic meanings. For instance, an image object can belong to classes of ocean and mountain simultaneously. Previously for image retrieval, the image is considered as an instance with multiple labels. But the user may only interested in the concept of ocean instead of a mountain. To choose the right semantic meaning, MIML considers the image patches as instances of the image object, where each patch can be classified as one of the

classes. In other words, the ocean patches and mountains patches can be learned from the image. Similar to MIL, annotators are fully trusted such that Equation 2–1 is satisfied. In MIL, each instance in a bag sharing the same one bag level label. However, in MIML, each instance in a bag sharing the same subset of bag level labels. More formally, the bag level labels for an instance in MIML satisfy Equation 2–6.

2.1.8 Regression Labels

For previous different types of uncertain labels, the classes for an instance are usually pre-defined fixed categories. However, the labels can also be a continuous value in some applications. For instance, in the application of age estimation or pose estimation (Yan et al., 2008), the age is usually a real number, e.g. 26.5 years old. Thus, these applications are to predict a regression value for new data. The corresponding training labels are defined as regression labels. However, there is still label uncertainty for regression labels, for example, the 26.5 years old is an approximate number, leading to some noise in the training set. Motivated by this type of label uncertainty, Yan et al. (2008) proposed to use a label interval as the regression labels. In other words, using label interval $(26, 27)$ years old instead of 26.5 years old. Thus, the label information ℓ_{ij}^{m} is defined as a continues real-value number or a real-value interval, shown in Equation 2–10 and Equation 2–11, respectively. If the label interval is used as the regression labels, the annotator's labeling reliability is assumed to be fully trusted, satisfying Equation 2–1. If the continuous value is used as the regression labels, the labeling reliability satisfies Equation 2–2. In literature, the label interval can be modeled as two inequality constraints of a non-linear regression solved by semidefinite programming (SDP) (Yan et al., 2008).

2.2 Hyperspectral Unmixing with Endmember Variability

Label imprecision is a common issue in remotely sensed imagery applications, such as hyperspectral unmixing and classification. There is a sort of imprecise labels in hyperspectral imagery, including 1) multiple instance labels, the high level label assigned to a group of pixels; 2) probabilistic labels, the confidence value for the labeling accuracy; 3) multi-labels, a couple

of co-existing labels assigned for each mixed pixel or each image patch, depending on the applications; and 4) noisy labels, wrong labels assigned because of bad imaging condition or less reliable annotators. In particular, the MI labels and probabilistic labels are studied in the thesis. Most conventional MI classification methods tend to estimate a "concept" instance (Maron and Lozano-Pérez, 1998; Zhang and Goldman, 2002) for the positive class, where one instance is generally not enough to capture the variability of feature vectors in the positive class. The most recent state-of-the-art approach estimate multiple "concept" instances for the positive class, which can alleviate the variability problem in MIL (Jiao et al., 2017). The drawback of learning a single "concept" instance (or a few) is more obvious in remotely sensed images. For instance, each tree species class in a hyperspectral image usually have a large spectral variation. In addition, the spectral difference among some species classes is very small. Thus, learning one "concept" instance for each species class may not be able to capture the overall features of each class, resulting in a weak classification/unmixing performance. In the literature, a number of approaches have been proposed to address spectral variability in hyperspectral unmixing. There are two categories for representing these approaches: endmembers as sets and endmembers as statistical distributions (Zare and Ho, 2014). The former category is based on the linear mixture model, and the latter category can be regarded as the stochastic mixing model. In the following subsections, methods that account for these two categories are reviewed, respectively. In this thesis, the variability will be incorporated in MIL with the point of view of endmembers as statistical distributions.

2.2.1 Hyperspectral Classification and Unmixing

Hyperspectral image (HSI), collected using a hyperspectral camera, is a stack of image planes, where each plane corresponds to radiances at a specific electromagnetic wavelength acquired over all pixels in a scene (Bioucas-Dias et al., 2012). The wavelength for each image plane can range from visible to near infrared (e.g. 0.4 μm to 2.5 μm). Since HSI is a three-dimensional data cube, each pixel location in HSI is a two-dimensional radiance spectrum vector (or called spectral signature). Due to the spatial resolution limit of the hyperspectral

camera, The spatial resolution of HSI is often low, resulting in mixed pixels. Mixed pixels are pixels composed of more than one pure material. Each pure material in a hyperspectral scene is called an endmember. Thus, the spectral signature of each mixed pixel can be regarded as a mixture of a number of endmembers in the scene. However, different mixed pixels may be consisted of a number of endmembers with different weights. The weight is called proportion, representing the fractional proportion of each endmember in the mixed pixel.

Hyperspectral classification is the process to assign a class label or multi-class labels to each pixel in the hyperspectral image, according to the applications and data. Single-label classification is usually assumed when the spatial resolution of the hyperspectral image is high, resulting in a majority of pure pixels in the scene. Multi-label classification is often assumed when spatial resolution is low, such that a majority of pixels are mixed pixels in the scene. Thus, one class label is not enough for mixed pixel. However, single-label classification can still be applied to mixed pixels for an approximated and simple classification.

Hyperspectral unmixing aims to decompose the mixed pixel signature into a set of endmembers with an associated proportion vector. Hyperspectral unmixing algorithms can be categorized into two groups based on the expected types of mixing, which are linear mixing models and nonlinear mixing models (Bioucas-Dias et al., 2012). Linear mixing models assume that pure materials are uniformly partitioned on the surface of the mixed pixels, and there is only macroscopic scattering on the surface. But nonlinear mixing assumes a more complicated case that distribution of materials can be nonuniform and there is microscopic, multiple scattering among materials in a mixed pixel.

2.2.2 Linear Mixture Model

Linear Mixture Model (LMM) is a series of algorithms assuming the linear mixing case in a hyperspectral image. Therefore, each pixel is well represented by a convex combination of several endmember signatures weighted by the associated proportions (and additive random noise in some scenarios). Suppose there are M endmembers in a hyperspectral image. The kth

endmember signature is denoted by e_k and the proportion vector of the kth endmember for this ith pixel is p_{ik}. The observed pixel signature $\mathbf{x}_i$ is represented by

$$\mathbf{x}_i = \sum_{k=1}^{M} \mathbf{e}_k p_{ik} + \epsilon_i \tag{2–21}$$

where ϵ_i is the error term accounting for noise.

2.2.3 Endmember Variability

Endmember variability models each endmember as a set or distribution instead of a single signature because endmember has variability. There are many reasons resulting in endmember variabilities such as different compositions or different lighting conditions. For instance, the left image in Figure 2-1 is the RGB image from the MUUFL hyperspectral dataset (Gader et al., 2013). The highlighted region contains a red roof building, which can be considered as an endmember called red roof. It can be assumed most of the building roof is pure pixels. However, in fact, different locations of red roof show different spectral signatures mainly because of the various illumination intensities (i.e. shadow). There are two manually picked red roof pixels as well as their associated spectral signatures shown on the right part of Figure 2-1. The spectral signature of the red roof pixel towards the sunlight has a higher reflectance magnitude than that of the pixel on the shadow location. In addition, the shapes of the two pixel signatures are slightly different. It is because of other factors resulting in endmember variabilities such as chemical composition ratio variability on the same material or different textures on the material surface.

Endmember has variability could simply because of how it is defined, depending on the applications. For example, the endmembers in the MUUFL dataset are normally regarded as red roof, grey roof, vegetation, asphalt and soil. However, for vegetation in the scene, it is composed of trees and grasses. In addition, there could be more than one tree species for trees and more than one grass species for grasses. Therefore, it is oversimplified to represent the endmember vegetation with only one endmember signature. In the thesis, endmembers are

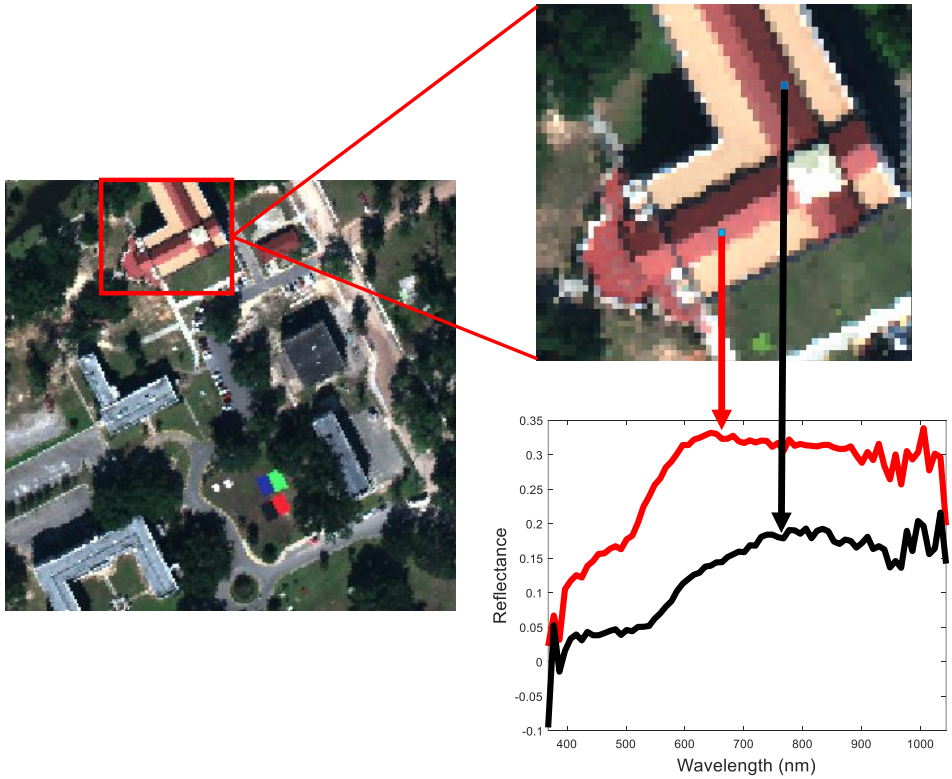


Figure 2-1. Spectral variability by illumination (Dataset: MUUFL hyperspectral dataset)

considered as statistical distributions to model the endmember variability. A more accurate and faster endmember variability algorithm will be proposed.

2.2.4 Endmembers as Sets

Some algorithms address spectral variability by constructing a set of spectral signatures for each endmember instead of a single spectral signature under the linear mixture model. The pixel signature can then be viewed as the convex combination of elements of endmember sets (one element from one set in most algorithms). From the perspective of the spectral library, the algorithms are further divided into two types, with or without a spectral library.

For algorithms given the spectral library, Multiple Endmember Spectral Mixture Models (MESMA), proposed by Roberts et al. (1998), and its extensions (Combe et al., 2008; Song, 2005; Asner et al., 2003) exhaustively search for one or more signatures for each endmember from the spectral library such that the estimated proportion values satisfy some predefined criteria. The signatures found, corresponding to each endmember, are grouped to form

the endmember sets. Also, some other algorithms rely on a given spectral library, such as endmember bundles (Bateson et al., 2000), band selection or weighting (Somers et al., 2011), and SVM unmixing (Mianji and Zhang, 2011; Bovolo et al., 2010)

For algorithms without the spectral library, the endmember sets are directly obtained from the pixels in the hyperspectral image. Automated endmember bundles (Somers et al., 2012) automatically selects a portion of pixels and applies an endmember extraction algorithm, for example, Vertex Component Analysis (VCA) (Nascimento and Dias, 2005), to estimate endmembers. This process is repeated several times and generates a number of endmembers. Then the endmember sets are obtained by applying a clustering algorithm, for instance, K-means (Lloyd, 1982), to these endmembers. After estimating the endmember sets, the proportion values can be estimated using any previous unmixing algorithms. Additionally, other algorithms, such as sparse unmixing (Castrodad et al., 2011) and local unmixing (Canham et al., 2011; Goenaga et al., 2013), can also learn the endmember sets from the hyperspectral image.

2.2.5 Endmembers as Distributions

Another type of method is based on modeling the endmember as a statistical distribution. Under the endmembers as statistical distributions approach, each endmember is considered as a statistical distribution rather than a set or a single value. Each sample from the endmember distribution can be regarded as a variant of the endmember signature. The variant is shown in Equation 2–22.

$$\mathbf{e}_k \sim \mathscr{F}\left(\cdot|\theta_k\right) \tag{2–22}$$

where $\mathscr{F}$ is the statistical distribution for the kth endmembers $\mathbf{e}_k$ and θ_k represents the unknown parameters for this distribution.

Therefore, for endmembers following the statistical distribution, each pixel is regarded as the convex combination of these distribution-based endmembers. To be more specific, the pixel signature $\mathbf{x}_i$ can be written as

$$\mathbf{x}_i = \sum_{k=1}^{M} p_{ik}\mathbf{e}_k \tag{2–23}$$

where p_{ik} is the proportion value of one variant of the endmember distribution $\mathbf{e}_k$ for pixel $\mathbf{x}_i$. Thus, compared with the standard linear mixture model, this category of models assumes that each pixel signature can be represented by a linear combination of variants from endmember distributions with associated proportion values.

From the perspective of the assumed distributions, the algorithms are further divided into several types including Normal Composition Model (NCM), Gaussian Mixture Model (GMM) which can be viewed as a variation of NCM, Beta Compositional Model (BCM) and Spatial Compositional Model (SCM).

2.2.5.1 Normal compositional model based algorithm

A large number of endmember-distribution based, unmixing algorithms are based on a Bayesian framework. Once the distributions are indicated, both the distribution parameters and proportion values can be estimated simultaneously. The most commonly used distribution to represent endmembers is the normal distribution, that is,

$$\mathscr{F}(\mathbf{e}_k|\theta_k) = \mathscr{N}(\mathbf{e}_k|\mu_k, \mathbf{\Sigma}_k) \tag{2–24}$$

where μ_k is the mean parameter and $\mathbf{\Sigma}_k$ is the covariance parameter for endmember $\mathbf{e}_k$. The corresponding model assuming normal distributions for endmembers is named the Normal Compositional Model (Stein, 2003). We assume the variants from endmember distributions are mutually independent normal distribution variables. According to Equation 2–22, 2–23 and 2–24, the pixel signature $\mathbf{x}_i$ under the NCM is represented as

$$\mathbf{x}_i \sim \mathscr{N}\left(\cdot \middle| \sum_{k=1}^{M} p_{ik}\mu_k, \sum_{k=1}^{M} p_{ik}^2\mathbf{\Sigma}_k\right) \tag{2–25}$$

A number of techniques for addressing spectral variability by assuming a normal compositional model have been developed.

Parameter estimation was addressed initially by Stein (2003) with a method based on the nested stochastic expectation maximization (SEM) algorithm (Diebolt and Ip, 1996). The proportion values p_{ik} are regarded as latent, hidden variables. The complete likelihood function is

$$p(\mathbf{x}_1, ..., \mathbf{x}_N, p_{11}, ..., p_{1M}, ..., p_{N1}, ..., p_{NM} | \{\mu_k, \mathbf{\Sigma}_k\}) \tag{2–26}$$

$$= \prod_{i=1}^{N} \mathcal{N}(\mathbf{p}_i; \mu(\mathbf{p}_i), \mathbf{\Sigma}(\mathbf{p}_i)) p(p_{i1}, ..., p_{iM}) \tag{2–27}$$

where N is the number of pixels and M is the number of endmembers in the data set.

$$\mu(\mathbf{p}_i) = c\mu_0 + \sum_{k=1}^{M} p_{ik}\mu_k \tag{2–28}$$

$$\mathbf{\Sigma}(\mathbf{p}_i) = c\mathbf{\Sigma}_0 + \sum_{k=1}^{M} p_{ik}^2 \mathbf{\Sigma}_k \tag{2–29}$$

It first chooses initial distribution parameters and latent proportion values. There are two types of distribution parameters, Gaussian means and Gaussian covariance matrices. The Gaussian means are initialized by applying an endmember extraction method, for example, VCA. The covariance matrices are initialized as the sample covariance matrices of clusters of points that are nearest to the endmember. It then iterates between E and M steps, sampling latent proportions values and maximizing likelihood function until convergence. Specifically, in the E step, the likelihood function value is (re-)calculated with the updated distribution parameters and proportion values. In the M step, the proportion values are sampled such that the likelihood functions are maximized. Besides, the distribution parameters are updated with a gradient descent method.

Eches et al. (2010a,b) and Kazianka (2012) suggested using a Markov Chain Monte Carlo (MCMC) sampler to estimate proportion values and endmember covariances under NCM given endmember mean signatures or a spectral library.

An example of NCM based MCMC method with given endmember mean signatures (Eches et al., 2010a) is presented below. It generates samples distributed according to the

joint posterior of proportion values, endmember variance values and one hyperparameter. The endmember extraction algorithm such as VCA or N-FINDR, is applied first to estimate the endmember means. Then, the endmember covariances and proportion values are estimated using MCMC. Several parameter priors are imposed. The Dirichlet prior is assigned to the proportions to enforce the non-negative and sum-to-one constraints, that is, $\mathbf{p}_n \sim \mathscr{D}(\cdot|\mathbf{1})$. The covariance matrix of each endmember can be written as $\sigma^2\mathbf{I}_L$, where $\mathbf{I}_L$ is the $L \times L$ identity matrix and σ^2 is the endmember variance in any spectral band. A conjugate inverse Gamma distribution is imposed on the variance σ^2 as the prior, given by $\sigma^2|\delta \sim \mathscr{IG}(\nu, \delta)$, where ν and δ are two user-defined hyperparameters (shape and scale parameters). A non-informative Jeffreys' prior is assigned to the δ, which is $f(\delta) \propto \frac{1}{\delta}\mathbf{1}_{R^+}(\delta)$.

The parameter estimation of this method contains two major steps, initialization and sampling using a hybrid Gibbs sampler. In the initialization step, the proportion values are initialized from a uniform distribution and normalized to be sum-to-one. The variance values are initialized from the pdf of an inverse Gamma distribution, $\sigma^2|\delta \sim \mathscr{IG}(\nu, \delta)$. Then the scale parameters are initialized from the pdf of non-informative Jeffreys' prior, $f(\delta) \propto \frac{1}{\delta}\mathbf{1}_{R^+}(\delta)$. In the sampling step, the proportion values are suggested to be sampled using a Metropolis-within-Gibbs algorithm such that the non-negative and sum-to-one constraints are both satisfied. The variance values (of the covariance matrices) are sampled following an inverse-Gamma distribution, which is

$$\sigma^2|\mathbf{x}, \mathbf{p}, \delta \sim \mathscr{IG}(\frac{L}{2} + 1, \frac{\|\mathbf{x} - \mu(\mathbf{p})\|^2}{2c(\mathbf{p})} + \delta) \tag{2–30}$$

For scale parameter δ, it is sampled from a Gamma distribution, which is

$$\delta|\sigma^2 \sim \mathscr{G}(1, \frac{1}{\sigma^2}) \tag{2–31}$$

where $\mathscr{G}(a, b)$ is the Gamma distribution with shape parameter a and scale parameter b.

Zare and Gader (2010) and Zare et al. (2013) presented MCMC sampler approach to estimate endmember spectral means and proportion values given endmember covariances under an NCM model.

In terms of hyperspectral data which are usually nonconvex, Zare et al. (2013) proposed to use several convex regions instead of a single convex region to represent the whole data set, motivated by the observation that hyperspectral data is usually nonconvex. The algorithm can automatically determine the number of endmember distribution sets by sampling from a Dirichlet process. Each set is viewed as a random simplex, where each vertex is modeled as an endmember distribution under the normal compositional model. The pixels are then divided into different sets according to the convex regions using a Dirichlet process prior. The Metropolis-within-Gibbs sampler is applied to divide the data set into convex regions with the learned number of regions and estimate the endmember distributions and proportion values for each convex region.

The endmember distribution covariances are assumed to be known in advance and data means are assumed to be drawn from the normal distributions $\mu_k \sim \mathcal{N}(\cdot|\mathbf{m}, \mathbf{C})$. The proportion vector for pixel $\mathbf{x}_i$ is modeled as $\mathbf{p}_i \sim \mathcal{D}(\cdot|\mathbf{1})$. The $\mathbf{m}$ and $\mathbf{C}$ are hyperparameters defining the prior distribution on the endmember means.

The Sampling Piecewise Convex Unmixing and Endmember Estimation (S-PCUE) starts with the initialization of the endmember means (the covariance matrices are given) of each convex set using VCA and proportion values of each pixel by drawing from a uniform Dirichlet distribution. Then S-PCUE iterates to sample the proportions for each pixel for each set of endmember using a Metropolis-Hasting step, sample each endmember mean in each set using another Metropolis-Hasting step and sample the hyperparameters for endmember means. Additionally, in each iteration, K new potential partitions, consisting of K new sets of endmember distributions and proportion values, are sampled. After sampling the new partitions, the DP partition probabilities for each pixel are calculated, which determines if the

convex set the pixel belongs to should be changed to a new partition or existing partition. The S-PCUE iterates to sample all these parameters until converging.

Some other approaches also address the NCM based hyperspectral unmixing problem. Zhang et al. (2014) introduced a particle swarm optimization expectation maximization (PSO-EM) method to estimate the endmember spectral means, endmembers covariances and proportion values. Zou and Zare (2017) introduced the Partial Membership Latent Dirichlet Allocation (PM-LDA) unmixing approach to estimate all endmember distributions and proportion values under the NCM while leveraging spatial information. Additionally, Gaussian Mixture Model (GMM) can be viewed as an extension of NCM (Zhou et al., 2018). GMM uses Gaussian Mixtures to model the underlying endmember distributions, motivated by the observation that the distribution of spectra from a material may be multi-modal.

2.2.5.2 Spatial compositional model based algorithm

Zhou et al. (2016) relaxes the assumption in NCM that the pixels are independent random variables and proposed Spatial Compositional Model (SCM). The authors defined a new concept named endmember uncertainty, similar but different than endmember variability, to model the error of endmembers. Since the pixels are not assumed to be independent, compared to standard NCM, the full likelihood of the pixels are estimated to obtain the endmember uncertainty. The author also applied a smoothness term that works locally to promote spatial similarity. To be more specific, a Markov Random Field (MRF) prior is assumed on the proportion values to drive the neighboring pixels to be similar. The parameter estimation of SCM is accomplished by maximizing the posteriori using the block coordinate descent method. However, the SCM results are sensitive to the weighting parameters on the spatial similarity term, i.e., a badly tuned weighting parameter may result in over-smoothing or under-smoothing.

2.2.5.3 Beta compositional model based algorithm

Under Beta Compositional Model (BCM), proposed by Du et al. (2014), each endmember is a random variable distributed according to a beta distribution, motivated by the observation

that the underlying endmember distribution may be skewed. The authors found that the beta distributions, for some hyperspectral datasets, have a better fit than normal distributions by comparing the quantile-quantile plots between the assumed distributions and hyperspectral data. There are two types of BCM algorithm, which are BCM-spectral and BCM-spatial. Both methods require an initial step to identify the pixels that have similar proportion values. The BCM-spectral method only considers the spectral information in the initial step while the BCM-spatial utilizes both the spatial and spectral information. The BCM-spectral is solved by quadratic programming (QP). The proportion values are estimated by minimizing the difference between the original and reconstructed data means. BCM-spatial uses a Metropolis-Hasting (MH) sampler to estimate the unknown unmixing parameters. The proportion values are estimated by minimizing the difference between both the original and reconstructed data means and variances. For both algorithms, the endmember distributions are estimated using the Maximum Likelihood (ML) method.

CHAPTER 3
PROPOSED ALGORITHM

Most state-of-the-art concept-based multiple instance learning methods (Maron and Lozano-Pérez, 1998; Zhang and Goldman, 2002; Jiao and Zare, 2015; Zare et al., 2017; Jiao et al., 2017) are modeled to estimate a positive instance concept, $\mathbf{s} \in \mathbb{R}^{1\times d}$, representing the most positive concept that is the closest to positive instances and the farest from negative instances (Maron and Lozano-Pérez, 1998; Zhang and Goldman, 2002; Jiao and Zare, 2015), or a discriminative concept, $\mathbf{s} \in \mathbb{R}^{1\times d}$, characterizing the difference between true positive and negative instances (Zare et al., 2017; Jiao et al., 2017), where d is the dimensionality of the data.This type of MIL methods can be called single instance estimation.

From the point of view of considering a bag as the sampling of the underlying true positive and/or negative distributions, instead of a finite set of fixed elements (Foulds and Frank, 2010), single instance estimation algorithms assume that one concept, $\mathbf{s}$, is enough to denote the whole distribution of true positive instances. The performance of single-instance estimation is satisfying if the underlying distribution of true positive instances is a simple Gaussian distribution, of which covariance has only non-zero and same diagonal values.

$$p\left(\mathbf{x}\right) \sim \mathcal{N}\left(\mathbf{x}|\mathbf{s}, \sigma^2\mathbf{I}\right) \tag{3–1}$$

where $\mathbf{x}$ is the random variable of true positive distribution, σ is the non-zero diagonal value and $\mathbf{I}$ is the identity matrix.

However, the assumption is often too ideal since the underlying distribution of true positive instances is usually multi-modal and has a non-sparse covariance. For example, in the tree species classification of hyperspectral imagery, if the PI genus is the positive class and the QU genus is the negative class, a single-instance estimation algorithm can learn one instance prototype to present the PI genus or the differences between PI and QU. However, the PI genus is a collection of several PI species, including PIPA, PIEL and PITA species, where

each species has its own distribution as is shown in Figure 3-1. In other words, the PI positive instances can be considered as a mixture of several distributions.

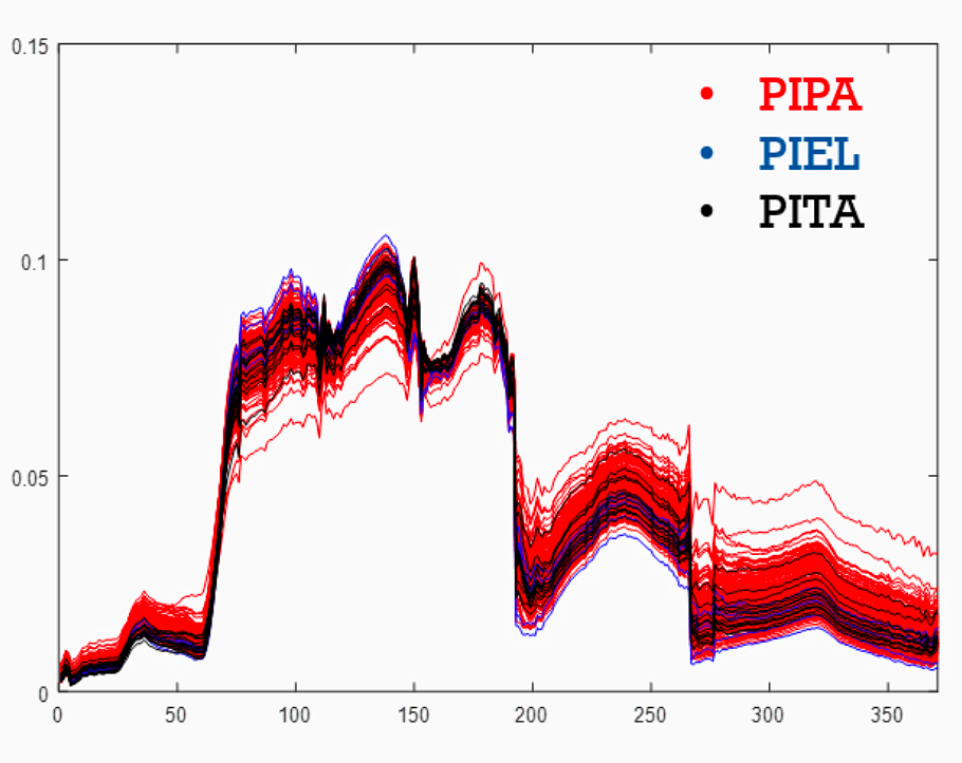


Figure 3-1. Spectral variability of PI positive instances

Therefore, MIL was recently extended to be able to learn more than one positive/discriminative concepts (Jiao et al., 2018), $\mathbf{S} \in \mathbb{R}^{c \times d}$, where c is the number of concepts, when the distribution of true positive instances is multi-modal such that one instance prototype is not enough to present the whole distribution of true positive instances. However, these methods are still inherently single-instance algorithms, for each modal/component of the distribution is still modeled by one concept and the covariance information is ignored.

In this work, a probabilistic MIL with distribution (pMILd) algorithm is proposed to model the distribution of true positive instances, $p(\mathbf{x})$, and generalize the positive bag labels from deterministic ($B_j^+ = 1$) to be probabilistic ($B_j^+ \in (0, 1]$).

$$p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|\mu, \mathbf{C}) \tag{3–2}$$

$$p(\mathbf{x}) \sim \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{C}_k) \tag{3–3}$$

A standard Gaussian model in Equation 3–2 and a Gaussian Mixture Model (GMM) in Equation 3–3 for MIL are proposed, respectively, where μ and $\mathbf{C}$ are the mean and covariance of the target distribution for the standard Gaussian model, μ_k, $\mathbf{C}_k$ and α_k are the mean, covariance and mixing weight of k-th target distribution for the GMM model. Additionally, the inference of distribution parameters based on Expectation Maximization (EM) is proposed.

Let $B^+ = \{B_1^+, ..., B_{N^+}^+\}$ be the positive bags where N^+ is the number of positive bags. Let $B^- = \{B_1^-, ..., B_{N^-}^-\}$ be the negative bags where N^- is the number of negative bags. Let $\mathbf{x}_{ij}$ be the i-th instance in the positive bag B_j. Let z_{ij} be the true instance-level label for $\mathbf{x}_{ij}$, which can be 1 (true positive) or 0 (true negative). Let $\mathbf{x}_{mn}$ be the m-th instance in the negative bag B_n.

Let $P(B_j^+)$ be the probabilistic label assigned to the positive bag. The parameters associated with the positive distribution are defined as Θ, which are unknown model parameters and depend on the selection of distribution type. According to the different type of assumed distribution, $\Theta = \{\mu, \mathbf{C}\}$ for Gaussian target distribution or $\Theta = \{\mu_k, \mathbf{C}_k, \alpha_k\}_{k=1:K}$ for GMM target distribution. Thus, the likelihood of each positive instance with respect to the positive distribution can be written as $p(\mathbf{x}_{ij}|\Theta)$. The pMILd algorithm is proposed to maximize the complete log-likelihood function in Equation 3–5.

$$\log p(\mathbf{X}, \mathbf{Z}|\Theta) = \ln \prod_{j=1}^{N^+} \prod_{\mathbf{x}_{ij} \in B_j^+} p(\mathbf{x}_{ij}|\Theta)^{z_{ij}} \tag{3–4}$$

$$= \sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij} \in B_j^+} z_{ij} \ln p(\mathbf{x}_{ij}|\Theta) \tag{3–5}$$

3.1 Distribution Parameters Estimation for a Target as a Gaussian Distribution

If the target distribution is assumed to be a Gaussian distribution, the complete log-likelihood is shown in Equation 3–6.

$$\log p(\mathbf{X}, \mathbf{Z}|\Theta) = \ln \prod_{j=1}^{N^+} \prod_{x_{ij} \in B_j^+} \left(\frac{1}{\sqrt{(2\pi)^d |\mathbf{C}|}} \right)^{z_{ij}} e^{-\frac{z_{ij}}{2}(x_{ij}-\mu)^T C^{-1}(x_{ij}-\mu)} \tag{3–6}$$

Equation 3–6 can be simplified to Equation 3–7.

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{Z}|\Theta) = &- \frac{d}{2} \left(\sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} z_{ij} \right) \ln (2\pi) + \frac{1}{2} \left(\sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} z_{ij} \right) \ln |\mathbf{C}^{-1}| \\ &- \frac{1}{2} \left(\sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} z_{ij} (\mathbf{x}_{ij} - \mu)^T \mathbf{C}^{-1} (\mathbf{x}_{ij} - \mu) \right) \end{aligned} \tag{3–7}$$

The model is optimized by the EM algorithm. Firstly, the expected complete log-likelihood can be shown as Equation 3–8.

$$\begin{aligned} &E_Z[log p(\mathbf{X}, \mathbf{Z}|\Theta)] \\ =&E_Z \Bigg[- \frac{d}{2} \left(\sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} z_{ij} \right) \ln (2\pi) + \frac{1}{2} \left(\sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} z_{ij} \right) \ln |\mathbf{C}^{-1}| \\ &- \frac{1}{2} \left(\sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} z_{ij} (\mathbf{x}_{ij} - \mu)^T \mathbf{C}^{-1} (\mathbf{x}_{ij} - \mu) \right) \Bigg] \end{aligned} \tag{3–8}$$

Equation 3–8 can be simplified to Equation 3–9.

$$E[log p(\mathbf{X}, \mathbf{Z}|\Theta)] = -\frac{d}{2} h \ln(2\pi) + \frac{1}{2} h \ln |\mathbf{C}^{-1}| - \frac{1}{2} \left(tr \left(V \mathbf{C}^{-1} \right) \right) \tag{3–9}$$

where $V = \sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)}) (\mathbf{x}_{ij} - \mu) (\mathbf{x}_{ij} - \mu)^T$ represents the weighted empirical covariance for the target distribution and $h = \sum_{j=1}^{N^+} \sum_{x_{ij} \in B_j^+} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)})$ is the cumulative confidence all positive instances to be true positive instances.

In this dissertation, the proposed true target probability $P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)})$ is assumed to have the two forms, based on two metrics, including Euclidean distance 3–11 and Mahalanobis distance 3–10.

$$P\left(z_{ij}|\mathrm{x}_{ij},\Theta^{(t-1)}\right)=\begin{cases} P(z_{ij}=1|\mathrm{x}_{ij}\in B_j^+,\Theta^{(t-1)})P(B_j^+)=\max\left[0,P(B_j^+)\left(e^{-\beta_1(\mathrm{x}_{ij}-\mu)^T\mathrm{C}^{-1}(\mathrm{x}_{ij}-\mu)}-e^{-\beta_2(\mathrm{x}_{ij}-\mu_\mathrm{b})^T\mathrm{C}_b^{-1}(\mathrm{x}_{ij}-\mu_\mathrm{b})}\right)\right] \\ P(z_{ij}=0|\mathrm{x}_{ij}\in B_j^+,\Theta^{(t-1)})P(B_j^+)=1-\max\left[0,P(B_j^+)\left(e^{-\beta_1(\mathrm{x}_{ij}-\mu)^T\mathrm{C}^{-1}(\mathrm{x}_{ij}-\mu)}-e^{-\beta_2(\mathrm{x}_{ij}-\mu_\mathrm{b})^T\mathrm{C}_b^{-1}(\mathrm{x}_{ij}-\mu_\mathrm{b})}\right)\right] \end{cases} \tag{3–10}$$

$$P\left(z_{ij}|\mathrm{x}_{ij},\Theta^{(t-1)}\right)=\begin{cases} P(z_{ij}=1|\mathrm{x}_{ij}\in B_j^+,\Theta^{(t-1)})P(B_j^+)=\max\left[0,P(B_j^+)\left(e^{-\beta_1(\mathrm{x}_{ij}-\mu)^T(\mathrm{x}_{ij}-\mu)}-e^{-\beta_2(\mathrm{x}_{ij}-\mu_\mathrm{b})^T(\mathrm{x}_{ij}-\mu_\mathrm{b})}\right)\right] \\ P(z_{ij}=0|\mathrm{x}_{ij}\in B_j^+,\Theta^{(t-1)})P(B_j^+)=1-\max\left[0,P(B_j^+)\left(e^{-\beta_1(\mathrm{x}_{ij}-\mu)^T(\mathrm{x}_{ij}-\mu)}-e^{-\beta_2(\mathrm{x}_{ij}-\mu_\mathrm{b})^T(\mathrm{x}_{ij}-\mu_\mathrm{b})}\right)\right] \end{cases} \tag{3–11}$$

where μ_b and $\mathbf{C}_b$ are the sample mean and sample covariance matrix of training instances over all negative bags.

Then the update equations for unknown variables $\Theta = \{\mu, \mathbf{C}\}$ are shown in the Equation 3–12 and 3–13, by taking the derivative of the expectation of the complete log-likelihood with respect to μ and $\mathbf{C}$, respectively.

$$\mu = \frac{\sum_{j=1}^{N^+}\sum_{\mathrm{x}_{ij}\in B_j^+} P(z_{ij}=1|\mathbf{x}_{ij},\Theta^{(t-1)})\mathbf{x}_{ij}}{\sum_{j=1}^{N^+}\sum_{\mathrm{x}_{ij}\in B_j^+} P(z_{ij}=1|\mathbf{x}_{ij},\Theta^{(t-1)})} \tag{3–12}$$

$$\mathbf{C} = \frac{\sum_{j=1}^{N^+}\sum_{\mathrm{x}_{ij}\in B_j^+} P(z_{ij}=1|\mathbf{x}_{ij},\Theta^{(t-1)})(\mathbf{x}_{ij}-\mu)(\mathbf{x}_{ij}-\mu)^T}{\sum_{j=1}^{N^+}\sum_{\mathrm{x}_{ij}\in B_j^+} P(z_{ij}=1|\mathbf{x}_{ij},\Theta^{(t-1)})} \tag{3–13}$$

Thus the EM optimization of the proposed target distribution as single Gaussian is estimated by iterating between an E step (using Equation 3–9) and M step (using Equation 3–12 and 3–13).

3.2 Distribution Parameter Estimation for a Target as a GMM Distribution

If the target distribution is assumed to be a GMM, there are K number of Gaussian components in the target GMM. Therefore, K sets of mean and covariance matrices are

required to be estimated, denoted by μ_k and $\mathbf{C}_k$, respectively, where $k = 1 : K$. Additionally, two more unknown parameters, γ_{ijk} and α_k, to be estimated. The first parameter, γ_{ijk}, usually called the responsibility, denotes the membership of the jth instance in ith positive bag with respect to the kth Gaussian component. The second parameter, α_k, often called the mixing coefficient, represents the mean of memberships of all target instances with respect to the kth Gaussian component. In the proposed model, the update equations for γ_{ijk} and α_k follow the standard GMM, shown in Equation 3–14 and 3–15.

$$\gamma_{ijk} = \frac{\alpha_k p\left(\mathbf{x}_{ij}|\Theta_k^{(t-1)}\right)}{\sum_{k=1}^{K} \alpha_k p\left(\mathbf{x}_{ij}|\Theta_k^{(t-1)}\right)} \tag{3–14}$$

$$\alpha_k = \frac{\sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij}} \gamma_{ijk}}{\sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij}} 1} \tag{3–15}$$

Therefore, all unknown parameters for the proposed target as GMM distribution can be presented by $\Theta = \{\mu_k, \mathbf{C}_k, \gamma_{ijk}, \alpha_k\}$. Similarly, the EM update equations for μ_k and $\mathbf{C}_k$ are in Equation 3–16 and 3–17.

$$\mu_k = \frac{\sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij} \in B_j^+} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)}) \gamma_{ijk} \mathbf{x}_{ij}}{\sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij} \in B_j^+} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)}) \gamma_{ijk}} \tag{3–16}$$

$$\mathbf{C}_k = \frac{\sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij} \in B_j^+} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)}) \gamma_{ijk} (\mathbf{x}_{ij} - \mu_k)(\mathbf{x}_{ij} - \mu_k)^T}{\sum_{j=1}^{N^+} \sum_{\mathbf{x}_{ij} \in B_j^+} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)}) \gamma_{ijk}} \tag{3–17}$$

where the target probability $P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)})$ is also assumed to have two forms, based on two metrics, including Euclidean distance 3–19 and Mahalanobis distance 3–18.

$$P\left(z_{ij}|\mathbf{x}_{ij}, \Theta^{(t-1)}\right) = \begin{cases} P(z_{ij}=1|\mathbf{x}_{ij} \in B_j^+, \Theta^{(t-1)}) P(B_j^+) = \max\left[0, P(B_j^+)\left(e^{-\beta_1 \min_k \left[(\mathbf{x}_{ij}-\mu_k)^T C_k^{-1} (\mathbf{x}_{ij}-\mu_k)\right]} - e^{-\beta_2 \min_q \left[(\mathbf{x}_{ij}-\mu_q)^T C_q^{-1} (\mathbf{x}_{ij}-\mu_q)\right]}\right)\right] \\ P(z_{ij}=0|\mathbf{x}_{ij} \in B_j^+, \Theta^{(t-1)}) P(B_j^+) = 1 - P(z_{ij}=1|\mathbf{x}_{ij} \in B_j^+, \Theta^{(t-1)}) P(B_j^+) \end{cases} \tag{3–18}$$

$$P\left(z_{ij}|\mathrm{x}_{ij},\Theta^{(t-1)}\right)=\begin{cases} P(z_{ij}=1|\mathrm{x}_{ij}\in B_j^+,\Theta^{(t-1)})P(B_j^+)=\max\left[0,P(B_j^+)\left(e^{-\beta_1\min_k\left[(\mathrm{x}_{ij}-\mu_k)^T(\mathrm{x}_{ij}-\mu_k)\right]}-e^{-\beta_2\min_q\left[(\mathrm{x}_{ij}-\mu_q)^T(\mathrm{x}_{ij}-\mu_k)\right]}\right)\right] \\ P(z_{ij}=0|\mathrm{x}_{ij}\in B_j^+,\Theta^{(t-1)})P(B_j^+)=1-\max\left[0,P(B_j^+)\left(e^{-\beta_1\min_k\left[(\mathrm{x}_{ij}-\mu_k)^T(\mathrm{x}_{ij}-\mu_k)\right]}-e^{-\beta_2\min_q\left[(\mathrm{x}_{ij}-\mu_q)^T(\mathrm{x}_{ij}-\mu_k)\right]}\right)\right] \end{cases} \tag{3–19}$$

where it is assumed that there are Q number of Gaussian mixing components for the non-target distribution of negative bags. Let μ_q and $\mathbf{C}_q$ be the mean and covariance matrix for the qth ($q = 1 : Q$) component of the non-target distribution, which can be simply solved by a standard EM.

The update equations for μ_k and $\mathbf{C}_k$ in Equation 3–16 and 3–17 can be viewed as a generalization of the target distribution as a single Gaussian, where the probability of a positive instance $\mathbf{x}_{ij}$ to be a true positive is generalized from $P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)})$ to $P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)})\gamma_{ijk}$. If $\gamma_{ijk} = 1$ for a specific kth component, the GMM downgrades to a single Gaussian model.

The target probability, $P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta^{(t-1)})$, for a target as a GMM is different from a target as a Gaussian since there are possibly multiple components for both the target and non-target distributions. Therefore, the distance to the target/non-target distributions is inferred from the distance to each component of the target/non-target distributions in the proposed method. In the proposed method, this distance is defined as the minimum distance among distances of one instance with respect to all target components or non-target components.

3.3 Classifier Parameter Estimation

After learning the unknown parameters, Θ, for either a target as a single Gaussian or a target as a GMM, the next step is to learn a classifier that is capable of classifying a test bag as either positive or negative. A threshold-based MIL classifier is proposed, including instance-level confidence estimation, bag-level confidence estimation and classification threshold estimation. The process uses the training data itself to estimate these classifier parameters.

3.3.1 Instance-level Confidence Estimation

The first step of training a classifier is to assign target confidence/probability for each training sample, or instance-level confidence estimation. Specifically, the target probability for $\mathbf{x}_{ij}$ is estimated by Equation 3–11 (Mahalanobis-based) and 3–10 (Euclidean-based) for a target as a Gaussian or by Equation 3–18 (Mahalanobis-based) and 3–19 (Euclidean-based) for a target as a GMM, where the distribution parameters, Θ, have been estimated from the proposed EM-based method.

3.3.2 Bag-level Confidence Estimation

The second step of training is to infer the target confidence/probability to each training bag, called bag-level confidence estimation, since the final goal in this dissertation is to classify the MIL bags instead of instances. The bag-level confidence values can be aggregated/inferred from instance-level confidence values with several different proposed strategies to choose from based on the application.

The first strategy is taking the maximum of instance-level confidence values. This strategy follows the strict definition of MIL that a bag is called positive if there is at least one true positive instance in the bag. However, it is sensitive to outliers and noise. For instance, an outlier in a negative bag can have high instance confidence, resulting in a high bag-level confidence for the negative bag.

$$P\left(+|B_i\right) = \max_j P\left(z_{ij} = 1|\mathbf{x}_{ij}, \Theta\right), \forall i \tag{3–20}$$

The second strategy is confidence averaging. Specifically, the bag-level target confidence is the mean value of all instance-level confidence values in this bag. This strategy does not follow the strict definition of MIL, but it works well for certain scenarios. For example, when a large portion of instances in the positive bags are true positive instances and a large portion of instances in the negative bags are true negative instances.

$$P(+|B_i) = \frac{\sum_{x_{ij} \in B_i} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta)}{\sum_{x_{ij} \in B_i} 1}, \forall i \tag{3–21}$$

The third strategy is confidence averaging on selected high instance confidence values. First, the instances in a bag are sorted in descending order based on their instance-level confidence values. Then, the top η confidence values are selected, resulting in a subset $B_{i,\eta}$. Lastly, the bag-level confidence value is the average of these selected η instance-level confidence values. This strategy can be viewed as a transition between the previous two strategies. It follows the definition of MIL to some extent, but is robust to outliers/noise due to the averaging.

$$P(+|B_i) = \frac{\sum_{x_{ij} \in B_{i,\eta}} P(z_{ij} = 1|\mathbf{x}_{ij}, \Theta)}{\sum_{x_{ij} \in B_{i,\eta}} 1}, \forall i \tag{3–22}$$

3.3.3 Classification Threshold Estimation

Once the bag-level confidence values are estimated, the final step is to estimate a threshold value τ to separate the positive and negative bags based on their bag-level confidence values. The classification threshold is determined by selecting a threshold value τ such that the misclassification rate is minimized:

$$\arg\min_{\tau} \left(FP(\tau) + FN(\tau)\right) \tag{3–23}$$

where FP is the number of negative training bags that are misclassified as positive and the FN is the number of positive training bags that are misclassified as negative with τ.

Let **B** and **Y** be the training bags and training labels, respectively. The pseudo code of the proposed target as a single Gaussian method is shown in the Algorithm. 3.1. The pseudo code of the proposed target as a GMM method is shown in the Algorithm. 3.2.

Algorithm 3.1. *MIL training phase of the target as a single Gaussian algorithm*

Input: **B**, **Y**

1: *Initialize* $\Theta = \{\mu, \mathbf{C}\}$
2: ***if*** *stoping criteria is not met* ***then***

3: E step:
4: *Update the target probability* $P\left(z_{ij}|\mathbf{x}_{ij}, \Theta^{(t-1)}\right)$ *for each positive instance using Equation 3–10 or 3–11*
5: *Update the expectation using Equation 3–9.*
6: M step:
7: *Update* μ *of target Gaussian distribution using Equation 3–12*
8: *Update* $\mathbf{C}$ *of target Gaussian distribution using Equation 3–13*
9: ***end if***
10: ***Instance-level confidence estimation****: Update the target probability* $P\left(z_{ij}|\mathbf{x}_{ij}, \Theta^{(t-1)}\right)$ *for each positive or negative instance using Equation 3–10 or 3–11*
11: ***Bag-level confidence estimation****: Generate the bag-level confidence value from the instance-level confidence values within each positive or negative bag using Equation 3–20 or Equation 3–21 or Equation 3–22.*
12: ***Classification threshold estimation****: Estimate the threshold* τ *using Equation 3–23.*
Output: Θ, τ

Algorithm 3.2. *MIL training phase of the target as a GMM algorithm*

Input: $\mathbf{B}$, $\mathbf{Y}$
1: *Initialize the* $\Theta = \{\mu_k, \mathbf{C}_k, \gamma_{ijk}, \alpha_k\}$
2: ***if*** *stoping criteria is not met* ***then***
3: E step:
4: *Update the target probability* $P\left(z_{ij}|\mathbf{x}_{ij}, \Theta^{(t-1)}\right)$ *for each positive instance using Equation 3–19 or 3–18*
5: *Update the responsibility,* γ_{ijk} *using Equation 3–14*
6: *Update the expectation using Equation 3–9.*
7: M step:
8: *Update* μ_k *of k-th component of target GMM distribution using Equation 3–16*
9: *Update* $\mathbf{C}_k$ *of k-th component of target GMM distribution using Equation 3–17*
10: *Update* α_k *using Equation 3–15*
11: ***end if***
12: ***Instance-level confidence estimation****: Update the target probability* $P\left(z_{ij}|\mathbf{x}_{ij}, \Theta^{(t-1)}\right)$ *for each positive or negative instance using Equation 3–10 or 3–11*
13: ***Bag-level confidence estimation****: Generate the bag-level confidence value from the instance-level confidence values within each positive or negative bag using Equation 3–20 or Equation 3–21 or Equation 3–22.*
14: ***Classification threshold estimation****: Estimate the threshold* τ *using Equation 3–23.*
Output: Θ, τ

3.4 Testing Phase of Proposed Method

Once the distribution parameters, Θ, and classification threshold, τ, are estimated in the learning phase following Algorithm 3.1 for a target as a single distribution or Algorithm 3.2 for a target as a GMM, the goal of testing is to classify any unknown bags as either target or

non-target. Let $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{Y}}$ be the testing bags and their associated unknown labels. The pseudo code of the testing phase of the proposed method is shown in the Algorithm 3.3.

Algorithm 3.3. *MIL testing phase of the proposed algorithm*

Input: $\tilde{\mathbf{B}}$, Θ, τ

1: ***Instance-level confidence estimation***: *Calculate the target probability* $P\left(z_{ij}|\mathbf{x}_{ij}, \Theta^{(t-1)}\right)$ *for each positive or negative instance of each bag of* $\tilde{\mathbf{B}}$ *using Equation 3–10 or 3–11*

2: ***Bag-level confidence estimation***: *Generate the bag-level confidence value from the instance-level confidence values within each positive or negative bag using Equation 3–20 or Equation 3–21 or Equation 3–22.*

3: ***Bag classification***: *Classify each bag of* $\tilde{\mathbf{B}}$ *as target if its bag-level confidence is larger than the threshold* τ *or as non-target if its bag-level confidence is smaller than the threshold* τ

Output: $\tilde{\mathbf{Y}}$

3.5 Multi-class MIL Classification

The proposed pMILd method follows a standard MIL assumption that there are only two classes (positive and negative class). Distribution parameters Θ is estimated to characterize the underlying distribution of true positive samples, as discussed in the section 3.1 and 3.2. In addition, the threshold value τ is estimated to classify the testing bag into either positive or negative, discussed in section 3.3.3.

For the more general G-class (G>2) classification based on the MIL framework, there are G(G-1) pairwise MIL classifiers are trained where each classifier follows the standard pMILd training process. For the classification of the testing bag in this scenario, the testing bag is feed into every pairwise MIL classifier where each classifier predicts one of the two classes. A simple and straightforward way for the final class prediction is based on the majority voting of all predictions from each pairwise classifier. Let $\mathbf{B}$ be the total training bags for G classes, where $\mathbf{B}_g \in \mathbf{B}$ denotes the training bags for g-th class. Let $\tilde{\mathbf{B}}$ be a testing bag. The pseudo code of the training and testing phase of the proposed method in multi-class classification is shown in the Algorithm 3.4 and 3.5.

Algorithm 3.4. *Multi-class training phase of the proposed algorithm*

Input: $\mathbf{B}$, $\mathbf{Y}$

1: ***for*** $i = 1 : G - 1$ ***do***

2: **for** $j = i : G$ **do**
3: **pMILd training process for** i-**vs**-j **classifier**: *Estimate the distribution parameter* $\Theta_{i,j}$ *and threshold* $\tau_{i,j}$ *with* $\mathbf{B}_i$ *as positive bags and* $\mathbf{B}_j$ *as negative bags*
4: **pMILd training process for** j-**vs**-i **classifier**: *Estimate the distribution parameter* $\Theta_{j,i}$ *and threshold* $\tau_{j,i}$ *with* $\mathbf{B}_j$ *as positive bags and* $\mathbf{B}_i$ *as negative bags*
5: **end for**
6: **end for**

Output: $\{\Theta_{i,j}\}_{i \neq j}$, $\{\tau_{i,j}\}_{i \neq j}$

Algorithm 3.5. *Multi-class testing phase of the proposed algorithm*

Input: $\tilde{\mathbf{B}}$

1: **for** $i = 1 : G - 1$ **do**
2: **for** $j = i : G$ **do**
3: **pMILd testing process for** i-**vs**-j **classifier**: *Feed* $\tilde{\mathbf{B}}$ *into pairwise classifier* $\{\Theta_{i,j}, \tau_{i,j}\}$, *classify as* i *or* j, *and add the predicted class to* U
4: **pMILd testing process for** j-**vs**-i **classifier**: *Feed* $\tilde{\mathbf{B}}$ *into pairwise classifier* $\{\Theta_{j,i}, \tau_{j,i}\}$, *classify as* j *or* i, *and add the predicted as to* U
5: **end for**
6: **end for**
7: *Majority voting on* U: $\tilde{\mathbf{Y}} = mode(U)$

Output: $\tilde{\mathbf{Y}}$

3.5.1 pMILd with Confidence Aggregation

In the multi-class classification based on the MIL framework, the proposed confidence thresholding and majority voting based classification in the Algorithm 3.4 and 3.5 has been shown to achieve a satisfying performance in the next experimental results section. However, thresholding on each pairwise classifier makes a crisp classification and ignores how confident the classification is. A toy example is shown in Figure 3-2. $\mathbf{x}_1$ is a sample on the boundary region between two classes so thresholding actually imposes a 100% confidence that $\mathbf{x}_1$ is either class, while a more real confidence should be about half-half confidence to classified as either class. $\mathbf{x}_2$ is a sample that is far away from both classes, which can be considered as an outlier, while the current setting enforces it to be classified as either class. A confidence could alleviate the drawback of thresholding and majority voting. In this subsection, two alternative methods based on the confidence aggregations are proposed. The default setting for pMILd is based on the distribution as GMM and Mahalanobis distance metric for target probability.

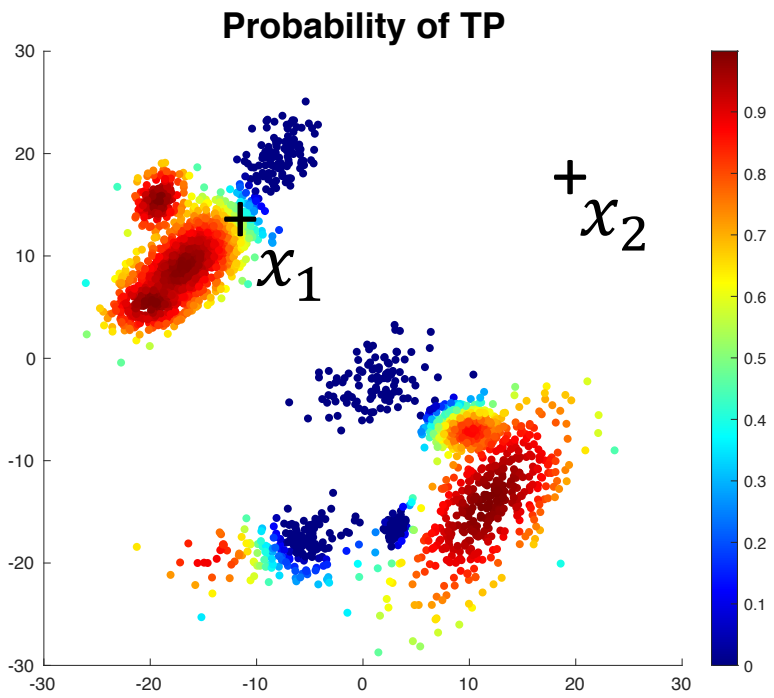


Figure 3-2. Motivation for using confidence aggregation

The first confidence aggregation is target probability (TP) based and the second confidence aggregation is likelihood based. Both approaches don't directly make a crisp prediction on each one-vs-one classifier (e.g. G1 is predicted in G1-vs-G2), but they allow a soft prediction. More specifically, the TP based method uses the bag level target probability value as a confidence value for the testing bag as target class shown in Equation 3–18 (e.g. $P\left(B_j|\Theta^{G1}\right) = 0.7$ in G1-vs-G2). Similarly, the likelihood based approach uses the likelihood value of the testing bag with respect to positive or negative class as either confidence value as the positive class or negative class shown in Equation 3–24 and Equation 3–25 (e.g. $P\left(B_j|\Theta^{G1}\right) = 0.6$ and $P\left(B_j|\Theta^{G2}\right) = 0.3$ in G1-vs-G2). The definition of likelihood with respect to positive and negative class of w-th one-vs-one classifier for j-th bag are defined in Equation 3–26 and 3–27. w_t and w_b denotes the positive class and negative class of w-th one-vs-one classifier.

$$P\left(\mathbf{x}_{ij}|\Theta^{w_t}\right) = e^{-\min_k\left[(\mathbf{x}_{ij}-\mu_k)^T G_k^{-1}(\mathbf{x}_{ij}-\mu_k)\right]} \quad \forall i, j, w \tag{3–24}$$

$$P\left(\mathbf{x}_{ij}|\Theta^{w_b}\right) = e^{-\min_q\left[(x_{ij}-\mu_q)^T G_q^{-1}(x_{ij}-\mu_q)\right]} \quad \forall i,j,w \tag{3–25}$$

$$P\left(B_j|\Theta^{w_t}\right) = \frac{1}{N_j}\sum_{i=1}^{N_j} P\left(\mathbf{x}_{ij}|\Theta^{w_t}\right) \quad \forall j,w \tag{3–26}$$

$$P\left(B_j|\Theta^{w_b}\right) = \frac{1}{N_j}\sum_{i=1}^{N_j} P\left(\mathbf{x}_{ij}|\Theta^{w_b}\right) \quad \forall j,w \tag{3–27}$$

$$P\left(B_j|\Theta^{g}\right) = \frac{1}{2*(G-1)}\left(\sum_{w_t=g} P\left(x_{ij}|\Theta^{w_t}\right) + \sum_{w_b=g} P\left(x_{ij}|\Theta^{w_b}\right)\right) \quad \forall j,g \tag{3–28}$$

The definition of target probability with respect to positive of w-th one-vs-one classifier for j-th bag are defined in Equation 3–29.

$$P\left(B_j|\Theta^{w_t}\right) = \frac{1}{N_j}\sum_{i=1}^{N_j} P\left(\mathbf{x}_{ij}|\Theta^{w_t}\right) \quad \forall j,w \tag{3–29}$$

$$P\left(B_j|\Theta^{g}\right) = \frac{1}{(G-1)}\left(\sum_{w_t=g} P\left(x_{ij}|\Theta^{w_t}\right)\right) \quad \forall j,g \tag{3–30}$$

where $P\left(\mathbf{x}_{ij}|\Theta^{w_t}\right)$ is calculated using Equation 3–18

The pseudocode for the likelihood-based confidence aggregation method is shown in the Algorithm. 3.6. The target probability based confidence aggregation, instead, uses target probability to calculate confidence for each pairwise classifier.

Algorithm 3.6. *Multi-class testing phase of the likelihood-based confidence aggregation algorithm*

Input: $\tilde{\mathbf{B}}$, $\{U_g\}_{g=1:G} = \emptyset$

1: ***for*** $i = 1 : G-1$ ***do***
2: ***for*** $j = i : G$ ***do***
3: ***pMILd testing process for*** i-***vs***-j ***classifier***: *Feed* $\tilde{\mathbf{B}}$ *into pairwise classifier* $\{\Theta_{i,j}, \tau_{i,j}\}$, *calculate confidence* u_i *as class i using Equation 3–26 and confidence* u_j *as class j using Equation 3–27 and update* $U_i = U_i \cup u_i$, $U_j = U_j \cup u_j$
4: ***pMILd testing process for*** j-***vs***-i ***classifier***: *Feed* $\tilde{\mathbf{B}}$ *into pairwise classifier* $\{\Theta_{j,i}, \tau_{j,i}\}$ *and calculate confidence as class j using Equation 3–26 and class i using Equation 3–27*
5: ***end for***

6: ***end for***
7: *Calculate average confidence per class using* $\bar{U}_c = \frac{1}{2\times(G-1)} \sum_{u_c \in U_c} u_c$
8: *Classification:* $\tilde{\mathbf{Y}} = \arg\max_c \bar{U}_c$
Output: $\tilde{\mathbf{Y}}$

For both approaches, instead of counting the votes for each class in majority voting, the aggregated confidence values are calculated by taking the average of all confidence values associated with each class. For example, the aggregation confidence for j-th bag with respect to g-th class is shown in Equation 3–28 for likelihood based method. The one class the has the maximum aggregated confidence value is predicted if a crisp prediction is needed. The major difference between the proposed two confidence aggregation methods is what confidence metric to use. TP based method uses discriminative TP values, emphasizing how more likely the bag is more similar to positive class rather than negative class, with carefully selected hyperparameters β_1 and β_2. Likelihood based method focuses more on the estimated distributions of positive class and negative classes.

3.5.2 pMILd with Confidence Aggregation and Confidence Calibration

As discussed in the previous subsection, the proposed confidence aggregation methods leverage the confidence values from each pairwise classifier to boost the classification performance. However, directly averaging the confidences per class has a potential problem, which is that the confidence value estimated from each pairwise classifier is not on the same scale. Therefore, two proposed confidence calibration methods are proposed to rescale the confidence to the same scale before averaging the confidences for final classification.

The basic idea of confidence calibration is based on comparing the difference between the original confidence value and the estimated threshold value of each pairwise classifier. Let the confidence value be u. Let the threshold value be τ. The proposed bilinear confidence calibration estimates the rescaled confidence u' are shown in Equation 3–31 and 3–32.

$$u' = 0.5 + 0.5 \times \frac{u - \tau}{1 - \tau}, \quad if \quad u > \tau \tag{3–31}$$

$$u' = 0.5 \times \frac{u}{\tau}, \quad if \quad u <= \tau \tag{3–32}$$

The proposed threshold-deduction confidence calibration estimates the rescaled confidence x' are shown in Equation 3–33 .

$$x' = x - \tau \tag{3–33}$$

Algorithm 3.7. *Multi-class testing phase of the TP-based confidence aggregation and calibration algorithm*

Input: $\tilde{\mathbf{B}}$, $\{U_g\}_{g=1:G} = \emptyset$
1: ***for*** $i = 1 : G - 1$ ***do***
2: ***for*** $j = i : G$ ***do***
3: ***pMILd testing process for*** i-***vs***-j ***classifier***: *Feed* $\tilde{\mathbf{B}}$ *into pairwise classifier* $\{\Theta_{i,j}, \tau_{i,j}\}$, *calculate confidence* u_i *as class i using Equation 3–29, calculate* u_i' *using Equation 3–32 or Equation 3–33 and update* $U_i = U_i \cup u_i'$.
4: ***pMILd testing process for*** j-***vs***-i ***classifier***: *Feed* $\tilde{\mathbf{B}}$ *into pairwise classifier* $\{\Theta_{j,i}, \tau_{j,i}\}$ *calculate confidence* u_j *as class j using Equation 3–29 calculate* u_j' *using Equation 3–32 or Equation 3–33 and update* $U_j = U_j \cup u_j'$.
5: ***end for***
6: ***end for***
7: *Calculate average confidence per class using* $\bar{U}_c = \frac{1}{2\times(G-1)} \sum_{u_c \in U_c} u_c$
8: *Classification:* $\tilde{\mathbf{Y}} = \arg\max_c \bar{U}_c$
Output: $\tilde{\mathbf{Y}}$

3.5.3 pMILd with KL Divergence based Dimensionality Reduction

For high dimensional data with a very limited number of training samples, it is usually not applicable to directly train a pMILd model in the original data space. Therefore, it is essential to reduce the dimensionality before the optimization step. Linear Discriminative Analysis (LDA), one of the most common dimensionality reduction methods, offers a linear mapping to reduce the dimensionality in a supervised manner. In this thesis, an unsupervised feature selection/dimensionality reduction approach is proposed based on the Kullback-Leibler (KL) divergence between the normalized histogram of target and background classes of each pairwise classifier.

The proposed KL based dimensionality reduction contains three main steps, which are histogram calculation, KL divergence calculation and peak/feature selection. The basic idea is that the feature statistics of which target and background classes are mostly different (measured by KL divergence) contain the most discriminative information to separate two classes. In addition, peak selection prevents to select the neighboring features that are highly redundant.

Let $\mathbf{h}_i$ and $\mathbf{h}_j$ be the normalized histogram for each dimensionality of $\mathbf{B}_\text{i}$ and $\mathbf{B}_\text{j}$, respectively. Let $\mathbf{div}_{ij}$ be the symmetric KL divergence value between each dimension of $\mathbf{h}_i$ and $\mathbf{h}_j$.Let hyperparameters min_{gap}, max_{feat} be the minimum number of features between two peaks and the maximum number of features to be selected, respectively. Let $\mathbf{f}_{ij}$ be the selected features for the i-vs-j classifier.

Algorithm 3.8. *KL divergence based dimensionality reduction for the i-vs-j classifier*

Input: $\mathbf{B}_\text{i}$, $\mathbf{B}_\text{j}$, min_{gap}, max_{feat}

1: ***Normalized histogram****: Compute* $\mathbf{h}_i$ *and* $\mathbf{h}_j$ *for each dimensionality*
2: ***KL divergence****: Compute* $\mathbf{div}_{ij}$ *between each dimension of* $\mathbf{h}_i$ *and* $\mathbf{h}_j$
3: ***Peak search****: Find all the peak values of the plot of* $\mathbf{div}_{ij}$*, when* min_{gap} *are met for every two peaks. Sort all peaks in descending order, leading to a preliminary feature set* $\mathbf{f}_{pkts}$
4: ***Feature selection****: Final feature set* $\mathbf{f}_{ij}$ *is the first* max_{feat} *number of features in* $\mathbf{f}_{pkts}$ *(or all features in* $\mathbf{f}_{pkts}$ *if* $|\mathbf{f}_{pkts}| < max_{feat}$*).*

Output: $\mathbf{f}_{ij}$

CHAPTER 4
EXPERIMENTAL RESULTS

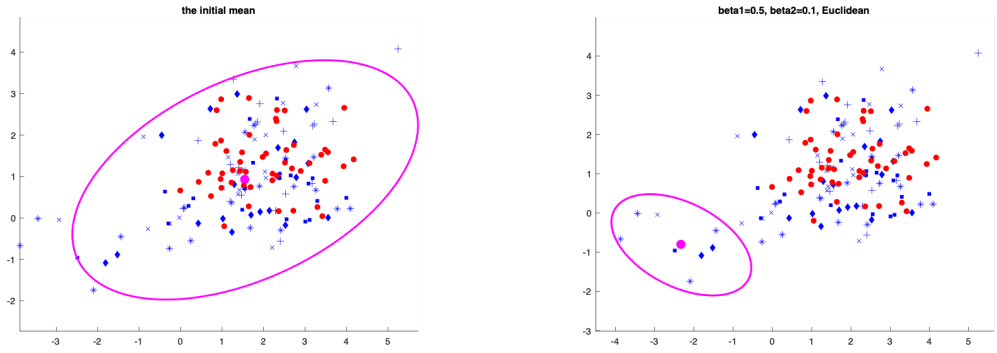
The performance of the proposed probabilistic multiple instance learning with distribution method is evaluated on both synthetic and real datasets. The goal of a synthetic dataset is to provide an intuitive understanding about the mechanism of the proposed algorithm in two-dimensional, three-dimensional or even higher dimensional space, corresponding to the 2D signal (e.g. grayscale image), 3D signal (e.g. RGB image) and high dimensional signal (e.g. hyperspectral image), respectively. In addition, the performance of the proposed algorithm can be best evaluated using synthetic datasets since the labels of real datasets always contain a certain amount of errors, resulting in a non-perfect evaluation for the proposed method. However, real datasets are still used since they are more realistic, showing how the algorithm performs towards real life scenarios.

4.1 Experiments on Synthetic Dataset

The synthetic dataset includes pure and mixed 2D, 3D and synthetic hyperspectral images. A pure sample represents a sample that completely consists of a candidate from one class, while the mixed sample denotes a sample that is a convex combination of candidate samples from more than one class. In the synthetic data experiments, pure data is used.

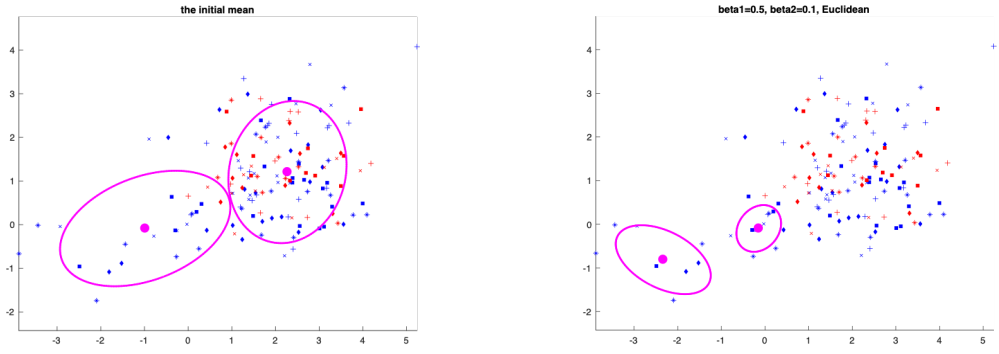
4.1.1 Standard pMILd on 2D Data

There are two bags (C_1 and C_2) generated for the 2D pure dataset, representing the target and non-target bags, respectively. The samples for both classes are drawn from two 2D Gaussian distributions ($\mathscr{N}_1(\cdot|\mu_1, \Sigma_1)$, $\mathscr{N}_2(\cdot|\mu_2, \Sigma_2)$) with different mean values ($\mu_1 = [1, 1]$, $\mu_2 = [2, 1]$)and covariance matrices ($\Sigma_1 = [5, 2.5; 2.5, 2.5]$, $\Sigma_2 = [1, 0; 0, 1]$). There are 100 training samples (50 samples drawn from $\mathscr{N}_1$ and 50 samples drawn from $\mathscr{N}_2$) that are generated for target bag and 50 training samples (50 samples drawn from $\mathscr{N}_2$) generated for non-target bag, respectively. Note that the target bag, C_1, contains both true target and true non-target instances. In addition, there are some Gaussian noise added to the dataset such that the underlying target and non-target distributions could be multi-modal. Two proposed



A B

Figure 4-1. 2D synthetic dataset 1 with the target as a single Gaussian model. Blue: positive instances. Red: negative instances. Pink: estimated Gaussian mean and covariance. A) Initialization; B) Estimation.



A B

Figure 4-2. 2D synthetic dataset 1 with the target as a GMM model. Blue: positive instances. Red: negative instances. Pink: estimated Gaussian means and covariances. A) Initialization; B) Estimation.

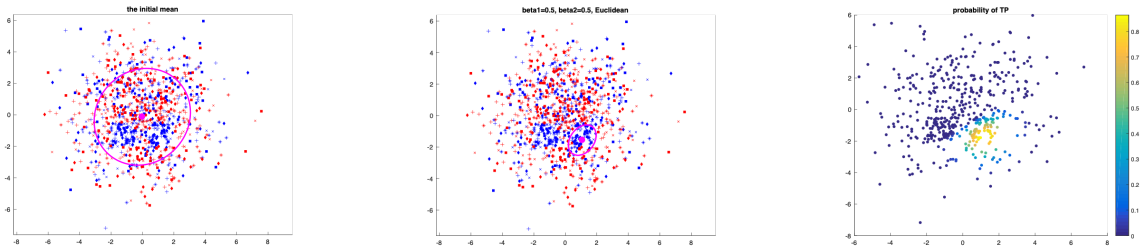
models, the target as a single Gaussian distribution and the target as a GMM, are both evaluated on the 2D pure dataset.

For the proposed target as a single Gaussian model, the initialization of the mean and covariance of the target distribution is simply the sample mean and sample covariance of the target bag. The initial target distribution is the shown in Figure 4-1 A where the initial target mean is the purple dot and the initial target covariance is the purple ellipse. The estimated target distribution using the proposed model is shown in Figure 4-1 B. The estimated target

distribution characterizes the sample distribution in the target bag that is different from the non-target bag.

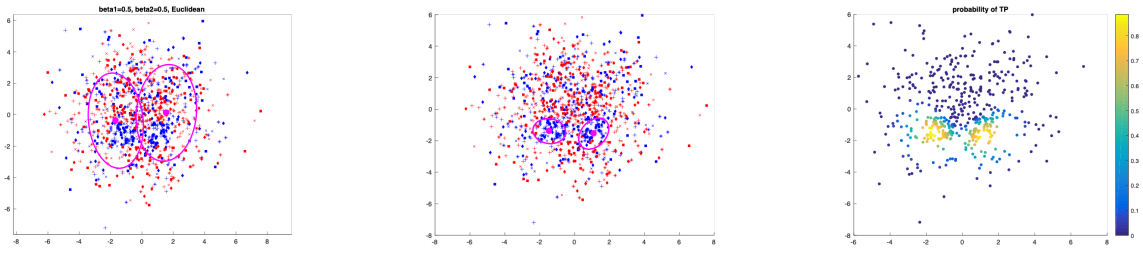
The estimated target distribution is simply a Gaussian distribution, which is very restricted which assumes the underlying target distribution follows Gaussian distribution. However, the underlying target distribution could be multi-modal such as the synthetic 2D pure dataset in this section. Therefore, the proposed target as a GMM model is also evaluated on the 2D pure dataset.

For the proposed target as a GMM model, the mean and covariance of each Gaussian component of the target distribution are initialized with K-means clustering method. Specifically, the mean of each Gaussian component is initialized with each cluster mean estimated by K-means and the covariance of each Gaussian component is initialized with each sample covariance estimated by K-means. The initial and estimated target distributions are shown in Figure 4-2 A and B, respectively. As it is shown in Figure 4-2 B, the estimated target distribution is capable of learning the multi-modal structure of the feature space where the target bags are different from non-target bags, which is the feature space of underlying target distribution. Note that even though one of the initial target component distribution is located in the feature space of non-target samples shown in Figure 4-2 A, this target component distribution is gradually pushed away from the space of non-target samples and finally converges to one of the underlying target component distribution.



A B C

Figure 4-3. 2D synthetic dataset 2 with the target as a single Gaussian model. A) Initialization; B) Estimation; C) Estimated target probability.



A B C

Figure 4-4. 2D synthetic dataset 2 with the target as a GMM model. A) Initialization; B) Estimation; C) Estimated target probability.

To further evaluate the performance of the proposed method, another 2D pure dataset is generated. For the second synthetic dataset, the target distribution is enclosed by the non-target distribution, different from the first synthetic data, where the target and non-target distributions are linearly separable.

There are five target bags and five non-target bags generated for the second 2D pure dataset. The samples for the target class are drawn from two 2D Gaussian distributions to simulate a multi-model underlying target distribution($\mathcal{N}_{11}\left(\cdot|\mu_{11},\Sigma_{11}\right)$, $\mathcal{N}_{12}\left(\cdot|\mu_{12},\Sigma_{12}\right)$) with different mean values ($\mu_1 = [1,-1]$, $\mu_2 = [-1,-1]$)and covariance matrices ($\Sigma_1 = [0.5, 0; 0, 0.5]$, $\Sigma_2 = [0.5, 0; 0, 0.5]$). The samples for the non-target class are drawn from a 2D Gaussian distribution ($\mathcal{N}_2\left(\cdot|\mu_2 = [0,0], \Sigma_2 = [5, 0; 0, 5]\right)$). There are 500 training samples (150 samples drawn from $\mathcal{N}_{11}, \mathcal{N}_{12}$ and 350 samples drawn from $\mathcal{N}_2$) that are generated for target bag and 1000 training samples (1000 samples drawn from $\mathcal{N}_2$) generated for non-target bag, respectively. Two proposed models, the target as a single Gaussian distribution and the target as a GMM, are both evaluated on the 2D pure non-linear dataset.

Experimental results in Figure 4-3 and Figure 4-4 show that the proposed method is capable of learning (a least one part of) the data structure of underlying true target samples. Moreover, modeling the underlying target distribution as a GMM instead of a single Gaussian can even more factually reflect the real multi-modal target distribution.

4.1.2 pMILd with Confidence Aggregation on 2D Data

The performance of confidence aggregation for both proposed methods are first evaluated on the 2-D synthetic dataset. The number of classes C is set to be 3, 4, 5 and 6 classes. The difficulty of classification increases as the number of classes increases. The positive and negative bags generation process for every two classes is in shown in Figure 4-5.

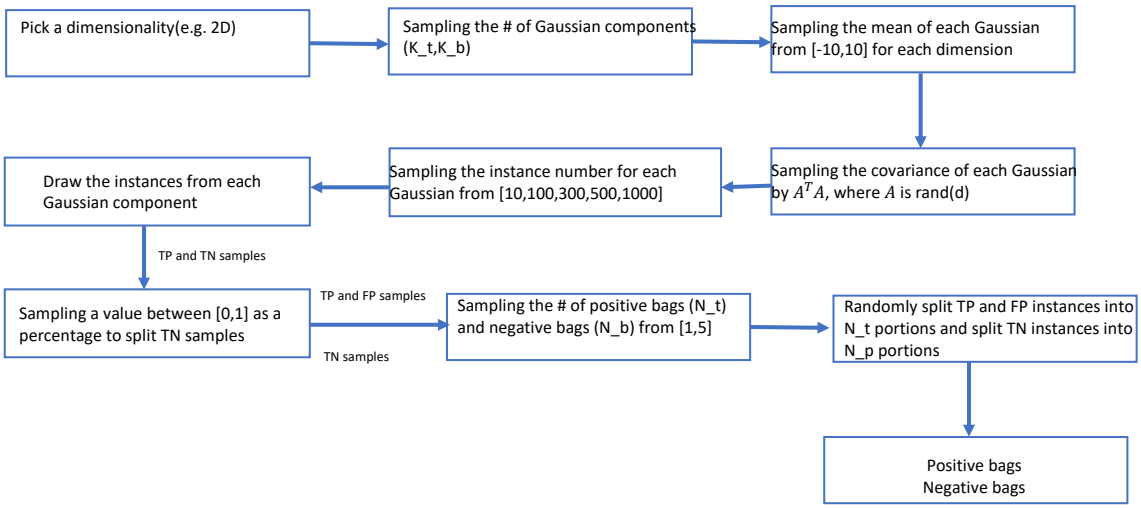


Figure 4-5. Synthetic MIL bags generations (standard two-classes)

For each setting (different number of classes), the experiments are repeated for 100 times and the average performance is investigated. The micro F1 and macro F1 scores are calculated for the two proposed methods (TP based and likelihood based) and the baseline method (thresholding and majority voting). Figure 4-6 shows the number of times each method has the highest F1 score among all three methods (can both or all be highest). Figure 4-7 presents the mean and standard deviation of two F1 score metrics of each method.

The experimental results show that as the number of classes increases, the TP based method has more times to achieve the highest F1 score. Both proposed method is shown to outperform the baseline method on the synthetic dataset. For the most difficult cases (C=6), the average macro F1 of TP and likelihood based method is 1.4% and 0.4% higher than the baseline method and the average micro F1 of TP and likelihood based method is 1.4% and 0.5% higher than the baseline method.

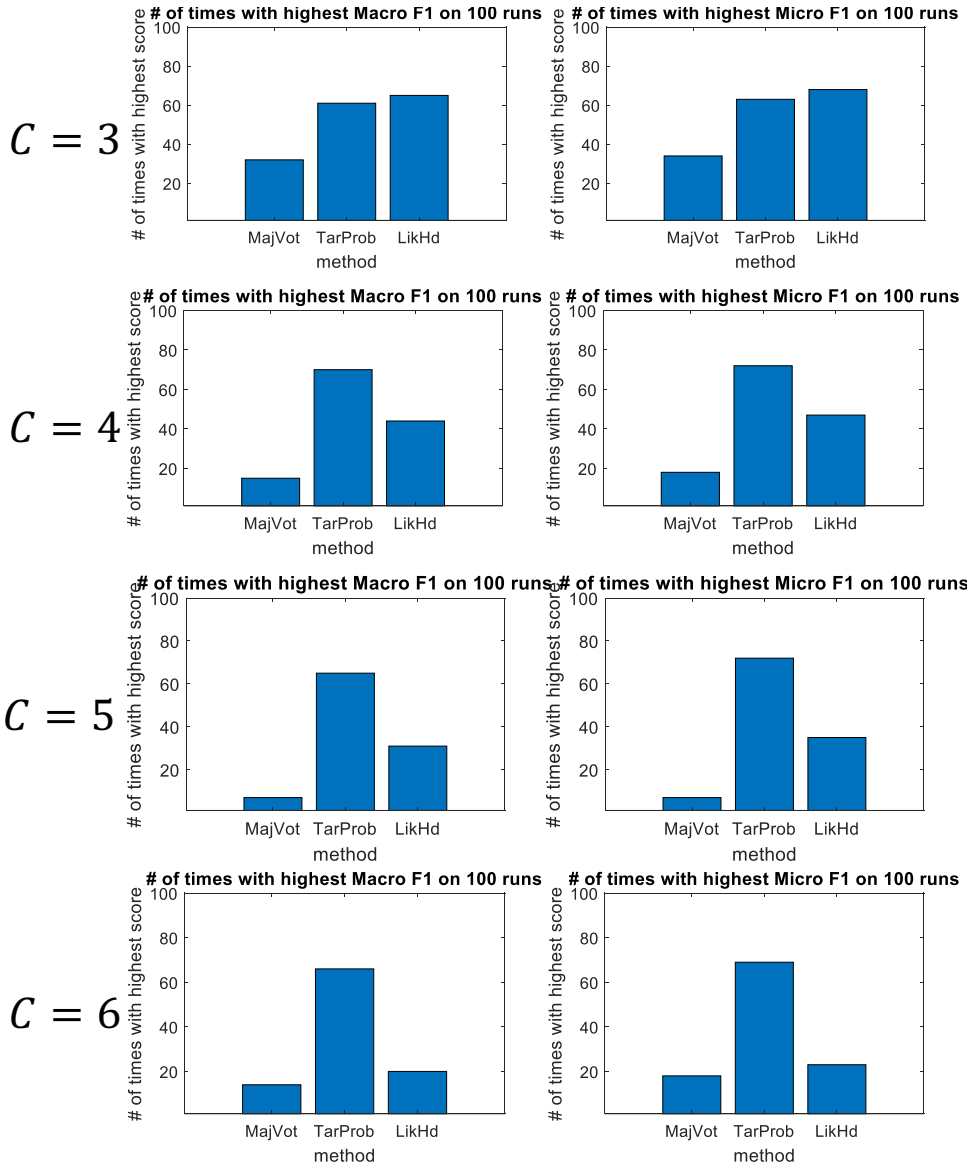


Figure 4-6. The number of times with tied highest F1 score over 100 runs for synthetic 2D multi-class classification with $C = 3, 4, 5, 6$

4.1.3 pMILd with Confidence Aggregation and Confidence Calibration on 2D Data

The experiments using two proposed confidence calibration methods are based on a synthetic 2D dataset with 6 classes. The experiments are repeated for 200 times. The results without using calibration are shown in Figure 4-8. The results using the proposed bilinear and threshold-deduction calibration are shown in Figure 4-9 and Figure 4-10. The experimental results show that the average micro F1 score is improved from 0.944 to 0.967

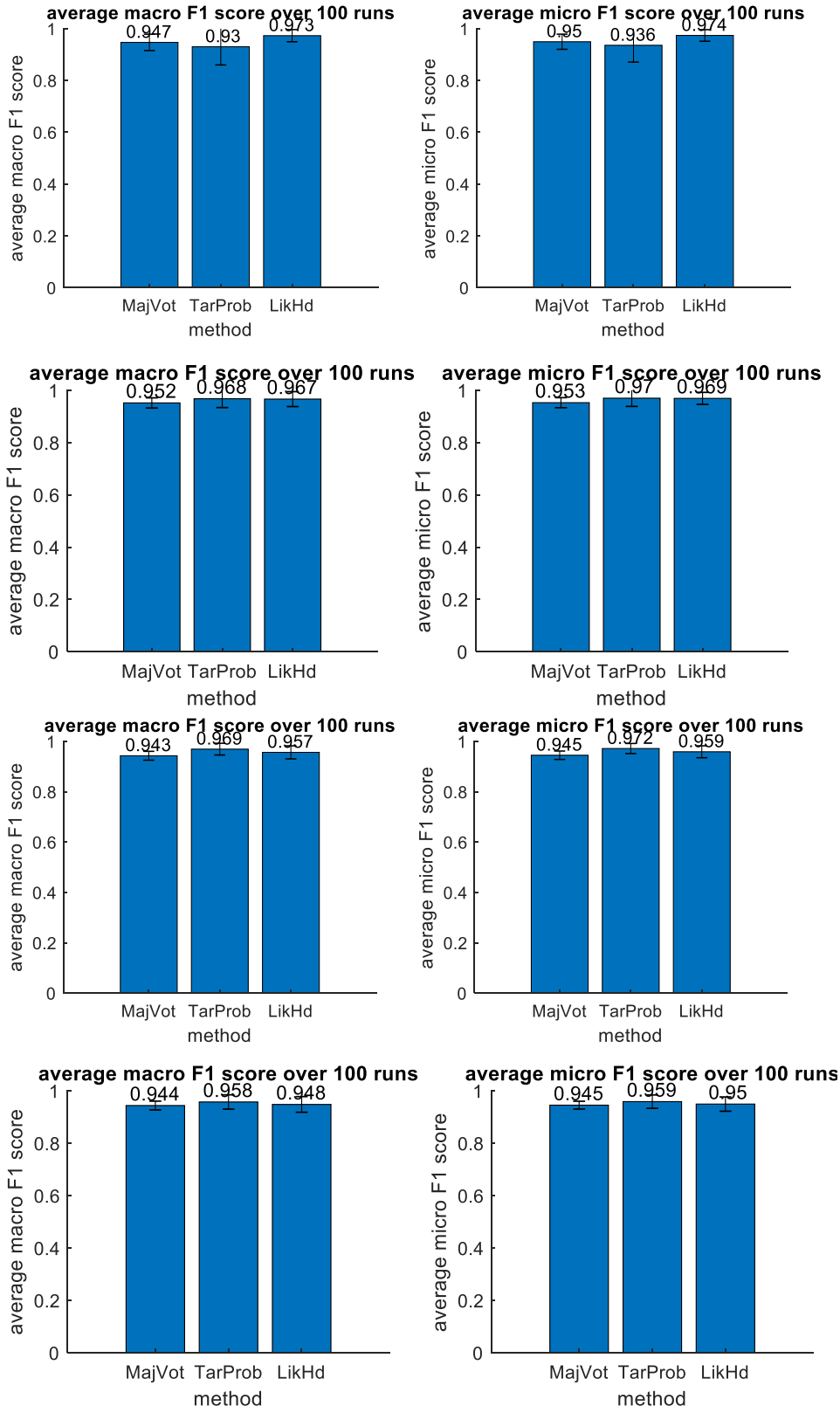
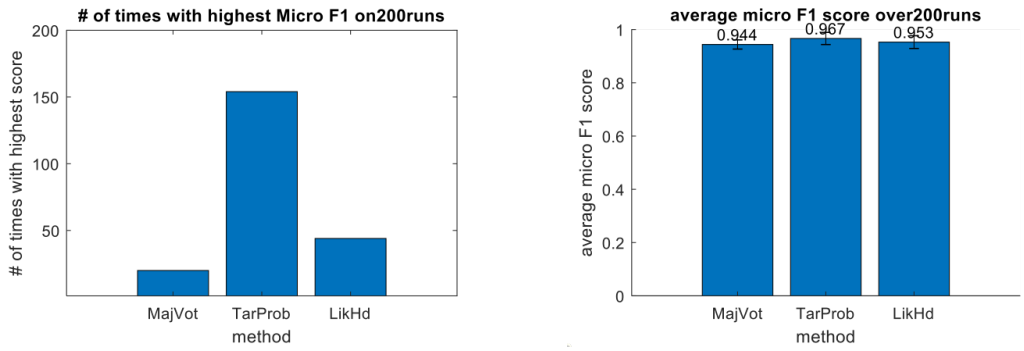
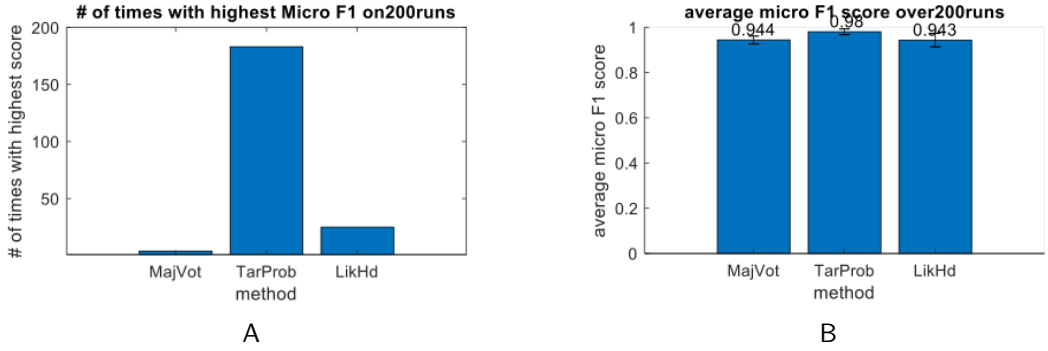


Figure 4-7. The average F1 score over 100 runs for synthetic 2D multi-class classification with $C = 3, 4, 5, 6$



A B

Figure 4-8. No confidence calibration. A) The number of times with tied highest micro F1 score; B) Average micro F1 score.



A B

Figure 4-9. Bilinear calibration. A) The number of times with tied highest micro F1 score; B) Average micro F1 score.

using the proposed TP-based confidence aggregation method and from 0.967 to 0.980 using the proposed bilinear calibration method.

4.2 Experiments on Real Dataset

4.2.1 Tree Species Classification on UCSB Data

The proposed pMILd is first applied for tree species classification of UCSB hyperspectral image dataset. The imagery was collected at Santa Barbara with the AVIRIS sensor. The collected hyperspectral images consist of 224 bands of radiance between 360 and 2500 nm and a spatial resolution of 18 meters. There are in total 10 flight lines covering a diverse landscape where the most of the area is Los Padres National Forest (LPNF). There are totally 27 classes including various plant species and land cover classes that were manually selected

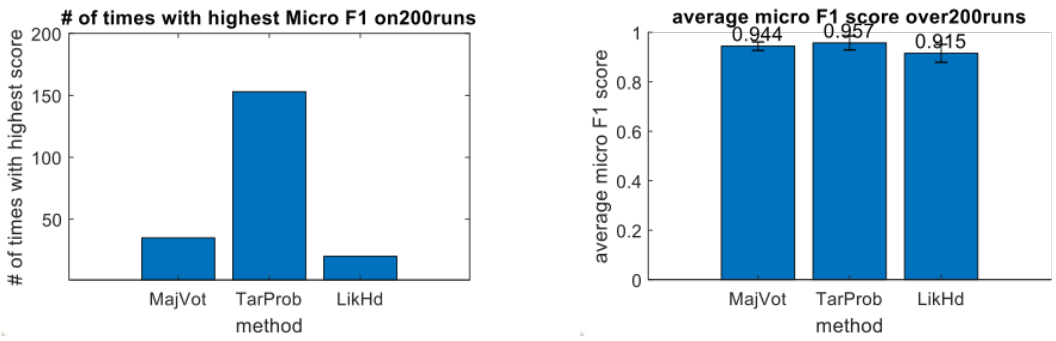


Figure 4-10. Threshold-deduction calibration. A) The number of times with tied highest micro F1 score; B) Average micro F1 score.

from the 10 flight lines by ecologists. To be more specific, each class consists of a series of polygons outlined by ecologists, where each polygon has over 75% labeled species (Meerdink et al., 2019). Since all polygons of all classes are mostly not pure, the assigned class labels for polygons are mostly imprecise. Thus, the proposed multiple instance learning based method can be used because each polygon can be viewed as a MIL bag and each polygon label can be regarded as a bag-level label.

Tree species classification with hyperspectral data is a challenging task. Firstly, unlike other classification problem such as classification among buildings, vegetations and sidewalks in the urban area, the feature vectors (spectral signatures) of different tree species are usually very similar to each other. Secondly, the within-class variability for each tree species are different and can be complex, such as multi-modal. For instance, a single tree species class contains polygons collected at different flight lines could have different signatures, due to growing conditions, data collection conditions or even they are different sub-species. However, the proposed method is capable of capturing the complex within-class variability by estimating a discriminative GMM to model the underlying complex distribution of each class.

4.2.1.1 Classification with LDA as dimensionality reduction

Since the UCSB tree species classification task is a 27-class classification problem. The proposed multi-class pMILd framework in section 3.5 is used. The experimental settings of

the pMILd algorithm for UCSB dataset are as follows. The distributions are assumed to be GMM for both positive and negative class in each pairwise classifier. The Mahalanobis distance metric is selected for computing the target probability. The original 224-D data is firstly reduced to 174-D by removing the water bands and further reduced to 26-D followed by a LDA dimensionality reduction method to as suggested by Meerdink et al. (2019). The initial numbers of Gaussian components for positive and negative class of each pairwise classifier are $K = 3$ and $Q = 3$, respectively. A pruning hyperparameter ζ is set to be $\zeta = 0.05$ to prevent the potential singularity issue of GMM. If the mixing proportion of any target Gaussian component, α_k is less than ζ before convergence, this Gaussian component is pruned and the related pairwise classifier is retrained with $K - 1$ number of target Gaussian components. For the weights β_1 and β_2 in the target probability (Equation 3–18) are selected in the initialization step (before EM iterative updation) such that the average value of $e^{-\beta_1 \min_k \left[(x_{ij}-\mu_k)^T C_k^{-1}(x_{ij}-\mu_k)\right]}$ is 0.85 and the average value of $e^{-\beta_2 \min_q \left[(x_{ij}-\mu_q)^T C_q^{-1}(x_{ij}-\mu_q)\right]}$ is 0.1 across all instances of positive bags. The way of selecting β_1 and β_2 leads to the initial average target probability is around 0.75, for the polygons of the dataset contain at least 75% of labeled class as discussed before.

The experiments are designed in a fashion of cross validation. There are in total 10 repetitive experiments, where 80% of polygons of each class are randomly selected as the training data and the rest 20% are as the testing data.

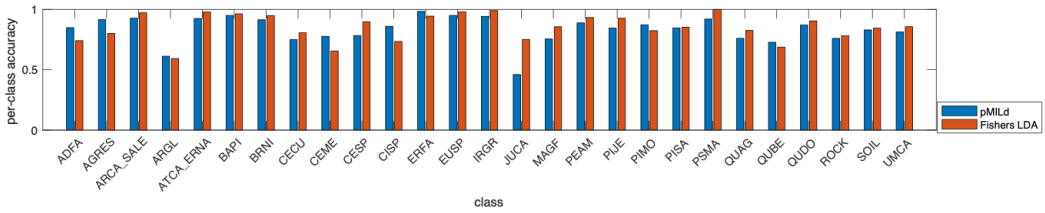


Figure 4-11. Comparison between pMILd and LDA-based method on each class

The overall accuracy (OA) is 84% across all 27 species using the proposed pMILd method, maintains the the best accuracy estimated by a LDA based method (Meerdink et al., 2019). The results of per-class accuracy are shown in the Figure 4-11. Although the proposed method

maintains the best overall accuracy performance on this dataset in the literature, the per-class accuracy shows that JUCA class has a much lower accuracy than the LDA method. JUCA is also the class with the least number of samples (178 samples) in the data. Because the proposed distribution based method has relatively more parameters to estimate (such as multiple means and covariance matrices for each class) so that it is not very suitable for dataset with very small training size.

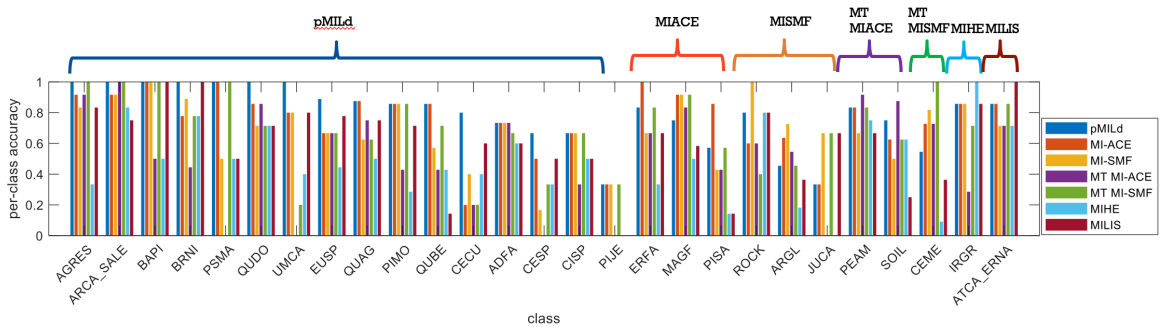


Figure 4-12. Comparison between pMILd and other MIL method on each class

The proposed method is also compared with some of the state-of-the-art MIL methods in the literature. The per-class accuracy results are shown in the Figure 4-12. The pMILd method demonstrates a top winner in regarding to have the most number of best per-class accuracy (16 classes among totally 27 classes).

4.2.1.2 Classification with KL divergence as dimensionality reduction

The default dimensionality reduction method is LDA dimensionality reduction method. In this subsection, the novel proposed KL divergence based method is also used on the UCSB dataset and compared with previous results.

The idea of the proposed KL divergence based feature selection method is to pick the most salient subset of features, for each pairwise classifier. For instance, for the classifier between ADFA-vs-ARGES, the first step is to calculate the normalized histogram for both classes on each band. The normalized histograms of 36-th band for both classes are shown in Figure 4-13, which is one the most salient feature, since two classes have quite different histograms (distributions) on 36-th band.

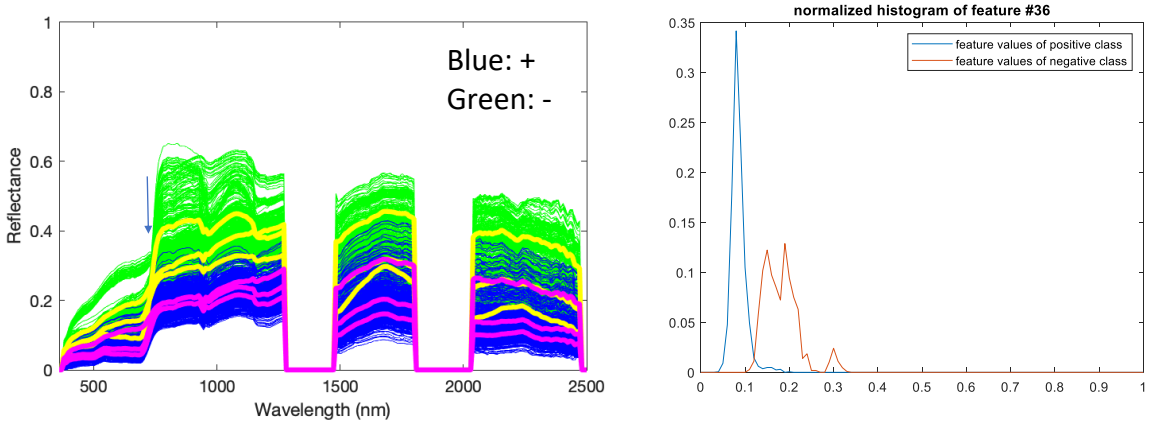


Figure 4-13. Left figure: the spectra of ADFA (blue) and ARGES (green). Right figure: normalized histogram on feature 36

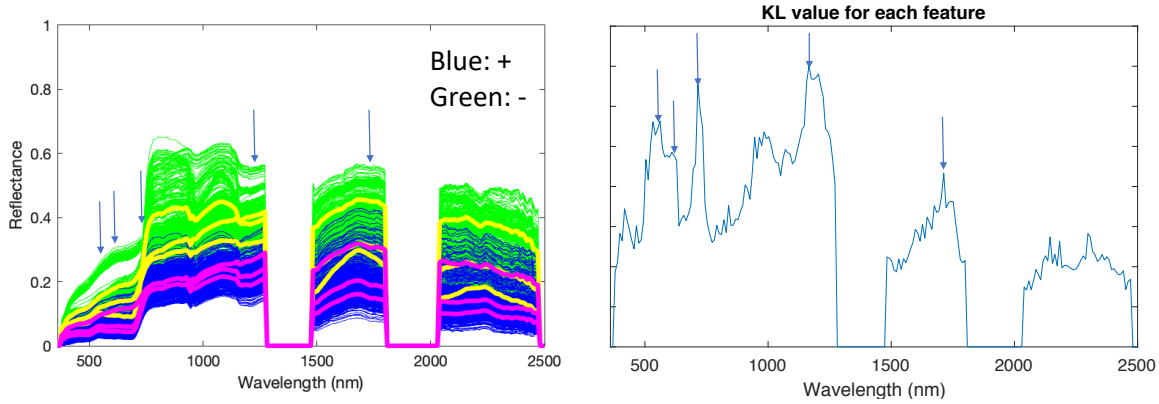


Figure 4-14. Left figure: the spectra of ADFA (blue) and ARGES (green). Right figure: KL divergence values on each band

KL divergence values are then computed to quantify how salient each feature is, as shown in Figure 4-14. These local maximum peaks are selected using the proposed peak selection method in section 3.5.3. The Figure 4-14 shows the final selected features with the parameter min_{gap} is 10 and max_{peaks} is 5, which select one peak for each local maxima in the KL divergence space.

The similar cross validation experiments but with the proposed feature selection method shows a slightly 2.7% lower than results using LDA as dimensionality reduction. However, there are many improvements to be investigated such as developing a better way to select min_{gap} and max_{peaks} instead of using a hand-picked fixed value for all pairwise classifiers. Even within

a single pairwise classifier, min_{gap} can vary in the KL divergence space. min_{gap} is actually used to find the major humps in the space of KL divergence and prevent selecting redundant neighboring features. A possible future work is to smooth the curve of KL plot to remove neighboring redundant peaks and then select large peaks in the plot. As discussed before, training a pMILd in a high dimensional space with very limited number of training data could lead to numerical issues such as covariance singularity. Therefore, a possible solution to find better max_{peaks} is to select max_{peaks} proportional to the number of samples in the training set.

4.2.1.3 Class variability interpretation

Figure 4-15. Satellite images of two QUBE polygons/bags

The proposed pMILd method can not only maintain the highest classification performance as the state-of-the-art method as shown in the previous subsection, but offer an interpretation about how data of each class is distributed on the level of samples. This is achieved by estimating a GMM for each class in each pairwise classifier and a target probability value for each sample. Therefore, the confidence of each sample with respect to each Gaussian component can also be easily calculated.

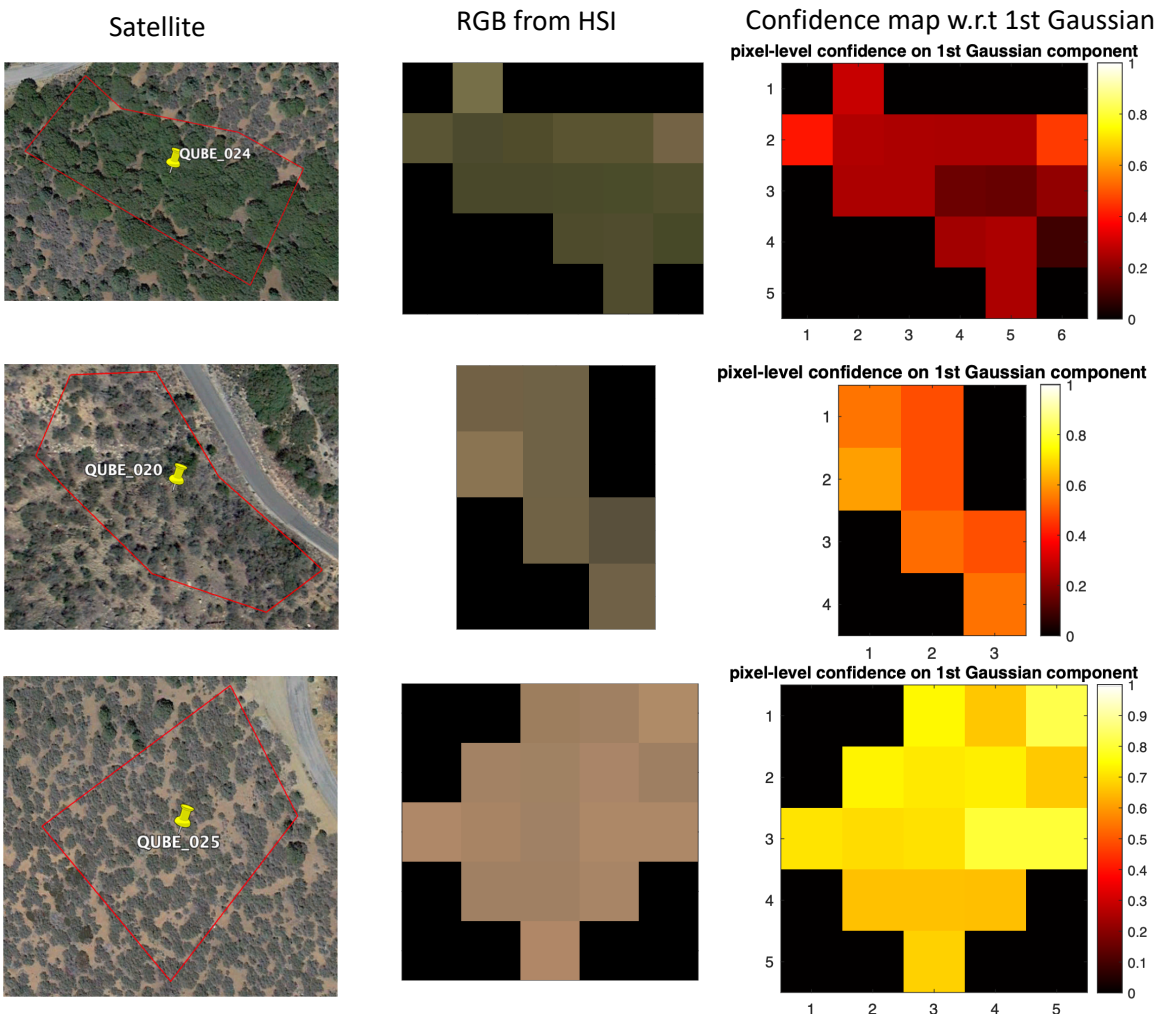
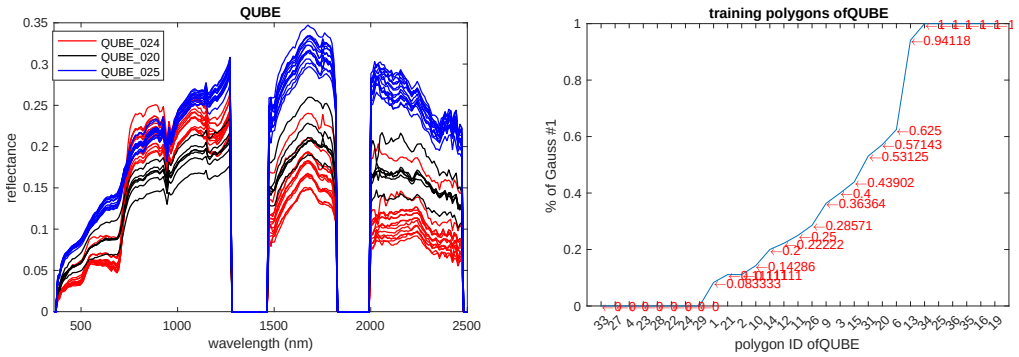


Figure 4-16. From left to right: satellite images, RGB images extracted from HSI and pixel-wise confidence map with respect to first estimated target Gaussian component

For the QUBE class in UCSB dataset as an example, the pMILd method learns a GMM for QUBE class where the number of Gaussian components is two. Without loss of generality, the GMM of QUBE estimated from the pairwise classifier of QUBE-vs-ROCK is used as an example. Two QUBE polygons/bags are shown in Figure 4-15, where two trees from the same class have significant visual different (including color). After feeding in all training data into the proposed model, the membership values of all QUBE samples with respect to each QUBE Gaussian component are computed. The membership values are then normalized to ensure sum-to-one constraint over all Gaussian components. The pixel level membership/confidence

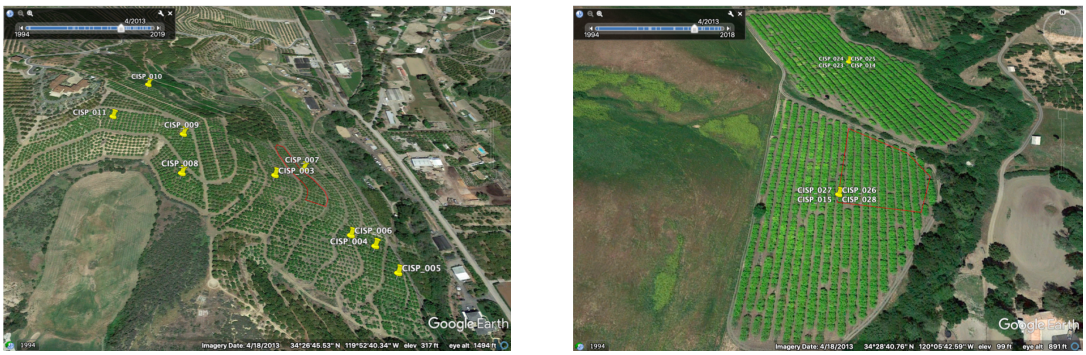


A B

Figure 4-17. Spectra and prediction of three QUBE polygons. A) spectra of three example QUBE polygons; B) membership to the first Gaussian component of QUBE class.

values with respect to the first QUBE Gaussian component are shown in Figure 4-17. These three selected QUBE polygons denote three different cases of pixel level confidence with respect to the first Gaussian component. For $QUBE_{024}$, all confidence values are less 0.5, showing a low membership with respect to the first estimated Gaussian component. In other words, this polygon has a high membership with respect to the second Gaussian. For $QUBE_{025}$, all confidence values are larger than 0.5, showing a high membership with respect to the first estimated Gaussian component. For $QUBE_{020}$, about half of samples are over 0.5 and the rest are below 0.5, showing memberships to both Gaussian components.

By visualizing both satellite images and the RGB images pulled out from HSI data, $QUBE_{024}$ polygon shows a relatively green color, $QUBE_{025}$ polygon shows a relatively brown color and $QUBE_{020}$ polygon shows an intermediate color between green and brown. Therefore, the GMM of QUBE can be inferred that two different groups of QUBE trees are learned by the proposed model, where the multi-model within-class variability could be captured by each Gaussian component. In this case, the first Gaussian component captures some of the more brown QUBE trees while the second Gaussian component captures the other more green QUBE trees. Figure 4-17 further shows the spectra of three example polygons and the bag-level membership by setting 0.5 as the sample-level threshold value. The spectra of $QUBE_{024}$ show a higher hump around the green wavelength range compared with $QUBE_{025}$, and the spectra



A B

Figure 4-18. CISP polygons. A) CISP polygons in one region; B) CISP polygons in another region

of $QUBE_{020}$ are mostly in between the other two polygons, which further demonstrating that pMILd learns two color variability within QUBE class. The sample-level membership with respect to each Gaussian component also offers information about the degree of each pixel is similar to the corresponding variability. Overall, pMILd visualizes the multi-modal within-class variability, which could further be used by domain scientist (such as biologist) to understand more deeply about the reason causing the variability and feed back the domain knowledge back to data scientist for future data collection and algorithm development.

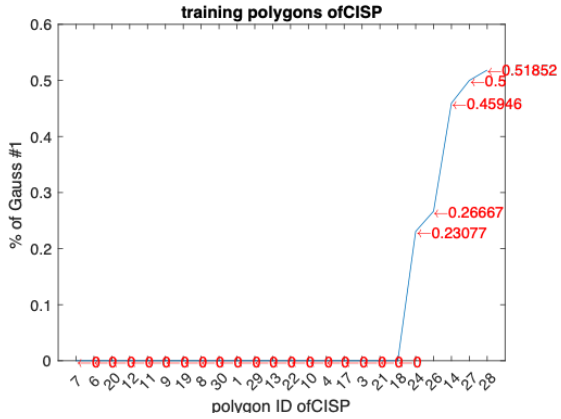


Figure 4-19. Membership to the first Gaussian component of CISP class

Another example is CISP class. Two CISP plantation areas are found to have high memberships on one of the two estimated Gaussian components of CISP class, respectively. Overall, there are potentially many possible reasons causing within-class variability, especially

for UCSB dataset where plants are collected at different geological areas/growing conditions and on different sunlight/sun angle. The proposed method is shown to be able to learn the underlying variability and visualize it on sample level by using membership with respect to each estimated Gaussian component. The visualization can further be used for interpretation with domain knowledge and assisting understating of reasons leading to the variability.

4.2.2 Tree Species Classification on NEON Data

The proposed pMILd is then applied for tree species classification of NEON hyperspectral image dataset from NEON 2020 competition. The imagery was collected at three NEON sites in the eastern United States, including Ordway-Swisher Biological Station, Florida (OSBS), Mountain Lake Biological Station, Virginia (MLBS) and Talladega National Forest, Alabama (TALL). The collected hyperspectral images consist of 426 bands of radiance between 380 and 2510 nm and a spatial resolution of 1 meter. There are in total 33 classes in the training set including various tree species that were manually selected from MLBS and OSBS sites by ecologists. To be more specific, each class consists of a series of bounding boxes generated by a tree crown delineation algorithm, where each box contains a single labeled tree species. The testing set was generated by trees from all three sites. For the testing set, there are 19 classes that are the same classes in the training set as well as some new classes that are not in the testing set, all called the 'others' class.

As shown in Figure 4-20, these tree labels assigned to bounding boxes are imprecise, in terms of pixel-level labels. All most all boxes contain a certain level of pixels of other neighboring species. For some of these boxes, these noisy pixels are quite dominant. The major reasons are as follows.

1) Bounding boxes are rectangular while trees are irregular so it is inevitable there are some noisy pixels. 2) Trees at these NEON sites usually grow densely, which leads to a challenging task to extract regions containing single trees only, even with the-state-of-the-art superpixel methods. 3) There could be trees grow at an oblique or horizontal angle or

Figure 4-20. A region of MLBS containing multiple bounding boxes

underneath another tree, leading to a mismatch between the labels collected at tree truck and spectral data of tree crown collected from the plane.

Similar to classification of UCSB data, the proposed multiple instance learning based method can be used because each box can be viewed as a MIL bag and each box label can be regarded as a bag-level label. Note that these imprecise single-label box labels at NEON sites are not implausible to used for evaluating the performance such as classification accuracy, since the true underlying tree species in each box could be multi-label or even different from the assigned single label.

This dataset consists of some tree species classes that are rare classes. For these rare classes, only a few boxes/trees present and each box contains a few pixels (usually less than ten pixels). Therefore, these rare classes with only a few pixels are challenging to be used and classified in current model. For now, a subset dataset is generated such that only classes with larger or equal than 30 training samples are selected, leading to 27 classes to be classified, which are shown in the Table. 4-1. Note that the 'others' class in the testing data is also

classified to one of the 27 classes in the training set. The future work is to propose an outlier detection method to filter out these 'other' classes that are not observed in the training set.

Table 4-1. NEON classes used for training and their corresponding box/tree numbers

Class	Box numbers
ACPE	8
ACRU	138
ACSA3	1
AMLA	43
BETUL	6
CAGL8	3
FAGR	6
LITU	17
MAGNO	15
NYBI	2
NYSY	45
OXYDE	11
PIEL	6
PINUS	7
PIPA2	295
PITA	4
PRSE2	7
QUAL	103
QUCO2	52
QUGE2	18
QUHE2	5
QULA2	71
QUMO4	12
QUNI	3
QURU	166
ROPS	2
TSCA	4

4.2.2.1 Classification with LDA as dimensionality reduction

Since the NEON tree species classification task is a 33-class classification problem. The proposed multi-class pMILd framework in section 3.5 is used. The experimental settings of the pMILd algorithm for NEON dataset are as follows. The distributions are assumed to be GMM for both positive and negative class in each pairwise classifier. The Mahalanobis distance metric is selected for computing the target probability. The original 426-D data is firstly reduced to 369-D by removing the water bands and further reduced to 4-D followed by

a LDA dimensionality reduction method to as suggested in Meerdink et al. (2019). The initial numbers of Gaussian components for positive and negative class of each pairwise classifier are $K = 2$ and $Q = 2$, respectively. The reason to select 4-D and 2 Gaussian components is that there are some classes with as low as 30 training samples. Therefore, too many model parameters could lead to overfitting or even numerical instability. A pruning hyperparameter ζ is set to be $\zeta = 0.05$ to prevent the potential singularity issue of GMM. For the weights β_1 and β_2 in the target probability (Equation 3–18) are selected in the initialization step (before EM iterative updation) such that the average value of $e^{-\beta_1 \min_k \left[(x_{ij}-\mu_k)^T C_k^{-1}(x_{ij}-\mu_k)\right]}$ is 0.8 and the average value of $e^{-\beta_2 \min_q \left[(x_{ij}-\mu_q)^T C_q^{-1}(x_{ij}-\mu_q)\right]}$ is 0.1 across all instances of positive bags. The way of selecting β_1 and β_2 leads to the initial average target probability is around 0.7.

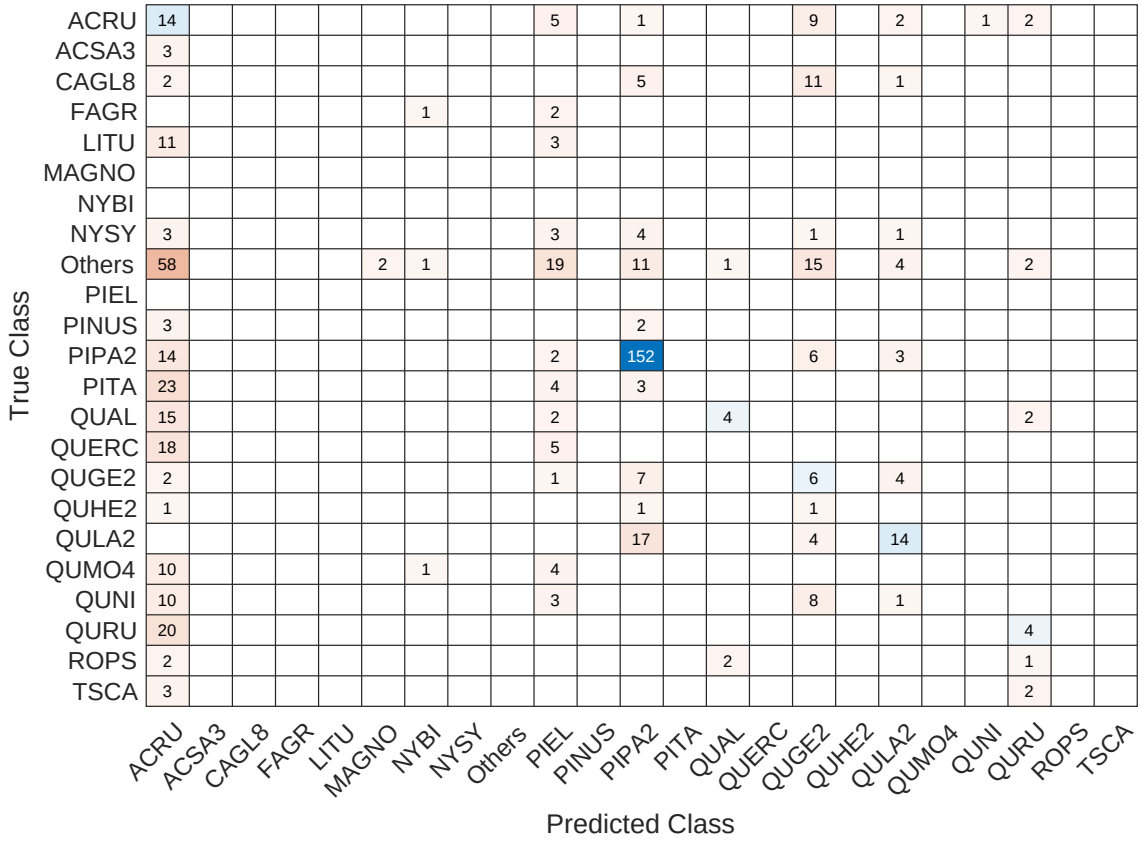


Figure 4-21. Confusion matrix on testing data using pMILd with LDA dimensionality reduction and majority voting

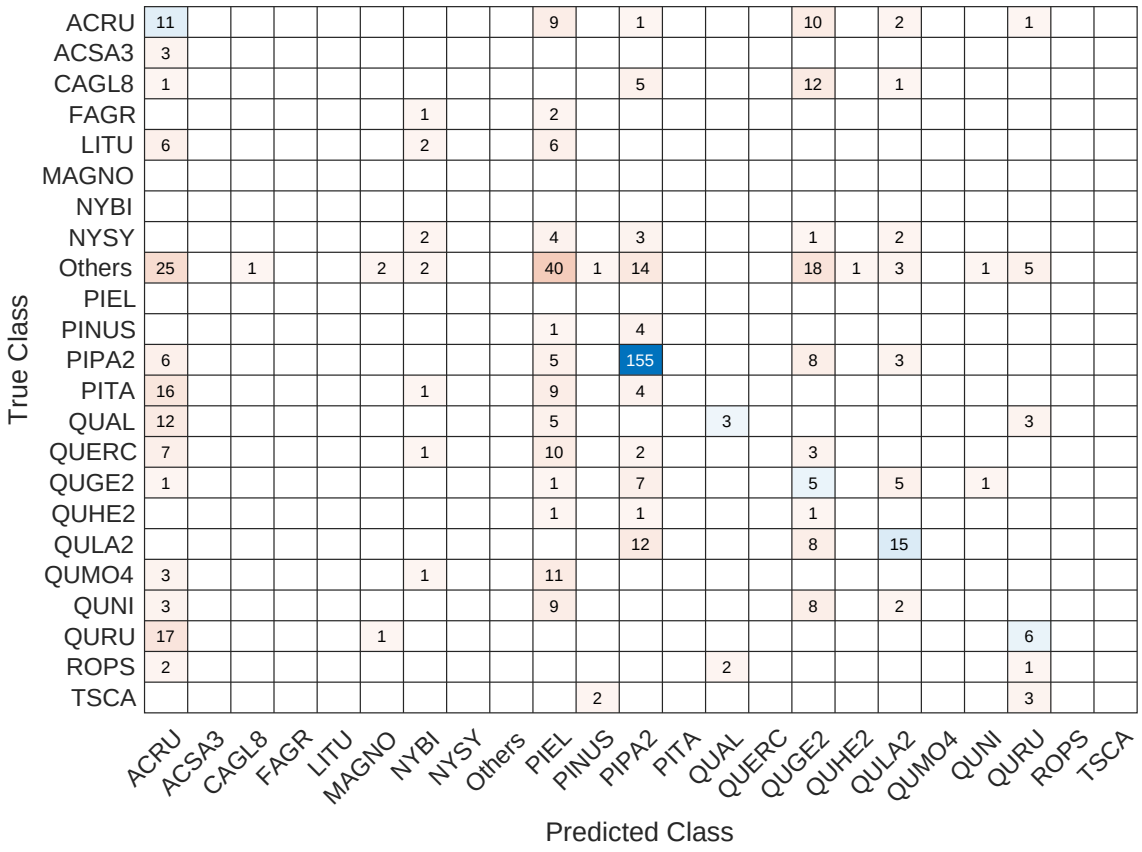


Figure 4-22. Confusion matrix on testing data using pMILd with LDA dimensionality reduction and confidence aggregation and calibration

The standard (majority voting based) pMILd method and pMILd with confidence aggregation and calibration are both used on this dataset. The confusion matrices of the testing data are shown in Figure 4-21 and 4-22, respectively. The results show that the performance of classification of each class is highly affected by the training samples of that class. Most of the mispredicted trees are misclassified as ACRU and PIPA2, the two most dominant classes. The quantitative evaluation are shown in the next subsection.

4.2.2.2 Classification with KL divergence as dimensionality reduction

The proposed pMILd with KL divergence based dimensionality reduction method is also used on the NEON HSI dataset. Since the NEON HSI dataset is very unbalanced, fixing the two hyperparameters min_{gap} and max_{peak} could lead to numerical issues. In other words, On one hand, if a fixed high dimensionality is selected by assigning a high max_{peak}, the model

parameters of classes with very limited training samples are more likely to overfit or even become singular. A rule of thumb is that the number of training samples should be over the number of model parameters to be estimated. On the other hand, if a fixed low dimensionality is selected, the classes with enormous training samples could be underfit.

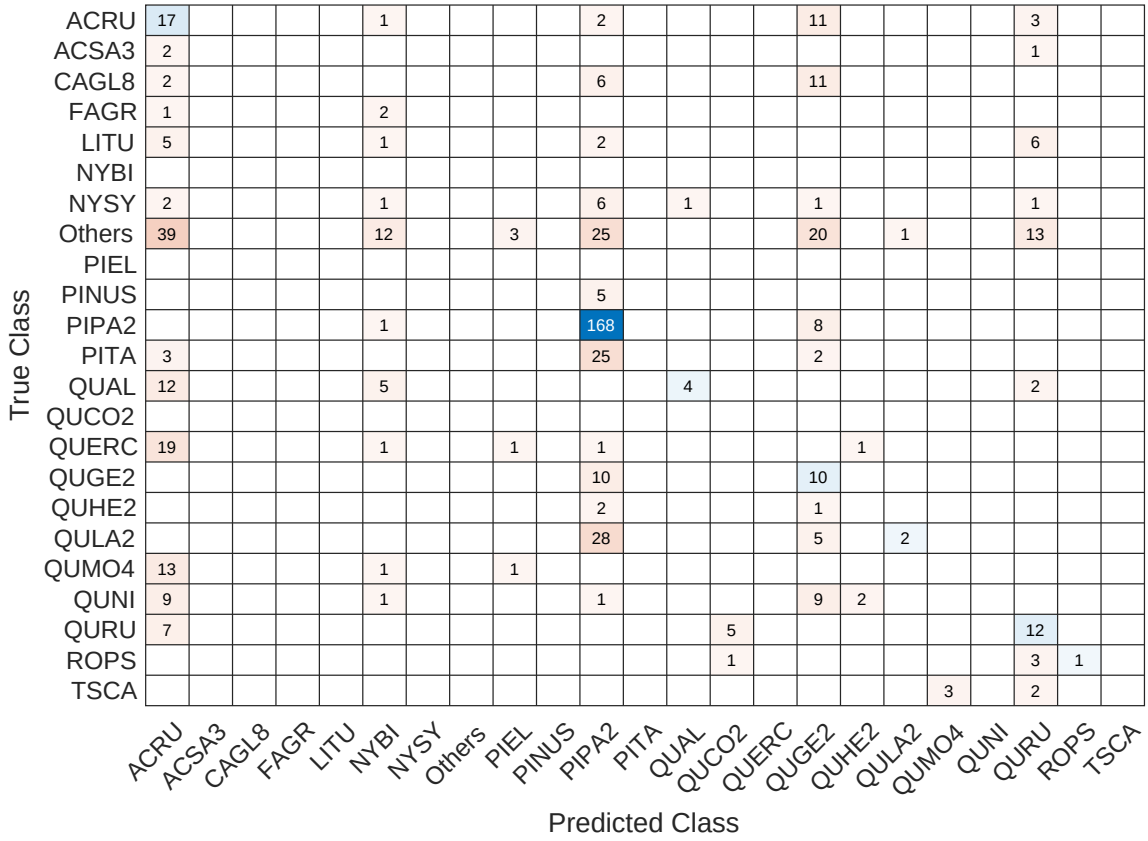


Figure 4-23. Confusion matrix on testing data using pMILd with KL dimensionality reduction and majority voting

Therefore, the following class-dependent strategy to select hyperparameters is proposed for each pairwise classifier.

(1) The first step is to select the number of Gaussian components for positive and negative classes, K and Q, based on training size (the number of training sample of the class). Let N_p and N_n for the total number of samples for positive and negative classes. If $N_p < 100$, then $K = 1$. If $100 \leq N_p \leq 2000$, then $K = 2$. If $N_p \geq 2000$, then $K = 3$. The same method is used to select Q. Note that this rule is only used for this dataset with limited training

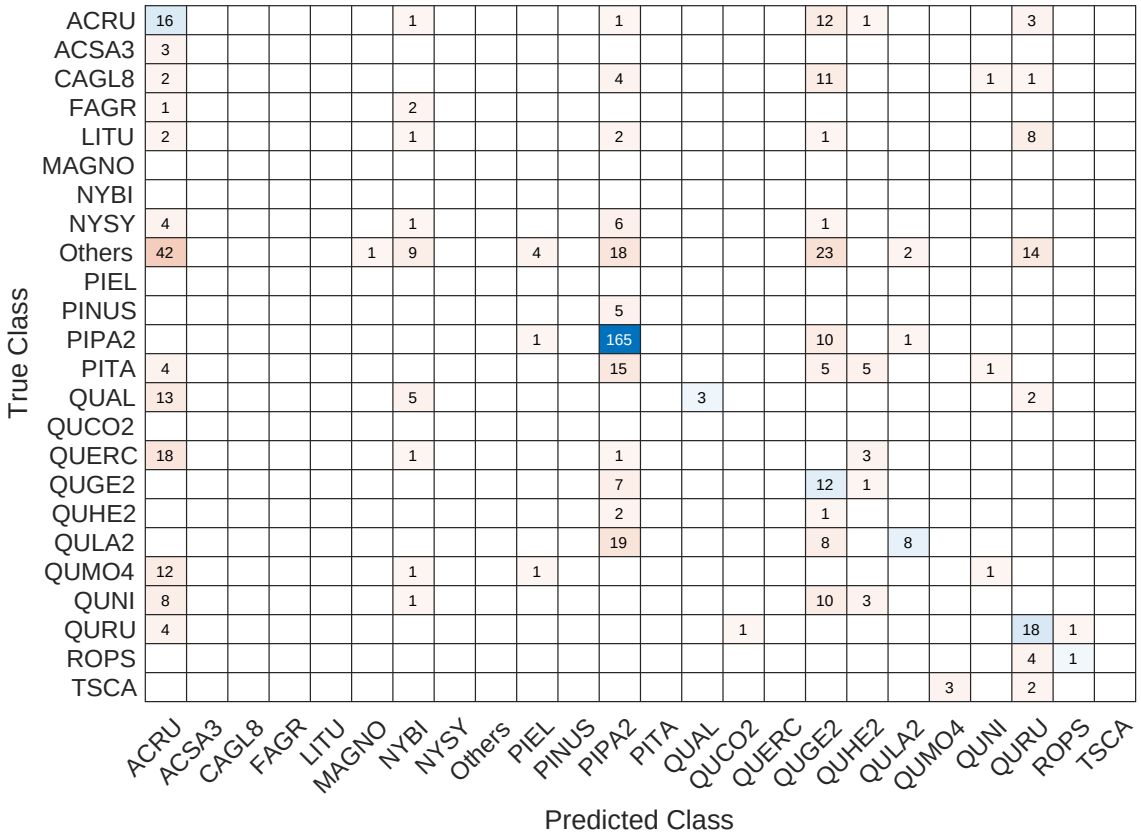


Figure 4-24. Confusion matrix on testing data using pMILd with KL dimensionality reduction and confidence aggregation and calibration

samples to prevent overfitting. For any other datasets with enough samples, K and Q can be set as a high value and the pruning process proposed can prune the unnecessary Gaussian components.

(2) The second step is to select the max_{peak} based on training size using Equation 4–3. For positive and negative classes, the upper bound of dimensionality d_p and d_n are shown in Equation 4–1 and 4–2. The term $\frac{d_p(d_p+1)}{2}$ represents the number of parameters in each Gaussian covariance and d_p denote the number of parameters in each Gaussian mean.

$$\text{Solve} \quad d_p : \left(\frac{d_p(d_p+1)}{2} + d_p\right) \times K = N_p \tag{4–1}$$

$$\text{Solve} \quad d_n : \left(\frac{d_n(d_n+1)}{2} + d_n\right) \times K = N_n \tag{4–2}$$

$$max_{peak} = \min([d_p, d_n]) \tag{4–3}$$

(3) The third step is to select min_{gap} based on the selected max_{peak}. The idea is that starting from $min_{gap} = 1$, gradually increasing min_{gap} with the increment of 1 and rerun the peak search method until the number of peaks detected is less than max_{peak} .

The standard (majority voting based) pMILd method and pMILd with confidence aggregation and calibration are both used on this dataset. The confusion matrices of the testing data are shown in Figure 4-23 and 4-24, respectively.

The accuracy, macro-F1 score and weighted F1 score are shown in Table 4-2. The quantitative results show that using the proposed KL divergence based dimensionality reduction and confidence aggregation and calibration technique yield the best performance in terms of overall classification accuracy and macro F1 score, and the second best performance of weighted F1 score.

Table 4-2. Quantitative evaluation of NEON dataset

Method	accuracy	macro F1	weighted F1
LDA, majority voting	0.354	0.104	0.319
LDA, confidence aggregation	0.333	0.087	0.302
KL, majority voting	0.374	0.102	0.281
KL, confidence aggregation	0.381	0.105	0.307

4.2.2.3 Class variability interpretation

The proposed method estimates a GMM to model both class in each pairwise classifier. Previously, the experiments on UCSB dataset demonstrate that different Gaussian components could capture the multi-modal within-class variability of tree species. Each Gaussian component denotes a subset of tree bags or pixels in the trees that is different from other subsets of the same tree species, such as QUBE trees with different colors. Learning the class variability using GMM not only models each class more precisely (compared with only learning one or multiple prototypes, or a single Gaussian), but more importantly offers a visualization of

class variability using pixel-level confidence map with respect to each Gaussian component for the ecologist or biologist to interpret.

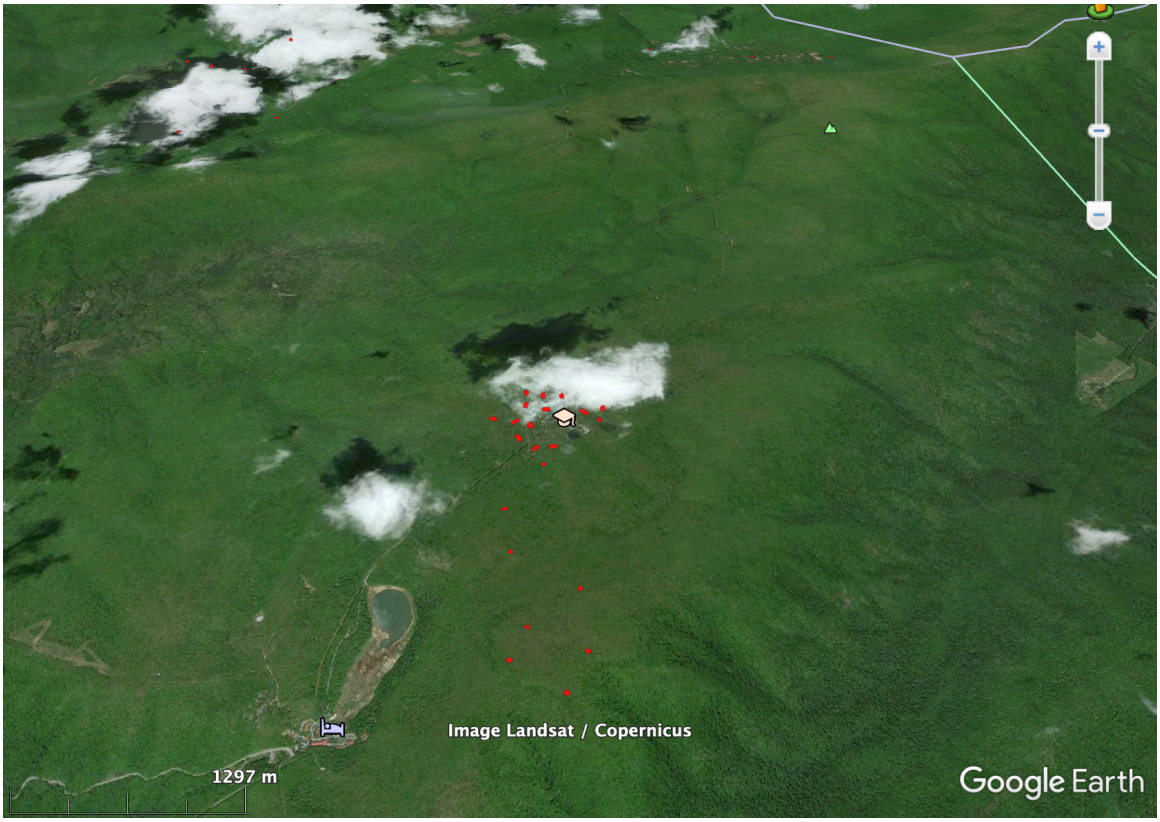


Figure 4-25. ACRU tree distribution (competition data only) in the MLBS site

For the NEON dataset, the class variability of ACRU class is high since it is one of the dominant tree species across a large geographic regions. The estimated ACRU GMM from ACRU-vs-QURU classifier is used as an example. The total number of ACRU boxes is 138. The majority of ACRU bounding boxes are distributed among three different areas of MLBS site (shown in Figure 4-25) and less than 10 ACRU are at OSBS site (not shown here). The pMILd algorithm starts with $K = 3$ Gaussian components for ACRU and is pruned to be $K = 2$ after converging.

The tree level membership with respect to the first ACRU Gaussian component are shown in Figure 4-26 and 4-27. The ACRU trees are concentrated in the low latitude region (latitude between 4134000 and 4136000), high latitude region (latitude between 4140000 and

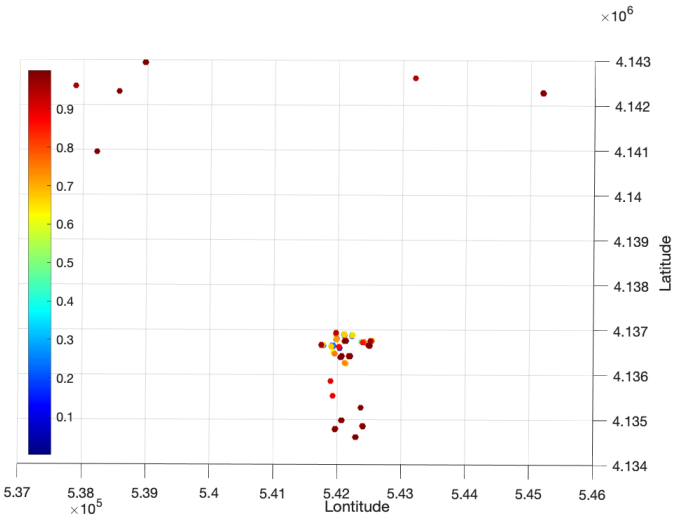


Figure 4-26. Relationship between 1st Gaussian membership and geographic location for ACRU (2D view)

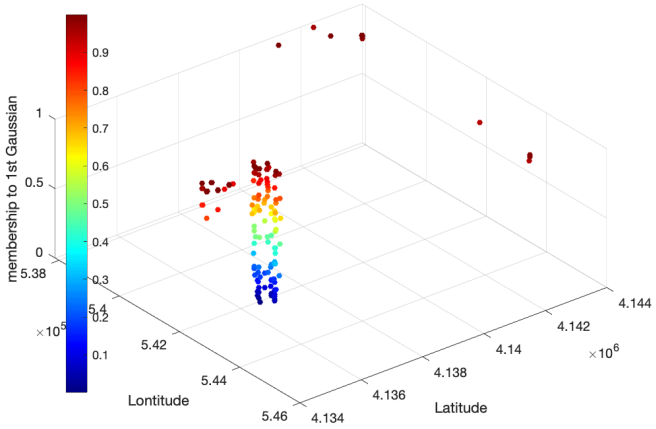


Figure 4-27. Relationship between 1st Gaussian membership and geographic location for ACRU (3D view)

4143000) and partial concentrated in the median latitude region (latitude between 4136000 and 4137000). Therefore, it can be inferred that the spectral signatures of ACRU trees, at locations which are even far away from each other (low and high latitude regions), are similar and their variability can be captured by first Gaussian component. While for ACRU trees at median latitude region, they have a much larger spectral variability. In other words, some of ACRU trees at this region are similar to the ACRU at high and low latitude region while some other ACRU trees have relatively more distinct spectral signatures. These more distinct

ACRU trees are automatically detected by the proposed model by being assigned with a low membership with respect to the first ACRU Gaussian component.

The pixel level membership with respect to the first ACRU Gaussian component also provides variability information at smaller scale. A simple and easy-to-use visualization tool is also developed on the QGIS platform, as shown in Figure. 4-28. The estimated pixel level membership values are overlaid on the original ACRU RGB images. The degree of red denotes the membership value for each pixel. Biologists can use this tool to further investigate the reasons causing the within-class variability from a biological perspective with the help of the membership map estimated by the proposed method.

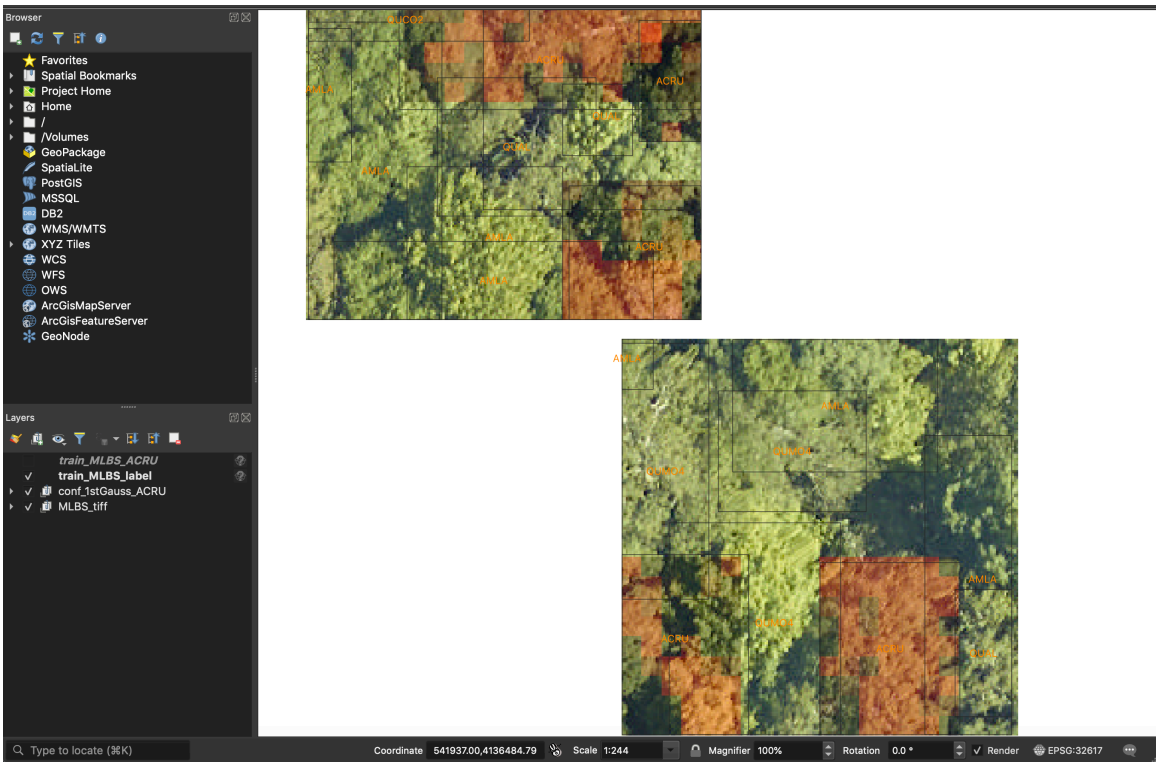


Figure 4-28. Pixel level membership visualization using QGIS (red: high membership)

Overall, at the large scale (e.g., MLBS site), ACRU trees at different locations show a large difference in regarding to spectral signatures, which can be captured by different Gaussian components of the proposed method. At the smaller scale (e.g., high latitude region

of MLBS site), ACRU trees at the same location shows a relatively smaller and simpler spectral variability, which is estimated and characterized by each corresponding Gaussian component.

The phenomenon that each Gaussian component of the pMILd method models the class variability at each geographic location is also widely seen for other species classes, such as QURU. $K = 2$ is set as the number Gaussian components for QURU. The estimated membership to each of the two Gaussian components are shown in Figure 4-29. The first Gaussian component captures a set of QURU trees in a region with tree ID less than 7000 while another Gaussian component represent a group of QURU trees in a region with tree ID larger than 7000.

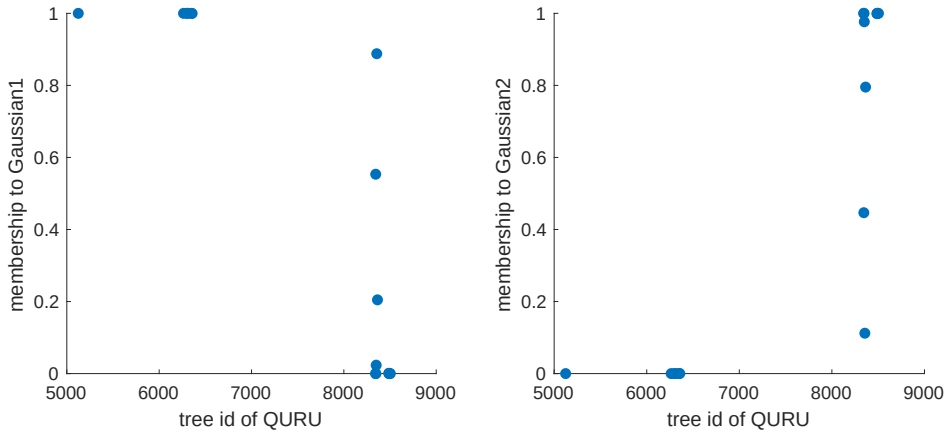


Figure 4-29. Relationship between 2nd Gaussian membership and tree IDs of QURU

4.2.3 Diabetic Retinopathy Classification on DIARETDB1 Data

One of the major task of the diabetic retinopathy classification in the retinal images is to detect the lesion based on the retinal images. In this section, the proposed pMILd method is used on the DIARETDB1 dataset. This dataset contains 89 images with weakly labeled ground truth by 4 doctors. These weakly labels polygons are considered as MIL bags. In addition, the bag label probabilities can also be generated by majority voting of 4 doctors for each pixel. Both the baseline pMILd (no probabilistic label) and the pMILd using probabilistic labels are investigated on the hard exudates (one of the diabetic retinopathy symptoms) classification.

4.2.3.1 Image pre-processing and feature extraction

The retinal images collected from different patients at different time may have different resolutions, illuminations and contrasts. Therefore, the images are pre-processed with an illumination equalization and contrast enhancement approach (Zhou et al., 2017).

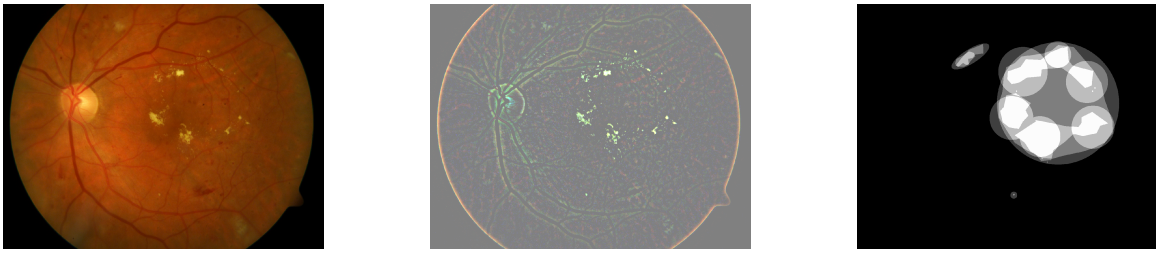


Figure 4-30. Left figure: RGB image of image 15, Middle figure: image after preprocessing, Right figure: ground truth for hard exudates

It has been shown that green and red channels are widely adopted by models in literature for diabetic retinopathy classification (Sisodia et al., 2017). Thus, the red and green intensity values are extracted as the feature vector after image pre-processing.

4.2.3.2 Classification with non-probabilistic bag labels

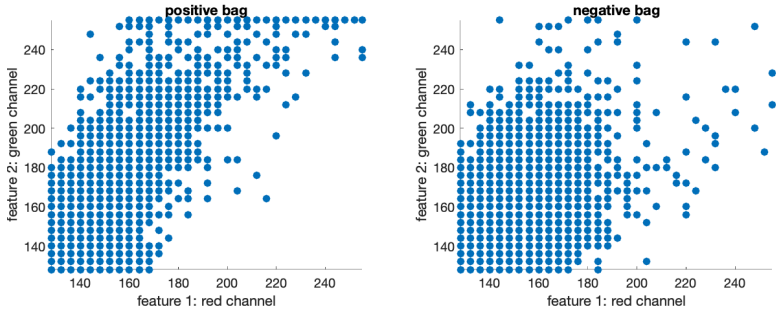


Figure 4-31. Left figure: The feature space (by sampling 1% from the dataset) of positive bags, Right figure: The feature space of negative bags

For the case of no probabilistic bag label, the positive bags are the regions where at least one doctor labeled as hard exudates. For instance, any pixels where intensity values on the right figure of Figure 4-30 that are not zero are regarded as positive instances. The number of Gaussian components are 2 for both positive and negative bags, respectively.

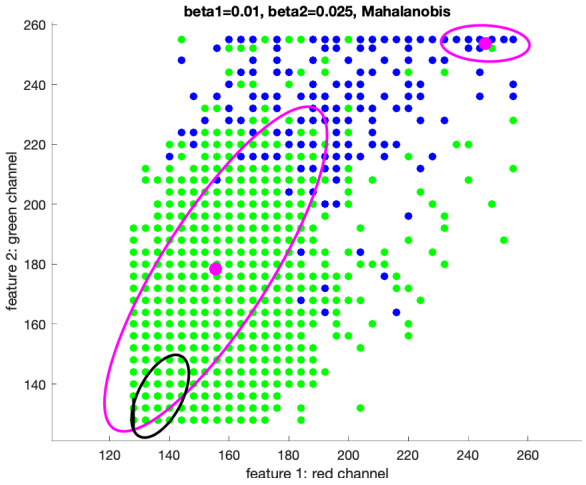


Figure 4-32. Estimated GMM for positive and negative class using non-probabilistic MIL labels. Blue: positive instances. Green: negative instances. Magenta: GMM for positive class; Black: GMM for negative class

The estimated GMM for both positive and negative class are shown in Figure 4-32 and the estimated pixel confidence values are shown in Figure 4-33. As can be seen from both figures, the high density regions on feature space for true positive instances are close to one of the target Gaussian mean (246 for red and 254 for green), which are around the top right edge of the feature space. In addition, for negative class, the high density regions are near the bottom right edge of the feature space.

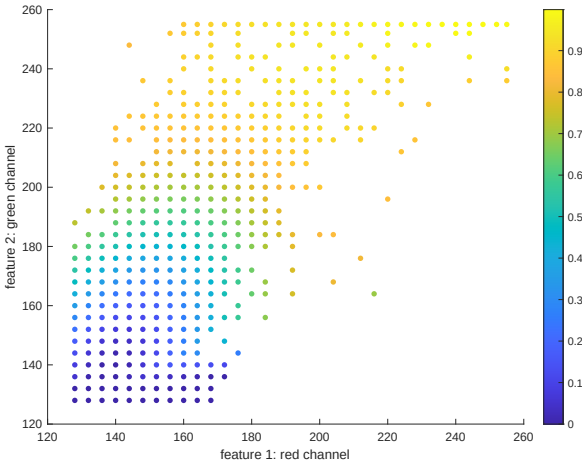


Figure 4-33. Pixel-level confidence estimated for training data

For the testing phase, the pixel-level confidence map can also be calculated for each testing image. The confidence map of testing image 15 in Figure 4-30 is shown in Figure 4-34.

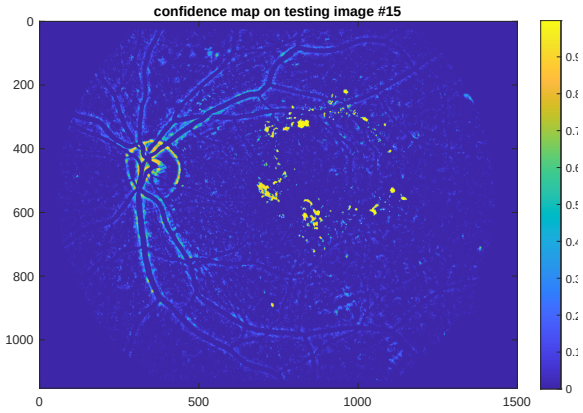


Figure 4-34. Estimated confidence map for a testing image (model is trained with non-probabilistic MIL labels)

The confidence map can successfully assign high confidence value (around 0.95) for the true positive regions. However, there are some parts on the image, such as optic disc, has a very similar color as the hard exudate. For these confusers, they also show a high confidence value to be positive class. Future work will be either to mask out the optic disc region in the image pre-precessing step or to use a different feature extraction method by considering the shape or texture to replace the RGB features.

The challenge of this dataset is that pixel level ground truth information is not available. Thus, the evaluation can only be on the lesion level (region) or image level. A 5-fold cross validation and following quantitative evaluation was performed on the region level. The details of the process are shown below.

The first step is to split the data into training and testing set. For each fold, randomly split the data into 50 images for training and 39 images for testing.

The second step is to extract target and background regions for each image. Splitting each image (images containing hard exudates) into two regions, where the target region is the image region that all 4 doctors assigned with highest confidence level and background region is

the image region that all 4 doctors assigned with zero confidence value. These regions are the best one can use for evaluation without the pixel-level ground truth.

The third step is to train the pMILd model using training set. In addition, for the training image regions, estimate the pixel level confidence and then aggregate to region level confidence value (averaging) for each target region and background region. Find a threshold such that the number of misclassified region is minimized.

The fourth step is the testing phase. For the testing image regions, estimate the pixel level confidence and then aggregate to region level confidence value (averaging) for each target region and background region. If the region level confidence is larger than the estimated threshold, it is classified as target region. Otherwise, it is classified as background region.

The last step is the evaluation phase. Estimate the accuracy of classification in term of the number of regions that are correctly classified on the testing set.

The mean of the accuracy values on the testing regions over 5 replicas is 0.994 and the standard deviation value of the accuracy values is 0.014.

4.2.3.3 Classification with probabilistic bag labels

Since there are 4 doctors and each doctor selected a confidence level out of three confidence levels for training pixels, there are totally 12 confidence levels using majority voting. Thus, the probabilistic labels are discretized as $P(B_j^+) \in \{ \frac{1}{12}, \frac{2}{12}, \ldots , \frac{12}{12}\}$.

As can be seen in Figure 4-35, one of the estimated Gaussian components of positive class is very close to top right of feature space with very small variance on green channel, leading to a further increased estimated confidence values of true hard exudates pixels for testing image shown in Figure 4-36. This reason is that by incorporating the probabilistic labels, these training pixels contributes more for estimating the mean and covariance of GMM, if more experts labeled them as target or with higher labeling confidence level.

Similarly, the quantitative evaluation was also performed for the proposed pMILd method with probabilistic bag labels. The same seeds used for splitting the training and testing sets in the Section 4.2.3.2 are also applied in the experiments of this section for a consistent

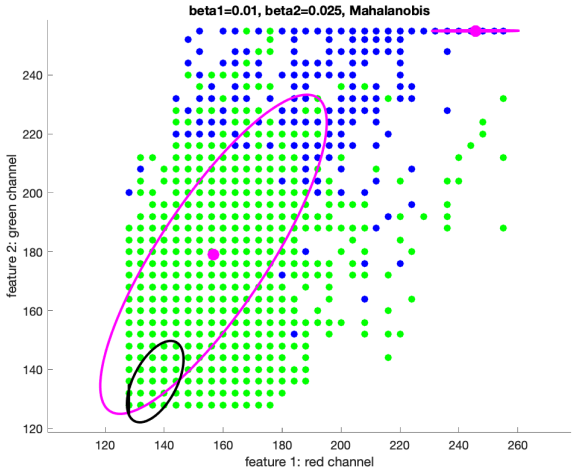


Figure 4-35. Estimated GMM for positive and negative class using probabilistic MIL labels. Blue: positive instances. Green: negative instances. Magenta: GMM for positive class; Black: GMM for negative class

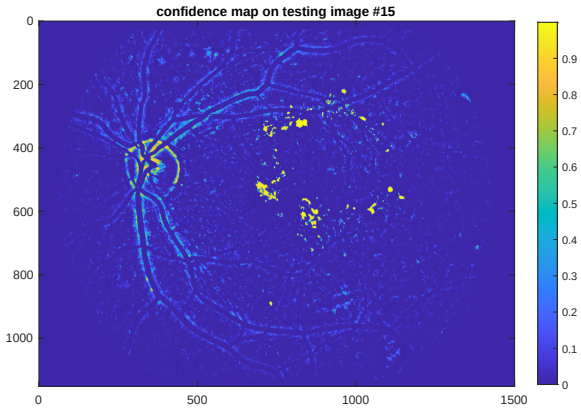


Figure 4-36. Estimated confidence map for a testing image (model is trained with probabilistic MIL labels)

comparison. The mean of the accuracy values on the testing regions over 5 replicas is 0.994 and the standard deviation value of the accuracy values is 0.014. The accuracy values are the same as the results without using probabilistic labels. The major reason is that the pMILd method without probabilistic labels has already yield high performance and the number of target pixels are still too few in the target regions so that the improvement of estimation on the pixel level using probabilistic label can barely seen on the region level for this dataset.

Therefore, metrics that shown in Equation 4–4 and 4–5 are proposed to analyze the degree of estimated target confidence value changes for both target and background region before and after incorporating probabilistic bag labels.

$$\Delta p_t = 0.5 * \frac{\sum_{n=1}^{N} p_n^{t,non-prob} - \sum_{n=1}^{N} p_n^{t,prob}}{\sum_{n=1}^{N} p_n^{t,non-prob}} + 0.5 * \frac{\sum_{n=1}^{N} p_n^{t,non-prob} - \sum_{n=1}^{N} p_n^{t,prob}}{\sum_{n=1}^{N} p_n^{t,prob}} \quad (4\text{–}4)$$

$$\Delta p_b = 0.5 * \frac{\sum_{n=1}^{N} p_n^{b,non-prob} - \sum_{n=1}^{N} p_n^{b,prob}}{\sum_{n=1}^{N} p_n^{b,non-prob}} + 0.5 * \frac{\sum_{n=1}^{N} p_n^{b,non-prob} - \sum_{n=1}^{N} p_n^{b,prob}}{\sum_{n=1}^{N} p_n^{b,prob}} \quad (4\text{–}5)$$

where N is the number of images, $p_n^{t,non-prob}$ denotes the estimated region level confidence for the n-th target region using the non-probabilistic pMILd method, $p_n^{t,prob}$ denotes the estimated region level confidence for the n-th target region using the probabilistic pMILd method, $p_n^{b,non-prob}$ denotes the estimated region level confidence for the n-th background region using the non-probabilistic pMILd method and $p_n^{b,prob}$ denotes the estimated region level confidence for the n-th background region using the probabilistic pMILd method. Thus, the ratio of confidence change, between using the non-probabilistic and probabilistic pMILd method, is Δp_t for target regions and Δp_b for background regions. For each replica, a Δp_t and a Δp_b are computed.

The mean value of Δp_t over 5 replicas is 0.0037 and the standard deviation is 0.0058. The mean value of Δp_b over 5 replicas is 0.0152 and the standard deviation is 0.0049. The results demonstrates that the probabilistic pMILd yields a much more decreasing ($1.52\% \pm 0.49\%$) of target confidence values in background regions, compared with the decreasing ($0.37\% \pm 0.58\%$) in target regions.

CHAPTER 5
CONCLUSIONS

A novel method based on Multiple Instance learning (MIL) from a probabilistic perspective was developed and tested. The probabilistic Multiple Instance learning with distributions (pMILd) introduces probabilistic distributions to represent classes within the MIL framework, capturing the complex, multi-modal class variability in the real application. In addition, one type of multi-imprecise label, the probabilistic MIL bag label, was introduced. The pMILd is also capable of learning from data with this type of multi-imprecise label.

The high scalability of the pMILd algorithm allows a selectable distribution, including Gaussian distribution, Gaussian Mixture Model, based on data and task. A multi-class classification framework, based on the pMILd, was also proposed. A novel dimensionality reduction method based on KL divergence was introduced for data with high dimensionality and small training size. Investigation into methods to automatically select hyperparameters of both this dimensionality reduction method the pMILd method can be done.

The interpretability of the pMILd model offers visualization for class variability on both the bag level and instance level using distributions. Study into how to use the class variability interpretation from the pMILd method to assist the domain experts to better understand the dataset can be done.

REFERENCES

Ambroise, Christophe, Denoeux, Thierry, Govaert, Gérard, and Smets, Philippe. "Learning from an imprecise teacher: probabilistic and evidential approaches." *Applied Stochastic Models and Data Analysis* 1 (2001): 100–105.

Amores, Jaume. "Multiple instance classification: Review, taxonomy and comparative study." *Artificial Intelligence* 201 (2013): 81–105.

Andrews, Stuart, Tsochantaridis, Ioannis, and Hofmann, Thomas. "Support vector machines for multiple-instance learning." *Advances in neural information processing systems*. 2003, 577–584.

Asner, Gregory P, Bustamante, Mercedes MC, and Townsend, Alan R. "Scale dependence of biophysical structure in deforested areas bordering the Tapajos National Forest, Central Amazon." *Remote Sensing of Environment* 87 (2003).4: 507–520.

Bateson, C Ann, Asner, Gregory P, and Wessman, Carol A. "Endmember bundles: A new approach to incorporating endmember variability into spectral mixture analysis." *IEEE transactions on geoscience and remote sensing* 38 (2000).2: 1083–1094.

Belongie, Serge, Malik, Jitendra, and Puzicha, Jan. "Shape matching and object recognition using shape contexts." Tech. rep., CALIFORNIA UNIV SAN DIEGO LA JOLLA DEPT OF COMPUTER SCIENCE AND ENGINEERING, 2002.

Bioucas-Dias, José M, Plaza, Antonio, Dobigeon, Nicolas, Parente, Mario, Du, Qian, Gader, Paul, and Chanussot, Jocelyn. "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches." *IEEE journal of selected topics in applied earth observations and remote sensing* 5 (2012).2: 354–379.

Bovolo, Francesca, Bruzzone, Lorenzo, and Carlin, Lorenzo. "A novel technique for subpixel image classification based on support vector machine." *IEEE Transactions on Image Processing* 19 (2010).11: 2983–2999.

Brodley, Carla E, Friedl, Mark A, et al. "Identifying and eliminating mislabeled training instances." *Proceedings of the National Conference on Artificial Intelligence*. 1996, 799–805.

Canham, Kelly, Schlamm, Ariel, Ziemann, Amanda, Basener, Bill, and Messinger, David. "Spatially adaptive hyperspectral unmixing." *IEEE Transactions on Geoscience and Remote Sensing* 49 (2011).11: 4248–4262.

Castrodad, Alexey, Xing, Zhengming, Greer, John B, Bosch, Edward, Carin, Lawrence, and Sapiro, Guillermo. "Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery." *IEEE Transactions on Geoscience and Remote Sensing* 49 (2011).11: 4263–4281.

Chen, Yixin, Bi, Jinbo, and Wang, James Ze. "MILES: Multiple-instance learning via embedded instance selection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006).12: 1931–1947.

Combe, J-Ph, Le Mouélic, S, Sotin, C, Gendrin, A, Mustard, JF, Le Deit, L, Launeau, P, Bibring, J-P, Gondet, B, Langevin, Y, et al. "Analysis of OMEGA/Mars express data hyperspectral data using a multiple-endmember linear spectral unmixing model (MELSUM): Methodology and first results." *Planetary and Space Science* 56 (2008).7: 951–975.

Denœux, Thierry and Zouhal, Lalla Meriem. "Handling possibilistic labels in pattern classification using evidential reasoning." *Fuzzy sets and systems* 122 (2001).3: 409–424.

Diebolt, Jean and Ip, Eddie HS. "Stochastic EM: method and application." *Markov chain Monte Carlo in practice*. Springer, 1996. 259–273.

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

Dong, Lin. *A comparison of multi-instance learning algorithms*. Ph.D. thesis, The University of Waikato, 2006.

Du, Xiaoxiao, Zare, Alina, Gader, Paul, and Dranishnikov, Dmitri. "Spatial and spectral unmixing using the beta compositional model." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7 (2014).6: 1994–2003.

Eches, O., Dobigeon, N., Mailhes, C., and Tourneret, J. Y. "Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery." *IEEE Transactions on Image Processing* 19 (2010a).6: 1403–1413.

Eches, O., Dobigeon, N., and Tourneret, J. Y. "Estimating the number of endmembers in hyperspectral images using the normal compositional model and a hierarchical Bayesian algorithm." *IEEE Journal of Selected Topics in Signal Processing* 4 (2010b).3: 582–591.

Eskin, Eleazar. "Detecting errors within a corpus using anomaly detection." *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, 2000, 148–153.

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

Frénay, Benoît and Verleysen, Michel. "Classification in the presence of label noise: a survey." *IEEE transactions on neural networks and learning systems* 25 (2014).5: 845–869.

Gaba, Anil and Winkler, Robert L. "Implications of errors in survey data: a Bayesian model." *Management Science* 38 (1992).7: 913–925.

Gader, Paul, Zare, Alina, Close, Ryan, Aitken, Jen, and Tuell, Grady. "Muufl gulfport hyperspectral and lidar airborne data set." *Univ. Florida, Gainesville, FL, USA, Tech. Rep. REP-2013-570* (2013).

Gärtner, Thomas, Flach, Peter A, Kowalczyk, Adam, and Smola, Alexander J. "Multi-instance kernels." *ICML*. vol. 2. 2002, 179–186.

Geng, Xin. "Label distribution learning." *IEEE Transactions on Knowledge and Data Engineering* 28 (2016).7: 1734–1748.

Goenaga, Miguel A, Torres-Madronero, Maria C, Velez-Reyes, Miguel, Van Bloem, Skip J, and Chinea, Jesus D. "Unmixing analysis of a time series of Hyperion images over the Guánica dry forest in Puerto Rico." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 6 (2013).2: 329–338.

Hüllermeier, Eyke and Beringer, Jürgen. "Learning from ambiguously labeled examples." *Intelligent Data Analysis* 10 (2006).5: 419–439.

Jiao, Changzhe, Chen, Chao, McGarvey, Ronald G, Bohlman, Stephanie, Jiao, Licheng, and Zare, Alina. "Multiple instance hybrid estimator for hyperspectral target characterization and sub-pixel target detection." *ISPRS journal of photogrammetry and remote sensing* 146 (2018): 235–250.

Jiao, Changzhe and Zare, Alina. "Functions of multiple instances for learning target signatures." *IEEE Transactions on Geoscience and Remote Sensing* 53 (2015).8: 4670–4686.

Jiao, Changzhe, Zare, Alina, and McGarvey, Ronald G. "Multiple Instance Hybrid Estimator for Hyperspectral Target Characterization and Sub-pixel Target Detection." *arXiv preprint arXiv:1710.11599* (2017).

Jin, Rong and Ghahramani, Zoubin. "Learning with multiple labels." *Advances in neural information processing systems*. 2003, 921–928.

Joseph, Lawrence, Gyorkos, Theresa W, and Coupal, Louis. "Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard." *American journal of epidemiology* 141 (1995).3: 263–272.

Kazianka, H. "Objective Bayesian analysis for the normal compositional model." *Computational Statistics & Data Analysis* 56 (2012).6: 1528–1544.

Lloyd, Stuart. "Least squares quantization in PCM." *IEEE transactions on information theory* 28 (1982).2: 129–137.

Maron, Oded and Lozano-Pérez, Tomás. "A framework for multiple-instance learning." *Advances in neural information processing systems*. 1998, 570–576.

Meerdink, Susan K, Roberts, Dar A, Roth, Keely L, King, Jennifer Y, Gader, Paul D, and Koltunov, Alexander. "Classifying California plant species temporally using airborne hyperspectral imagery." *Remote Sensing of Environment* 232 (2019): 111308.

Mianji, Fereidoun A and Zhang, Ye. "SVM-based unmixing-to-classification conversion for hyperspectral abundance quantification." *IEEE Transactions on Geoscience and Remote Sensing* 49 (2011).11: 4318–4327.

Nascimento, José MP and Dias, José MB. "Vertex component analysis: A fast algorithm to unmix hyperspectral data." *IEEE transactions on Geoscience and Remote Sensing* 43 (2005).4: 898–910.

Nguyen, Quang, Valizadegan, Hamed, and Hauskrecht, Milos. "Learning classification models with soft-label information." *Journal of the American Medical Informatics Association* 21 (2014).3: 501–508.

Nowak, Eric, Jurie, Frédéric, and Triggs, Bill. "Sampling strategies for bag-of-features image classification." *European conference on computer vision*. Springer, 2006, 490–503.

Opelt, Andreas, Pinz, Axel, Fussenegger, Michael, and Auer, Peter. "Generic object recognition with boosting." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006).3: 416–431.

Pérez, Carlos Javier, Girón, F Javier, Martín, Jacinto, Ruiz, Manuel, and Rojano, Carlos. "Misclassified multinomial data: a Bayesian approach." *RACSAM* 101 (2007).1: 71–80.

Raykar, Vikas C, Yu, Shipeng, Zhao, Linda H, Valadez, Gerardo Hermosillo, Florin, Charles, Bogoni, Luca, and Moy, Linda. "Learning from crowds." *Journal of Machine Learning Research* 11 (2010).Apr: 1297–1322.

Rekaya, R, Weigel, KA, and Gianola, D. "Threshold model for misclassified binary responses with applications to animal breeding." *Biometrics* 57 (2001).4: 1123–1129.

Roberts, Dar A, Gardner, M, Church, R, Ustin, S, Scheer, G, and Green, RO. "Mapping chaparral in the Santa Monica Mountains using multiple endmember spectral mixture models." *Remote Sensing of Environment* 65 (1998).3: 267–279.

Ruiz, M, Girón, FJ, Pérez, CJ, Martín, J, and Rojano, C. "A Bayesian model for multinomial sampling with misclassified data." *Journal of Applied Statistics* 35 (2008).4: 369–382.

Serre, Thomas, Wolf, Lior, Bileschi, Stanley, Riesenhuber, Maximilian, and Poggio, Tomaso. "Robust object recognition with cortex-like mechanisms." *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2007).3: 411–426.

Sisodia, Dilip Singh, Nair, Shruti, and Khobragade, Pooja. "Diabetic retinal fundus images: Preprocessing and feature extraction for early detection of diabetic retinopathy." *Biomedical and Pharmacology Journal* 10 (2017).2: 615–626.

Somers, Ben, Asner, Gregory P, Tits, Laurent, and Coppin, Pol. "Endmember variability in spectral mixture analysis: A review." *Remote Sensing of Environment* 115 (2011).7: 1603–1616.

Somers, Ben, Zortea, Maciel, Plaza, Antonio, and Asner, Gregory P. "Automated extraction of image-based endmember bundles for improved spectral unmixing." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5 (2012).2: 396–408.

Song, Conghe. "Spectral mixture analysis for subpixel vegetation fractions in the urban environment: How to incorporate endmember variability?" *Remote Sensing of Environment* 95 (2005).2: 248–263.

Stein, David. "Application of the normal compositional model to the analysis of hyperspectral imagery." *Advances in techniques for analysis of remotely sensed data, 2003 IEEE Workshop on*. IEEE, 2003, 44–51.

Teng, Choh Man. "Evaluating noise correction." *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2000, 188–198.

Teng, Choh-Man. "A Comparison of Noise Handling Techniques." *FLAIRS Conference*. 2001, 269–273.

Teng, Choh Man. "Dealing with data corruption in remote sensing." *International Symposium on Intelligent Data Analysis*. Springer, 2005, 452–463.

Wang, Jun and Zucker, Jean-Daniel. "Solving multiple-instance problem: A lazy learning approach." (2000).

Yan, Shuicheng, Wang, Huan, Tang, Xiaoou, Liu, Jianzhuang, and Huang, Thomas S. "Regression from uncertain labels and its applications to soft biometrics." *IEEE Transactions on Information Forensics and Security* 3 (2008).4: 698–708.

Zare, A. and Gader, P. "PCE: Piecewise convex endmember detection." *IEEE Transactions on Geoscience and Remote Sensing* 48 (2010).6: 2620–2632.

Zare, A., Gader, P., and Casella, G. "Sampling piecewise convex unmixing and endmember extraction." *IEEE Transactions on Geoscience and Remote Sensing* 51 (2013).3: 1655–1665.

Zare, Alina and Ho, KC. "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing." *IEEE Signal Processing Magazine* 31 (2014).1: 95–104.

Zare, Alina, Jiao, Changzhe, and Glenn, Taylor. "Discriminative multiple instance hyperspectral target characterization." *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2017).1: 1–1.

Zhang, B., Zhuang, L., Gao, L., Luo, W., Ran, Q., and Du, Q. "PSO-EM: a hyperspectral unmixing algorithm based on normal compositional model." *IEEE Transactions on Geoscience and Remote Sensing* 52 (2014).12: 7782–7792.

Zhang, Jianguo, Marszałek, Marcin, Lazebnik, Svetlana, and Schmid, Cordelia. "Local features and kernels for classification of texture and object categories: A comprehensive study." *International journal of computer vision* 73 (2007).2: 213–238.

Zhang, Min-Ling and Zhou, Zhi-Hua. "A review on multi-label learning algorithms." *IEEE transactions on knowledge and data engineering* 26 (2014).8: 1819–1837.

Zhang, Qi and Goldman, Sally A. "EM-DD: An improved multiple-instance learning technique." *Advances in neural information processing systems*. 2002, 1073–1080.

Zhang, Wensheng, Rekaya, Romdhane, and Bertrand, Keith. "A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer." *Bioinformatics* 22 (2005).3: 317–325.

Zhou, Lei, Zhao, Yu, Yang, Jie, Yu, Qi, and Xu, Xun. "Deep multiple instance learning for automatic detection of diabetic retinopathy in retinal images." *IET Image Processing* 12 (2017).4: 563–571.

Zhou, Yuan, Rangarajan, Anand, and Gader, Paul D. "A spatial compositional model for linear unmixing and endmember uncertainty estimation." *IEEE Transactions on Image Processing* 25 (2016).12: 5987–6002.

———. "A Gaussian mixture model representation of endmember variability in hyperspectral unmixing." *IEEE Transactions on Image Processing* 27 (2018).5: 2242–2256.

Zhou, Zhi-Hua. "A brief introduction to weakly supervised learning." *National science review* 5 (2018).1: 44–53.

Zhou, Zhi-Hua, Zhang, Min-Ling, Huang, Sheng-Jun, and Li, Yu-Feng. "Multi-instance multi-label learning." *Artificial Intelligence* 176 (2012).1: 2291–2320.

Zou, Sheng and Zare, Alina. "Hyperspectral unmixing with endmember variability using partial membership latent dirichlet allocation." *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, 6200–6204.

BIOGRAPHICAL SKETCH

Sheng Zou received his Bachelor of Science degree in applied physics from the Northeastern University of China in 2013. From 2013 to 2016, he continued his studies at the University of Missouri-Columbia to graduate with his Master of Science degree in computer engineering from the Department of Electrical and Computer Engineering in 2016. His research interests include machine learning, hyperspectral classification and unmixing, remote sensing and image processing.