CLUSTERING OF MULTIPLE INSTANCE DATA

Andrew D. Karem
B.S., Rice University, 2002
M.S., University of Louisville, 2007

A Dissertation Submitted to the Faculty of
the J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of CECS
University of Louisville
Louisville, Kentucky

May 2019

CLUSTERING OF MULTIPLE INSTANCE DATA

Andrew D. Karem
B.S., Rice University, 2002
M.S., University of Louisville, 2007

Dissertation Approved on

April 25, 2019

By the following Dissertation Committee

Dr. Hichem Frigui, Chair

Dr. Amir Amini

Dr. Olfa Nasraoui

Dr. Mehmed Kantardzic

Dr. Nihat Altiparmak

ACKNOWLEDGMENTS

My sincere gratitude goes to Dr. Hichem Frigui for his assistance and guidance, along with his deep and abiding patience with my admittedly slow progression through the PhD process. I want to thank the members of my committee for both their guidance and patience throughout this process as well.

I must thank my mother Anne, father David, brother Jeff, sister-in-law Kate, and nephews Rory and Ronan for being a better family than anyone deserves.

Emma, Lily, and Ruby – thanks for cheering me up and getting me outside every now and then.

This work was supported in part by U.S. Army Research Oce Grants Number W911NF-13-1-0066 and W911NF-14-1-0589. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Oce, or the U.S. Government.

ABSTRACT

CLUSTERING OF MULTIPLE INSTANCE DATA

Andrew D. Karem

April 25, 2019

An emergent area of research in machine learning that aims to develop tools to analyze data where objects have multiple representations is *Multiple Instance Learning* (MIL). In MIL, each object is represented by a bag that includes a collection of feature vectors called instances. A bag is positive if it contains at least one positive instance, and negative if no instances are positive.

One of the main objectives in MIL is to identify a region in the instance feature space with high correlation to instances from positive bags and low correlation to instances from negative bags – this region is referred to as a *target concept* (TC). Existing methods either only identify a single target concept, do not provide a mechanism for selecting the appropriate number of target concepts, or do not provide a flexible representation for target concept memberships. Thus, they are not suitable to handle data with large intra-class variation. In this dissertation we propose new algorithms that learn multiple target concepts simultaneously.

The proposed algorithms combine concepts from data clustering and multiple instance learning. In particular, we propose crisp, fuzzy, and possibilistic variations of the Multi-target concept Diverse Density (MDD) metric, along with

three algorithms to optimize them. Each algorithm relies on an alternating optimization strategy that iteratively refines concept assignments, locations, and scales until it converges to an optimal set of target concepts. We also demonstrate how the possibilistic MDD metric can be used to select the appropriate number of target concepts for a dataset. Lastly, we propose the construction of classifiers based on embedded feature space theory to use our target concepts to predict the label of prospective MIL data.

The proposed algorithms are implemented, tested, and validated through the analysis of multiple synthetic and real-world data. We first demonstrate that our algorithms can detect multiple target concepts reliably, and are robust to many generative data parameters. We then demonstrate how our approach can be used in the application of Buried Explosive Object (BEO) detection to locate distinct target concepts corresponding to signatures of varying BEO types. We also demonstrate that our classifier strategies can perform competitively with other well-established embedded space approaches in classification of Benchmark MIL data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

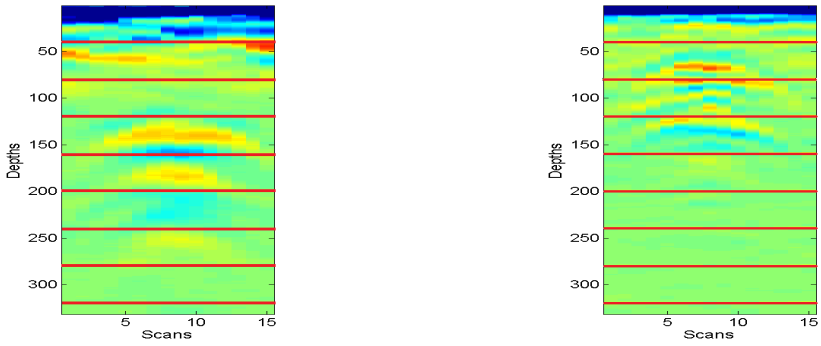
A standard machine learning task relies on the assumption that each data sample can be represented by a single feature vector. Unfortunately, there exist applications for which nature or logistics render the data impossible to describe using this singular representation. In these cases of interest, samples are instead characterized by multiple, alternate feature vectors – and it is unknown to the user which specific feature vector(s) has the correct description of the data sample. This class of problems is generally known as Multiple Instance Problems (MIP or MILP), and the class of machine learning solutions proposed to address these problems are refered to as Multiple Instance Learning (MIL). MIL analysis presents a nontrivial set of challenges that greatly complicate the utilization of conventional classifiers.

As an illustrative example, we consider the application of machine learning algorithms to automated buried explosive object (BEO) detection. In particular, we consider building a BEO classifier using data collected with a ground-penetrating

radar (GPR). First, a prescreener [52, 37, 20] detects anomalies and extracts a 3-dimensional data-cube of quantitative measurement at each location. Then, a classifier [20, 21, 50] is used to determine whether or not the characteristics of this data sample suggest it to be a BEO. During collection of the training data, the spatial location (down-track and cross-track) of the BEO can be located using ground truth and the GPS. On the other hand, the 3rd dimension (depth) of the BEO cannot be estimated without visual inspection of the data. To illustrate this phenomenon, figure 1.1 depicts the GPR signatures of the same BEO type buried at 3 inches deep in two geographically different sites. For the sake of simplicity, we only show a 2-dimensional view (down-track, depth) of the alarms for the central channel of the data cube. First, we note that the actual BEO signature does not extend over all depth values. Second, even if one is given the actual burial depth of a BEO, figure 1.1 makes it clear that the depth position within the GPR cube may still vary (depending on soil properties). Furthermore, BEO detection faces the challenge of diverse BEO types being encountered in the field. Figure 1.2 depicts two GPR data cubes representing two distinct BEO types taken from the same dataset and same site. Whereas signature for the small BEO in figure 1.2(a) spans approximately 150 depths from top to bottom, the larger BEO in figure 1.2(b) spans nearly 50 depths from top to bottom. As a consequence of these observations, extracting one global feature vector from the alarm may not discriminate between BEOs and clutter effectively.

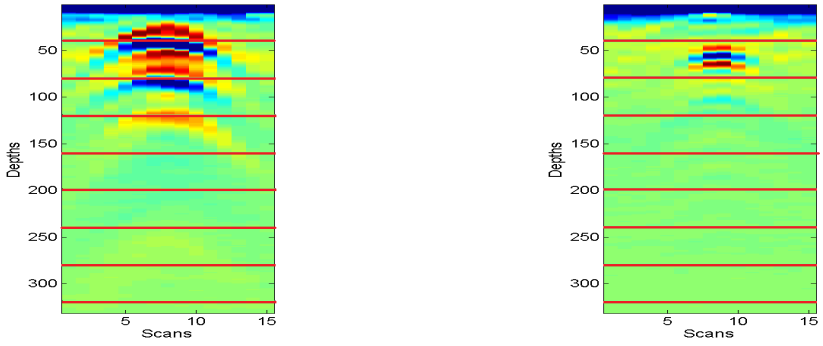
To better localize the BEO signature, many discrimination algorithms (e.g. [20, 21, 56]) divide the GPR alarm into overlapping windows along the depth.

To train such algorithms, the user needs to identify the ideal depth location for training for *each* alarm. Unfortunately, depth selection relies on an expert to visually inspect the 3-dimesional data cube and is a tedious, potentially unfeasible, task for large data sets. An alternative representation – specifically one that fits the MIL paradigm, would not require localizing the extent of the true signature within each training alarm. Instead, it would treat all overlapping windows for an alarm as a set of single instances where each instance is not labeled explicitly.



(a)BEO at site 1 (b) BEO at site 2

Figure 1.1: Depth of BEO signature depends on the soil properties of the site. The same BEO type is buried at a depth of 3 inches at both sites.



(a) Large BEO (b) Small BEO

Figure 1.2: Size of the BEO signature depends on the Type

1.2 Basic Terminology and Definitions

Approaches that work with multiple features for each data sample are referred to as *multiple instance learning (MIL)* [16]. These stand in contrast to conventional approaches that ignore the multiple instance problem, which are referred to as *single instance learning (SIL)*. Within the MIL framework, the multiple features representing one data sample are referred to as a "bag," and each individual feature within a bag is referred to as an "instance." Thus, a bag is comprised of a set of instances. Typically we represent the n^{th} bag in a dataset according to the notation B_n, and denote a positive bag as B_n^+ and a negative bag as B_n^-. We represent the i^{th} instance in bag B_n as b_{ni}. Figure 1.3 provides a 2-dimensional example of the contrast between the SIL and MIL feature spaces. For each plot, there are three positive data samples, denoted by the "+" superscript, and three negative samples, denoted by the "-" superscript. For the SIL example, each individual sample is represented by a single value in the feature space (e.g. the first sample is given as x_1^+), while for the MIL example, an individual sample has multiple representatives (e.g. the first sample, B_1^+ is shown as b_{11}^+, b_{12}^+, and b_{13}^+) It is clear that while all the positive samples on the SIL plot have similar features, only a few instances from positive samples have notably similar features for the MIL example.

An example of this representation for the BEO detection application can be observed in the same illustration of BEOs taken from multiple sites in figure 1.1. In this case one feature that is extracted from each sub-image (bounded by the

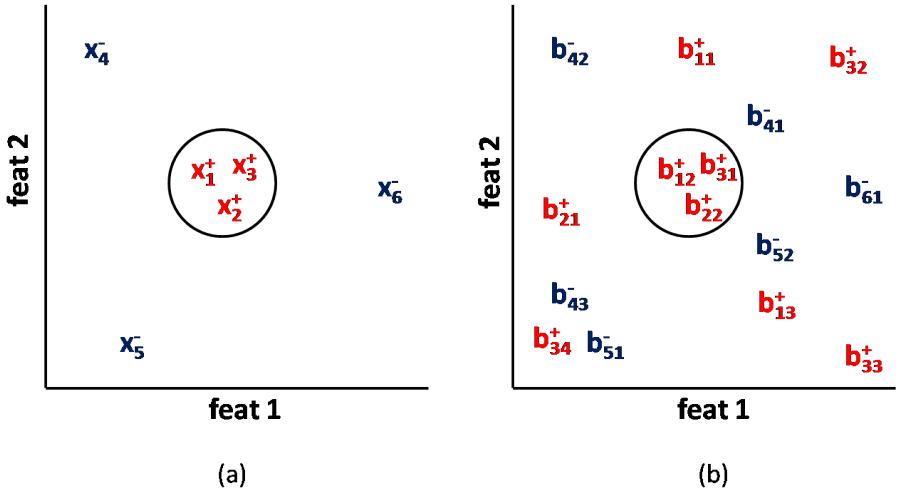


Figure 1.3: Sample Comparison of (a) Single Instance Learning (SIL) Feature Space and (b) Multiple Instance Learning (MIL) Feature Space

horizontal lines) would represent an instance. The collection of all instances from one alarm are grouped into a bag. The number of instances in a given bag is arbitrary, but must be at least one.

In the basic MIL formulation, both bags and instances carry a positive or negative label. Typically, a bag is positive if and only if at least one of its instances is positive. Concurrently, a bag is negative if and only if all of its instances are negative. More recent approaches have changed this requirement and allow for all or a subset of instances to play a role in classification. The former MIL problems rely on what is typically referred to as the Standard MIL assumption, while the latter are typically described as following the Collective or Presence-based assumptions. [5, 1].

In MIL, it is generally assumed that the label of the bags is given explicitly, but not the label of the instances. For example, in the BEO application presented in figure 1.1, we know if a given bag (i.e. entire alarm) belongs to the class of BEOs.

However, we do not know apriori which instance (i.e. sub-image) corresponds to the BEO signature and which one corresponds to background. In other words, we have access to all available bag labels when analyzing this data, but access to none of the instance labels. Typical MIL classifiers are then constructed with two potential goals in mind. On the one hand, the learned classifier may accurately predict labels at the instance level. On the other hand, it may predict labels at the bag level. While these goals are commonly synonymous (i.e. combining a set of labeled instances to predict an overall bag label, that need not always be the case, and a good bag classifier may not reliably predict instance labels or vice versa [1, 11].

To illustrate the instance-vs-bag dynamic further, we introduce another notable application suitable for MIL – automated image annotation. Given training images with global labels, the goal in automated image annotation is to construct a model that can accurately assign a subset of appropriate labels for new images. For example, figure 1.4 depicts sample images with a few labels describing their content. An effective classifier, built using conventional machine learning techniques, will require preprocessing and manual labeling. More specifically, standard single instance learning provides two main solutions to learn an appropriate classifier for automated image annotation. The first [46, 7, 60] segments each image into regions, manually labels each region, and extracts the relevant features for training. One major problem with this approach is that segmentation is a difficult problem, and there is no guarantee distinct objects will be correctly segmented. The other critical issue is that manual labeling is impractical for larger datasets. The second

SIL approach to image annotation [65, 60] is based on randomly selecting small patches from each training image. Each image patch is then manually labeled. Visual features are then extracted from the labeled image patches and used to train a classifier. While this approach may assist in circumventing the problems associated with bad image segmentation, the cost issue associated with manual labeling is likely to be as bad or worse.

A more recent and useful approach to global image annotation is based on multiple instance learning [2, 44, 9, 59]. Within this approach, images are also divided into small patches and visual features are extracted from each patch. However, labels of individual patches are not needed. Instead, features from all patches originating from the same image are grouped into a bag that has the same global labels as the image.

(a) tiger,sky,tree (b) sand,mountain,sky (c) grass,sky

Figure 1.4: Three manually annotated images

Broadly speaking, the scope of MIL research can be broken into two major branches[1] . The first of these addresses *instance-space* (IS) approaches, which rely on finding a boundary or other means of discrimination between features of instances from positive bags and negative bags [40, 16, 66]. The alternative

branch, which consists of *bag space* (BS) and *embedded space* (ES), rely on the use of a function to map each individual bag (as opposed to instance) to every other bag in a dataset, forming an entirely new set of metrics or features to model bag-to-bag relationships. Following the construction of this new feature space, conventional single-instance algorithms are used [10, 9, 67] to build a classifier. One principle that extends across both branches of MIL research is the so-called "target concept," which represents a point or region in a MIL feature space that defines or is heavily correlated with positive bags and/or instances from positive bags.

The goal of this thesis is to utilize clustering theory [29] to extend and generalize a staple instance-space MIL approach, known as the Diverse Density model [40], to robustly accomodate multiple target concepts. The proposed model, called Multi-target concept Diverse Density (MDD), strives to identify multiple dense regions in the feature space with maximal correlation to instances from positive samples, and minimal correlation to instances from negative samples. We propose crisp [29], fuzzy [3], and possibilistic [34] versions of the MDD. We also derive the necessary conditions to optimize the MDD and propose crisp, fuzzy, and possibilistic algorithms to cluster multiple instance data.

The organization of the rest of this proposal is as follows. In Chapter 2, related work is reviewed and its limitations are highlighted. In Chapter 3, we propose our crisp, fuzzy, and possibilistic generalizations of the DD metric, alongside the crispy, fuzzy, and possibilistic algorithms to find optimal target concepts given these metrics. In Chapter 4, we provide a few approaches to competitive MIL

classification using target concepts derived from our algorithms and, in particular embedded feature space theory. In Chapter 5, experimental results are provided, followed by a conclusion in Chapter 6.

CHAPTER 2

RELATED WORK

2.1 A Survey of MIL Approaches

Developments in Multiple Instance Learning research closely mirror the discovery of datasets that fit the MIL framework (i.e. data for which individual data samples are best represented by multiple feature vectors). One of the first documented applications that fit this criteria is handwritten digit segmentation and recognition [30, 45], for which multiple, overlapping sequences of digits within a complete sample are mapped to individual sets of features. Similarly, the authors in [14] noted that the application of drug design fits this criteria of multiple feature representation, with individual conformations for each prospective molecule being represented by a single feature. However, the initially forumlated representations and solutions provided in these and other related literature were application specific, and no formal or encompassing models were proposed.

To the best of our knowledge, Dietterich et. al [16] were the first to introduce the terms “Multiple Instance” and ”Multiple Instance Learning” in literature, as

well as the first to standardize the formulation and structure of the standard bag-instance Multiple Instance Problem. Dietterich et. al were also the first to propose a simple algorithm to solve the Multiple Instance Problem, called the Axis-parallel Rectangles (APR) algorithm. In a nutshell, the purpose of the APR algorithm is to construct a hyperrectangle in the instance feature space that encloses at least one instance from every positive sample in a training dataset, while excluding as many instances from negative data samples as is possible. They applied their APR algorithm to the drug design application by using molecular shape to predict whether a given molecule qualifies as "musky" in smell. Dietterich et. al reported that the APR algorithm significantly outperformed standard learning approaches that ignore the multiple instance problem [16]. As the APR algorithm represents a major step in the evolution of MIL design, further discussion of this approach, as well as a detailed examination of its critical limitations, will be provided in the next section.

The APR algorithm is the first proposed among many algorithms based around the Instance Space (IS) paradigm defined by Amores [1]. In the instance space paradigm, the determining factor in a model's discrimination between positive bags and negative bags rests at the level of the instance feature space (i.e. new data samples are presented to the discriminator one instance at a time, and individual confidences for each instance are amalgamated to an overall label as positive or negative). Global information and relationships between bags in the dataset (such as kernal space distances) are largely ignored in the IS paradigm.

A subsequent major step in MIL research was the formulation of the Diverse

Density (DD) approach [40]. In [40], the author defines the Diverse Density metric which combines the cumulative probability that the positive bags are correlated with a given point of interest, and the cumulative probability that the negative bags are not correlated with it. The point of interest in the instance feature space that is associated with a large DD value is dubbed the *target concept*. After finding the optimal target concept within a dataset, a classifier is then built with the target concept (point of highest diverse density) as a locus for discrimination. Whereas the APR approach ultimately uses only a single instance per bag in training the APR bounds, the DD approach allows multiple instances per bag to define the density metric. Maron utilizes the NOISY-OR metric [40], which is a probabilistic generalization of the logical OR [49], to estimate this density. In [40], the author reported that the DD classifier performs more robustly than the APR classifier for several datasets.

The basic DD methodology outlined in [40] has spawned a substantial number of variations. For example, the maxDD and QuickDD approaches introduced in [18] are claimed to be an efficient implementation to finding solutions of the same quality as the baseline DD approach. Still others have adopted the NOISY-OR metric in an effort to adapt conventional learning technqiues to the MIL domain. For example, the authors in [63] proposed the MIL Boost algorithm, which frames the weak classifier boosting strategy within a MIL context. Whereas conventional boosting (e.g. AdaBoost [19]) considers one weight per sample, the MIL Boost approach utilizes a separate bag-level weight for each bag as well as the instances within each bag. While negative bag weights are set to a constant -1 to reflect

the fact that instances in negative bags are uninformly negative, positive bag weights are reduced when one or more instances within the same bag are correctly identified by one of the weak classifiers as positive.

Another research direction has used the DD metric to identify points of the feature space suitable for feature mapping [1, 9, 10] (similar to kernal space transformation) to convert the multiple instance features to single-vector features, upon which conventional learning algorithms can be applied. Such approaches fall into the sphere of bag-space, or embedded-space multiple instance learning, both of which are described below.

A major focus in the MIL field has been to examine techniques that remap the base, multi-vector features utilized in MIL problems to a format suitable for the standard, single-instance learning (SIL) framework so that more conventional machine learning approaches can be applied to solve MIL problems. Amores defines such approaches as members of the Bag-Space (BS) paradigm [1]. Whereas IS-based approaches generate an output label for each instance's feature vector, and amalgamate the outputs into a single, net bag label, the BS based approaches effectively convert the multiple feature vectors for each bag into a single vector that defines its spatial relationship to the other bags in the dataset, and then (generally) utilizes a SIL approach to build a classifier.

The simplest of BS classifiers, which we refer to as the BS k-NN classifier [1], simply maps a prospective testing sample to all bags in the dataset using a most-likely cause estimate (MLCE) distance mapping (i.e. the distance between two bags is gauged as the minimum between any instance from the first and any in-

stance from the second), and then applies a k-NN approach to assign a label to the bag [1, 55]. A simple adjustment to the BS k-NN approach is to use an alternate distance mapping, such as the NOISY- OR metric as outlined in [1]. A slightly more sophisticated BS approach utilizes the kernel function to map bag-to-bag distances to kernel similarities, and implements a relatively standard support vector machine (SVM) for discrimination [1, 8]. In [55], the author formulated a KNN variant to solve the MIL problem using techniques garnered from citer-reference dynamics explored in library and information science; this so-called Citation-KNN approach relies on both the class of nearest neighboring bags references), as well as the neighbors of training bags (citers), to classify a prospective testing bag.

The last major, conventional paradigm outlined by Amores is the Embedded Space (ES) paradigm [1]. In a similar vein to the bag-space paradigm, ES approaches attempt to map multiple instances per bag to a single, overall feature vector. However, unlike BS approaches, ES approaches do not utilize simple distances or similarities when computing bag-to-bag relationships. Instead, they apply potentially complex mapping relationships from bags to bags and/or bags to instances to construct new, SIL-style features [1].

One approach based upon the ES paradigm is the Multiple-Instance Learning via Embedded Instance Selection (MILES) algorithm [9]. Unlike the DD or APR approaches, the MILES algorithm is built explicitly on the assumption that multiple target concepts (i.e. more than a single point or region in the feature space) define the nature of positive data samples. In deference to this assumption, the MILES approach assumes that any instance from any positive or negative bag

has the potential to correctly represent the positive or negative data. As a result, rather than mapping bag-to-bag distances in training, the MILES algorithm maps distances from a given bag to EVERY instance from EVERY positive and negative bag in the dataset. The MILES approach then utilizes a 1-norm, sparse SVM to select the best instance mappings to serve as a decision boundary for testing classification. As the presence of multiple target concepts plays a defining role in our MIL approach, we discuss the MILES algorithm and its construction in more detail in the next section.

In addition to delineation of the three different MI Space paradigms above, MIL research has addressed considerations of so-called Multiple Instance (MI) Assumptions [1] on bags and related instance distributions. The two major MI assumptions in MIL research are the standard MI assumption (SMI) [1] – which states that only a single, true positive instance is responsible for a positive bag's label, and the collective MI assumption (CMI) [1] – which states that a combination of instances per bag is responsible for at least a portion of the dataset's labels. The simpler IS approaches, such as the APR, are not suited to problems invoking the CMI. BS, ES, or more sophisticated IS approaches are required.

The authors in [11] examine the MI assumptions a step further by generating three artificial datasets to model three distinct data distributions for positive bags. The first of these, or the "Concept" dataset is generated such that one instance in each positive bag is taken from a distinctive region (i.e. no overlap between instances from positive and negative bags in the vicinity), while the rest are generated from the same distribution as instances in the negative bags. This dataset

satisfies the SMI assumption. The second dataset is referred to as the "Distribution" dataset. Positive and negative data were generated with significant overlap for this dataset, but with noticeably different distributions overall. This dataset therefore satisfies the CMI assumption, in that multiple instances within positive bags provide for better aggregate discrimination than just selecting one. The final dataset, termed the "Multi-concept dataset" is similar to the "concept" dataset in that one instance per positive bag is generated without overlap into the negative distrubtion. However, these instances are not concentrated in one or even a few regions densely populated by positive data, rendering most standard similarity metrics that we've discussed (e.g. NOISY-OR, MLCE) unsuitable for modeling relationships involving the true positive instances. Instead, the authors in [11] argue that dissimilarity metrics (e.g. Euclidean distance) are the best choice for discriminating positive instances in the multi-concept data, as they will tend to have a high relative distance to other instances in the dataset. We examine the utility of dissimilarity metrics more in Chapter 4, when we introduce the idea of negative target concepts.

More recent developments in MIL research include the incorporation of deep learning and attention mechanisms. In [58], the authors propose Deep Multiple Instance Learning (DMIL). DMIL adapts a standard Deep Neural Network to include assumptions and characteristics of MIL problems in the learning process. For example, the backpropagation step of the architecture used by DMIL factors the assumption that only a single instance in a positive bag need be positive into network feedback, while instances from negative bags are treated in standard

fashion. The authors of [24] reframe the MIL problem of instance ambiguity from the perspective of a Bernoulli distribution. The architecture proposed by [24] differs from that used in [58] in that an attention mechanism influences the weight individual instances within a bag have in determining its label.

Another conventional machine learning approach that has been adapted to the MIL setting recently includes randomized trees [32] In [32], the authors propose bag-level random trees that rely on only a single tuned parameter indicating instance-to-bag repsonse ratio for good performance in bag-level classification on datasets fitting different applications and with different characteristics.

In [6], the authors provide a summary of MIL problem characteristics, evaluate the distribution of real-world applications to which MIL approaches have been applied, and critique assumptions and steps taken in contemporary MIL research. In particular, the authors in [6] argue that MIL characteristics are often categorized using multiple names (e.g. scenarios in which positive bags have a small number of associated positive instances can be characterized as "sparse bag" problems or "low witness rate" problems), and that the commonly utilized Benchmark datasets (i.e. the MUSK and COREL-based datasets) may not be adequate for meaningful MIL performance analysis for most applications or approaches.

MIL research has seen application to a variety of real-world applications in recent years. In [35], the DMIL approach is adapted to perform Panchromatic (PAN) and multispectral (MS) imagery-based classification. The authors of [35] argue that their approach outperformed the accuracy of a standard deep learning network on a set of four airborne PAN and MS datasets.

DMIL is also utilized by the authors of [25], who seek to improve the ability of automated systems to predict the presence of malignant prostate cancer through analysis of prostate needle biopsies. A DMIL network is trained using nuclear morphologic images matching the presence or absence of the metastatic form of the disease, which in turn was found to have a higher success rate in predicting the presence of the disease in test samples than non-MIL based approaches.

Deep learning and multiple instance learning were also utilized by the authors in to approve cancer detection rate of imaging acquired through Digital breast tomosynthesis (DBT). Unlike DMIL-based approaches, the authors in [61] apply deep learning and MIL in separate steps. The former are used to locate a set of patterns in the DBT that are potentially associated with breast cancer (bags), while the latter is used as a mechanism to distinguish between false positive and true positive patterns (instances).

2.2 Overview of Relevant Non-Clustering MIL Approaches

In the following, we explore three of the aforementioned algorithms – specifically, the APR algorithm, DD algorithm, and MILES algorithm in more detail, as they are highly relevant to our research and proposed approaches.

2.2.1 The Axis-Parallel Rectangles Algorithm

The *Axis-parallel Rectangles (APR)* algorithm [16] was the first approach that formally addressed the multiple instance problem. The general method of learn-

ing constructs a set of axis-parallel rectangles in the feature space that correlate strongly with the features of positive training bags, in an effort to provide the best possible discrimination between prospective testing samples. This algorithm is best suited for tasks where a single instance from a positive bag is responsible for its label. It has the advantage of fairly efficient training and testing processes. However, the imposed hard boundaries leave the algorithm fairly sensitive to noise, and unable to cope with tasks for which multiple instances in a positive bag play a role in its classification. In the following, we highlight the training and testing processes of this approach.

- APR Model Training

The primary goal in training an APR-based classifier is to construct a set of bounds in feature space that span at least a single instance from every positive bag in the training set with as few instances from negative bags as possible. Three main approaches were proposed to achieve this goal. The first one, called *standard* design, constructs the smallest (in terms of total bound size) APR encasing all instances from positive samples, and hence ignores the multiple instance problem entirely. This approach is used as a basis for performance comparison. The second approach, referred to as the *outside-in*, creates a set of APR identical to that of the standard APR, and then shrinks the APR to exclude as many negative instances as possible while maintaining at least one instance from every positive bag. The third approach, called the *inside-out*, begins with a positive "seed" instance and then grows a set of APR to include additional instances from unique positive bags while including as few instances from negative bags as is possible. The authors in

[16] reported that the inside-out approach yields the best performance.

- APR Model Testing

Testing a new sample in the vanilla APR framework is straightforward: if the features of any individual instance in the bag are within the constructed APR, the bag is classified as positive. Otherwise, the bag is classified as negative.

The APR approach was initially designed with the task of drug design in mind [16]. This application consists of predicting bonding activity of molecules with a specific protein. Within the MIL framework, each molecule (bag) is represented by a number of conformations (instances), and only a single binding conformation (positive instance) is required for an active molecule (positive bag). The rigid nature of the APR approach is seemingly well-suited to this problem, as the data are typically noise-free and only a single positive instance is required to predict a bag's label as positive. Many other MIL datasets do not fit these criteria [1].

2.2.2 The Diverse Density Algorithm

Rather than constructing a set of rigid rules encasing positive instances, the *Diverse Density (DD)* algorithm [40] model focuses on finding a region in the feature space with high positive density (many instances from unique positive bags) and low negative density (few instances from unique negative bags). Before addressing the specifics of training and testing a DD classifier, we first examine two concepts critical to its construction: the *target concept*, and the NOISY-OR metric.

A *target concept* t is defined as a region in the instance feature space that has a strong correlation with instances from positive bags and weak correlation

with instances from negative bags. The target concept may theoretically be a set of rules defining an appropriate region in the feature space, as is the case in the axis-parallel rectangle approach. Alternatively, a target concept may be a single point in the feature space associated with a small distance to many instances from unique positive bags and large distance to instances from unique negative bags, as is the case with the diverse density approach. Early research in MIL assumed the presence of a single target concept [40, 5], while later approaches assume the potential need for multiple target concepts [9, 5].

The second concept, NOISY-OR, was first used in SIL to model the relationship of disjunctive concepts in Bayesian networking [49]. Given potential events $X_1 \dots X_n$ and associated probabilities $p(X_1) \dots p(X_n)$, the NOISY-OR probability of at least one of these events occuring is defined as

$$NOISYOR(X_1 \dots X_n) = 1 - \prod_{j=1}^{n}(1 - p(X_j)) \tag{2.2.1}$$

.

For the MIL framework, we assume that a bag B_n is represented by a set of instances $\{b_{n1} \dots b_{nI}\}$. We assume that we have a predetermined target concept t in the same feature space as every b_{ni}, and that each b_{ni} is assigned posterior probability $p(b_{ni}|t)$. The collection of such memberships for the instances in bag B_n can then be defined by disjoint probabilities $p(b_{n1}|t) \dots p(b_{nI}|t)$. If we view each $p(b_{ni}|t)$ as the probability of an individual event X_i occurring, it becomes apparent that we may apply (2.2.1) to model the probability that any of a bag's

individual instances meets the membership criteria for t using

$$NOISYOR(B_n, t) = 1 - \prod_{i=1}^{I}(1 - p(b_{ni}|t)) \tag{2.2.2}$$

Equivalently, the NOISY-OR could be defined in terms of similarity between bag b_{ni} and target concept t such that

$$NOISYOR_{sim}(B_n, t) = 1 - \prod_{i=1}^{I}(1 - sim(b_{ni}, t)) \tag{2.2.3}$$

assuming that $sim(b_{ni}, t) \in [0, 1]$.

The NOISY-OR metric in (2.2.2) is most commonly used as a probabilistic alternative to the *most likely cause estimate (MLCE)* or *MLC* standard [40]. The MLCE considers only the closest instance within a bag to the target concept. Given target concept t and bag B with n instances, the most-likely cause estimate is defined as

$$MLCE_{dist}(B_n, t) = \min_{i=1..I} d(b_{ni}, t) \tag{2.2.4}$$

In (2.2.4), d represents the Euclidean distance between instance b_{ni} and target concept t. We note that (2.2.4) is based on a distance metric, whereas the NOISY-OR metrics in (2.2.2) and (2.2.3) are built around probability or similarity measures. A comparable MLCE measure may be built around similarity using

$$MLCE_{sim}(B_n, t) = \max_{i=1..I} sim(b_{ni}, t) \tag{2.2.5}$$

The NOISY-OR metric is a preferable alternative to the The MLCE metric in

MIL problems where multiple instances per bag may play a role in determining its label, or in cases where a functional derivative is required (the max function is not differentiable at zero).

- DD Model Training

Training a DD classifier consists of finding a point in the feature space with the highest diverse density metric. Let t be a potential target concept, $\{B_1^+ ... B_{N_{pos}}^+\}$ be the set of N_{pos} bags with positive labels, and $\{B_1^- ... B_{N_{neg}}^-\}$ be the set of N_{neg} bags with negative labels. The diverse density of t is defined as

$$DD(t|B) = \prod_{n=1}^{N_{pos}} Pr(x = t|B_n^+) \prod_{n=1}^{N_{neg}} Pr(x = t|B_n^-) \tag{2.2.6}$$

Equation (2.2.6) can be broken into two components: the positive density, $\prod_{n=1}^{N_{pos}} Pr(x = t|B_n^+)$, and the negative density, $\prod_{n=1}^{N_{neg}} Pr(x = t|B_n^-)$. For a given positive bag $B_n \in B^+$with I instances $b_{n1}..b_{nI}$, the NOISY-OR metric in (2.2.2) is used to compute the n_{th} component of the positive density as:

$$Pr(x = t|B_n^+) = 1 - \prod_{i=1}^{I}(1 - sim(b_{ni}^+, t)) \tag{2.2.7}$$

Similarly, given a negative bag $B_n \in B^-$, (2.2.2) is used to compute the n_{th} component of the negative density as

$$Pr(x = t|B_n^-) = \prod_{i=1}^{I}(1 - sim(b_{ni}^-, t)) \tag{2.2.8}$$

We note that the partial diverse density is proportional to the *complement* of the

probability that all instances in a positive bag are dissimilar to target concept t, whereas the partial negative density is directly proportional to the probability that all instances in a negative bag are dissimilar to t. Hence, the diverse density is driven *up* by the presence of positive instances in its vicinity, and driven *down* by the presence of negative instances.

In the DD framework, the instance-level similarity function used in (2.2.7) and (2.2.8) is typically defined as

$$s(b_{ni}, t) = e^{-\|b_{ni}-t\|^2} \tag{2.2.9}$$

Assuming that instance b_{ni} corresponds to a L-dimensional vector, then $\|b_{ni} - t\|^2$ is simply a weighted Euclidean distance, i.e..

$$\|b_{ni} - t\|^2 = \sum_{l=1}^{L} s_l^2 (b_{nil} - c_l)^2 \tag{2.2.10}$$

In 2.2.10, s_l is the lth feature of a scaling vector of fixed a-priori or optimized as discussed below, and c_l is the lth feature for the centroid corresponding to target concept t.

Equations (2.2.6)-(2.2.10) may be used to compute the diverse density for a given dataset and potential concept t. However, the main goal of training a DD model is to find the optimal target concept, t_{opt}, that maximizes the DD in (2.2.6). Due to the precision difficulties in computing multiple products of small probabilities, the optimization problem for finding the DD is often reframed from the maximization of the quantity in 2.2.6 to minimizing its negative log measure,

or -log(DD). In [40], the author proposed a gradient descent approach to finding the optimal target concept t_{opt}, and the optimal scaling vector s_{opt}. This approach is outlined in algorithm 1.

Algorithm 1 A Gradient Descent Approach to Optimizing Diverse Density

Inputs: $\mathcal{B}^+$ and $\mathcal{B}^-$: the sets of + and - bags.
Outputs: c_{opt} and s_{opt}: Centroid and scales for an optimized target concept.

Select an initial target concept centroid c as a random positive instance b_{ni}^+ from a random positive bag B_n^+.
Initialize scaling vector s to be all ones.
repeat
Using (2.2.7)-(2.2.10), compute the negative log-likelihood gradients $\nabla F(c) = \frac{\partial(-log\widehat{DD}(t)}{\partial c}$ and $\nabla F(s) = \frac{\partial(-log\widehat{DD}(t)}{\partial s}$
Update c using $c^{(new)} = c^{(old)} + \alpha \nabla F(c)$
Update s using $s^{(new)} = s^{(old)} + \alpha \nabla F(s)$
until Some convergence criteria are met (e.g. $\|t^{new} - t^{old}\|$ less than some threshold).
return $c_{opt} = c$ and $s_{opt} = s$

Algorithm 1 is iterative and can converge to local minima. To alleviate this drawback, the author in [40] recommends training a DD model using multiple starting points (different instances from different positive bags) and selecting the target concept with the highest DD.

- DD Model Testing

Classifying new samples using a trained DD model is fairly straightforward. First, (2.2.4) is used to compute the MLCE distance to the target concept. If this value is less than trained threshold θ, the bag is classified as positive. Otherwise it is classified as negative.

2.2.3 Single vs Multiple Target Concepts

Figure 2.1: Sample images that illustrate the need for multiple target concepts to describe "sky."

The APR and DD approaches operate on the underlying principle that a single target concept, or region, best correlates with the criteria for a positive instance. While this assumption may be appropriate for applications in which positive instances lay within an unimodal distribution, such as the drug design application outlined in [16], it is unsuitable for applications with large within-class variations [9]. As a simple illustration, we consider figure 2.1 and the application of automated image annotation outlined earlier. Figure 2.1 depicts a small subset of images which fit the annotation "sky,", but have relatively little in common in terms of color or texture structue. Assuming color or texture features are being used, any MIL classifier constrained to single target concept behavior cannot represent the diversity of the concept "sky." Similarly, the aforementioned application of automated buried explosive object detection faces the same complication – specifically, the presence of disparate BEO types and disparate environments [20] points to the need for more than a single target concept. An example of the diversity of different target types is shown in figure (2.2), where the first image depicts

a wide, full signature, the second depicts a wide and hollow signature, while the third depicts a narrow BEO signature. With these issues in mind, it would be beneficial to generalize MIL approaches so that they can model positive bags with multiple target concepts. One of these approaches is the MILES algorithm, which is outlined below.

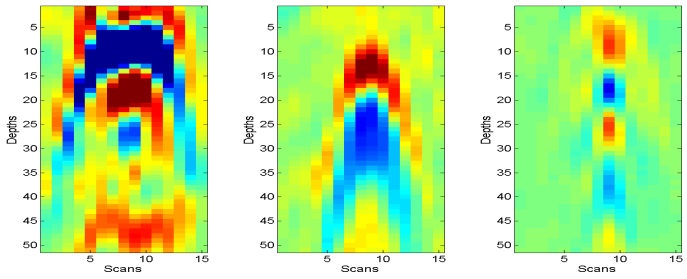


Figure 2.2: Three BEO Signatures with Distinct Features

2.2.4 The MILES Algorithm

The *Multiple Instance Learning via Embedded Structures (MILES)* algorithm [9] differs from the previous two approaches in that it relies on feature mapping to transform the MIL problem into a conventional SIL one. Furthermore, unlike the prior two approaches, multiple target concepts are integral to the training and testing processes. While training in the MILES model is computationally expensive, relative to the APR and DD algorithm [1], classification in the MILES model is accomplished through simple logistic regression, and hence the model is suitable for online testing. In [9], the authors argue that the MILES algorithm is, by nature of its sparse feature selector, particularly robust to noise. The main steps of MILES are described in the following subsections.

- Feature Mapping

The purpose of feature mapping in the MILES framework is to remap the multiple feature vectors of a given bag B_n, which effectively form a 2-dimensional matrix, to a single-dimensional feature vector ν_i based on the proximity of its individual instances $\{b_{n1}..b_{nI}\}$ to a predetermined set of p target concepts $C = \{t_1..t_p\}$. This is accomplished with a simple mapping function $f_{map}(B_n, C) = \nu_n$. The features corresponding to the instances of B_n and individual target concepts of C are assumed to be of the same size. Furthermore, every individual element of ν_n is constrained to be in the interval $[0, 1]$.

The standard feature mapping function in the MILES algorithm is a variant on the most-likely cause estimate (MLCE) similarity in (2.2.5). Formally, given bag B_n and a set of target concepts $C = \{t_1 \dots t_p\}$, ν_n is defined as

$$\nu_n = f_{map}(B_n, C) = [sim_{MLCE}(t_1, B_n), sim_{MLCE}(t_2, B_n), ...sim_{MLCE}(t_p, B_n)] \tag{2.2.11}$$

where

$$sim_{MLCE}(t_k, B_n) = \max_{1 \le j \le n} exp(\left(\frac{-\|b - t_k\|^2}{\sigma^2} \right)),\ for\ 1 \le k \le p \tag{2.2.12}$$

In (2.2.11, σ) is a predefined scaling factor. We note here the similarity between (2.2.12) and (2.2.5).

If the mapping process models the relationship between bags and instances in the dataset properly, the MIL problem of ambiguity is addressed, and each feature

vector has a one-to-one correspondence to each training label. Most importantly, standard SIL approaches that are robust and can learn relevant features may be applied to the samples with remapped features.

Despite the advantages conferred, feature mapping in the MIL context has several requirements, and at least one major drawback common to approaches in the BS or ES paradigm. More specifically, MILES requires consolidation of multiple distance or similarity metrics between the instances in a bag to a set of target concepts, potentially losing information in the process of the mapping transformation.

- MILES Model Training

Assume that we have a set of N bags. The MILES algorithm starts by mapping every bag B_n, $1 \leq n \leq N$, of the dataset using a set of target concepts C, as outlined in (2.2.11) and (2.2.12). The authors in [9] assume that *every* instance within a positive bag is a potential target concept. Hence, C is defined as $C = \{b_{11}^+, b_{12}^+, .., b_{1m_1}^+, b_{21}^+, .., b_{2m_2}^+, .., b_{Nm_N}^+\}$, where b_{xy}^+ is the yth instance of bag B_x^+ and m_x is the number of instances in bag B_x^+. As a consequence, this approach may not scale well when the training data includes a very large number of bags with many instances. Each remapped feature $\nu_i = f_{map}(B_n, C)$ inherits the bag label of its bag B_n. The labeled v_n are then used in training to learn a weight vector w and a bias parameter b corresponding to a sparse linear classifier of the form

$$y = sign(wv + b). \tag{2.2.13}$$

Since very few instances make for informative target concepts [9], stringent feature selection is a requirement for this approach to work effectively. The MILES algorithm accomplishes this through the use of a *1-norm support vector machine (SVM)* formulation [9]. Following a similar approach to the one in [12], MILES utilizes a soft-margin penalty in the optimization criteria and adopts a 1-norm regularization function.

The major steps of training the MILES SVM are as follows:

1. Construct set $C = \{t_1 \dots t_p\}$ consisting of all positive instances from all positive bags.

2. Use C, (2.2.11), and (2.2.12) to derive embedded feature vectors $v_1..v_m$.

3. Utilize linear programming to optimize a 1-norm regularization function and derive optimal weights w_{opt} and optimal bias b_{opt} for the linear classifier in)(2.2.13).

- MILES Model Testing

Testing a new sample B_t in the MILES framework is efficient and simple. Features are remapped as in training step 2, but only those indices corresponding to non-zero weights in C are required for testing purposes (reducing computation time needed with models trained for larger datasets). Lastly, the logistic regression in (2.2.13) is applied using the remapped feature vector ν_{opt}, weight vector w_{opt} and trained bias b_{opt}.

While using all *all* instances from *all* bags as a potential pool for effective class discrimination grants the MILES algorithm advantages from the standpoint

of using any instance as a potential target concept, it confers several drawbacks. Even though MILES uses a robust classifier with feature selection, the amount of noise can be too large to ignore. For instance, if we assume that each bag has 10 instances and only one of the instances is positive, 90% of the potential target concepts used for mapping are noisy and irrelevant. This issue is compounded with the possible presence of standard noise in the data. In addition, using all instances of all bags can result in vectors with a very large number of dimensions. For instance, if the training data has 10000 positive bags and each bag has 10 instances per bag, the mapped vectors would have 100000 dimensions. This may be too large even for a sparse SVM classifier. Moreover, since the dimensionality of the mapped features is larger than the number of training samples, the applied algorithm faces major potential to overfit the data. Finally, even if the classifier is robust and can identify relevant features, MILES can face scaling issues even for data sets with medium sizes.

As an alternative, searching the instance space for multiple target concepts simultaneously can be related to data clustering, in that the goal of each is to locate a a set of boundaries or regions in the feature space that consistently group together like data. In the following section, we outline the concept of data clustering and describe a few approaches that are common for SIL data.

2.2.5 Clustering Algorithms for Single Instance Learning Data

A staple approach in machine learning to organizing data is cluster analysis. Broadly speaking, the role of clustering analysis (henceforth referred to simply

as clustering) is to partition a larger set of data into multiple smaller subsets of data, referred to as *clusters*, such that samples assigned to the same cluster are as similar as possible and samples assigned to different clusters are as dissimilar as possible [54, 17, 4]. Consequently, cluster quality is typically defined in terms of similarity or distance metrics, and a principle objective in clustering is to find a set of clusters with maximal intra-cluster similarities (i.e. each point within a dataset is as similar to the other points in its cluster as possible) and/or minimal inter-cluster similarities (i.e. after clustering a representative from each cluster, or centroid, is used to summarize all data samples assigned to the cluster).

In the following, we examine three algorithms that are commonly used to cluster single instance data. These are the Crisp K-means [38], the Fuzzy C-means [3], and the Possibilistic C-means [34].

- Crisp K-means Algorithm

The Crisp K-means algorithm assumes that the data has K clusters and each cluster, k, is represented by its centroid, c_k [38]. It assigns each sample to one and only one of the clusters. Given N samples $\mathcal{X} = \{x_1..x_N\}$, the K-means algorithm seeks to minimize the objective function

$$J = \sum_{k=1}^{K} \sum_{x_n \in s_k} ||x_n - c_k||^2 \tag{2.2.14}$$

where $||x_n - c_k||^2$ is a distance function (typically Euclidean) between sample x_n and cluster k centroid for c_k, and S_k is the set of samples from $\mathcal{X}$ assigned to

cluster c_k.

Finding an optimal solution to the objective function in (2.2.14) is NP-hard, and heuristics are required to obtain a local minimum. Typically, the objective function in (2.2.14) is minimized using the LLoyd's algorithm [36]. This is an iterative, two-step assignment/update approach to finding an optimal partition. Initially, all data samples are assigned randomly to clusters. Then, the K-means algorithm alternates between two main steps. First, the cluster centroids are updated using

$$c_k = \frac{\sum_{x_n \in S_k} x_n}{||S_k||} \tag{2.2.15}$$

Second, data samples are reassigned to their closest clusters using

$$k = arg \min_{k=1..K} ||(x_n - c_k||^2 \tag{2.2.16}$$

for $n = 1..N$. The above two steps are repeated until convergence to a local optimum, which is guaranteed [36]. The K-means method is summarized in Algorithm 2. In general, the K-means algorithm is relatively efficient in comparison to other clustering algorithms [4]. It is expected to provide a good partition of the data when the clusters' boundaries do not overlap significantly. If this is not the case, fuzzy clustering may be the method of choice.

Crisp clustering is preferable only when there is expected to be little overlap between data classes, and is not suitable for noisy data or in cases that data samples may belong to more than a single class [3]. A more flexible means of

Algorithm 2 The K-means Algorithm

Inputs: $\mathcal{X}$: the complete set of data.
Outputs: $\mathcal{C}$: Centroids $c_1..c_K$ of the K clusters.
$\mathcal{S}$: Membership sets $s_1..s_K$ of the data samples assigned to clusters $1..K$.

Initialize C.
repeat
Find partition set S using (2.2.16).
Update C using (2.2.15)
until Centers do not change significatively or number of iterations is exceeded.
return $\mathcal{C}$ and $\mathcal{S}$

finding clusters, fuzzy clustering [3] is built upon the natural union between fuzzy set theory proposed in [62] and clustering theory. This approach is described below.

- The Fuzzy C-means Clustering (FCM) Algorithm

Under the fuzzy model of clustering, a sample need not belong to a single cluster. Instead, each data sample x_n is assigned a set of memberships $u_{n1} \cdots u_{nK}$, where u_{nk} represents the sample's membership assignment to cluster c_k. Given dataset X, the FCM algorithm seeks to minimize

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N} (u_{nk})^m ||x_n - c_k||^2 \tag{2.2.17}$$

, subject to the constraints

$$u_{nk} \in [0, 1], \quad \text{and} \quad \sum_{k=1}^{K} u_{nk} = 1. \tag{2.2.18}$$

where $m \in [1, \infty)$ is a fuzzifier parameter determining the fuzziness of the mem-

bership function. Minimization of (2.2.17) with respect to u_{nk}, subject to the constraints in (2.2.18), yields the following upate equation for u_{nk}:

$$u_{nk} = \frac{1}{\sum_{q=1}^{Q} (\frac{||x_n - c_k||^2}{||x_n - c_q||^2})^{\frac{1}{m-1}}} \tag{2.2.19}$$

Minimization of (2.2.17) with respect to c_k yields the following update equation for the clusters' centroids:

$$c_k = \frac{\sum_{n=1}^{N} (u_{nk})^m x_n}{\sum_{n=1}^{N} (u_{kn})^m} \tag{2.2.20}$$

The resulting FCM method is outlined in Algorithm 3.

Algorithm 3 The FCM Algorithm

Inputs: $\mathcal{X}$: the complete set of data.
m: A fuzzifier parameter **Outputs**: $\mathcal{C}$: Centroids $c_1..c_K$ of the K clusters.
$\mathcal{U}$: Membership matrix defining membership of each sample in X in each of the K clu

Initialize C.
repeat
Assign membership u_{ij} for each sample x_i and cluster c_j using (2.2.19)
Update cluster centroids $c_1..c_J$ using (2.2.20).
until Centers do not change significatively or number of iterations is exceeded.
return $\mathcal{C}$ and $\mathcal{U}$

A potential draw-back of fuzzy clustering is that, for a given data sample, its membership across all K clusters must sum to 1. In particular, it does not distinguish between a point that is equally close to multiple clusters from a point that is equally close to multiple clusters from a point that equally far away. Consequently,

the FCM cannot identify noise points and its resulting partition is often not robust. To alleviate this problem, the authors in [34, 33] proposed the possibilistic c-means (PCM) algorithm. This approach is outlined below.

- The Possibilistic C-means Clustering

The PCM approach [34] relaxes the FCM membership constraints that the membership of each point in all clusters must sum to 1. Instead, they introduced a scale parameter η_k for each cluster, k, and use absolute membership functions. The PCM algorithm minimizes the following objective function:

$$J(\mathcal{C},\mathcal{U};\mathcal{X}) = \sum_{j=1}^{J}\sum_{i=1}^{I}(u_{ij})^m||x_i - c_j||^2 + \sum_{j=1}^{J}\eta_j\sum_{i=1}^{I}(1-u_{ij})^m \quad (2.2.21)$$

, subject to the constraint,

$$u_{ij} \in [0,1] \quad (2.2.22)$$

Minimizing (2.2.21) with respect to u_{nk} leads to the following equation:

$$u_{ij} = \frac{1}{1+(\frac{||x_i-c_j||^2}{\eta_j})^{\frac{1}{m-1}}} \quad (2.2.23)$$

Similarly, minimization of (2.2.21) with respect to C leads to the following update for the centroids:

$$c_j = \frac{\sum_{i=1}^{I}(u_{ij})^m x_i}{\sum_{i=1}^{I}(u_{ij})^m} \tag{2.2.24}$$

The PCM algorithm requires the specification of the cluster-dependent scaling parameter, η. In [34], the authors recommend updating the parameter in every iteration based on the cluster's statistics. The PCM algorithm in summarized in algorithm 4.

Algorithm 4 The PCM Algorithm

Inputs: $\mathcal{X}$: the complete set of data.
m: A fuzzifier parameter **Outputs**: $\mathcal{C}$: Centroids $c_1..c_K$ of the K clusters.
$\mathcal{U}$: Membership matrix defining membership of each sample in X in each of the K clu
$\eta_1 \cdots \eta_K$: cluster-dependent scaling vectors for each of the K clusters.

Initialize C.
Initialize $\eta_1 \cdots \eta_K$.
repeat
  Assign membership u_{ij} for each sample x_i and cluster c_j using (2.2.23)
  Update cluster centroids $c_1..c_J$ using (2.2.24).
  If desired, update $\eta_1 \cdots \eta_K$ as outlined in [34].
until Centers do not change significatively or number of iterations is exceeded.
return $\mathcal{C}$, $\mathcal{U}$, and $\eta_1..\eta_K$.

2.3 Other MIL Clustering and Multiple Target Concept Approaches

Several other approaches to MIL utilizing multiple target concepts or clustering have been proposed. One category of such approaches looks to formulate target concepts and their relationships to bags using probabilistic models. For instance,

in [5] the authors used the grain-and-germ model and random set theory to model bags and a set-based target concept structure. In juxtaposition to the DD model, germs effectively represent target concept centroids and grains represent target concept scales. The authors of [5] argue that their approach can robustly predict a set of target concepts where the DD approach fails (i.e. specifically when multiple target concepts characterize the data). In [27], individual instances across the dataset are viewed as naturally generated data clusters based on multiple Gaussian mixture models with Dirichlet process priors on the weights. Instances in positive bags are viewed as members of either the "witness" (true positive) or "non-witness" (negative) class. The goal of the approach utilized in [27] is to maximize the margin between distributions associated with witness and non-witness/negative instances.

Other MI clustering works geared around the discovery of maximum margins have been proposed. In [64], the authors propose several variations of the Maximum Margin Multiple-Instace Clustering (M^3IC) approach. The principle goals of M^3IC algorithms are to identify cluster partitions that that maximize a bag margin metric across the bags. This bag margin metric for a given bag is defined by the dissimilarity of its most discriminative instance (i.e. the one with maximum value) to instances in bags from other data classes. The authors in [64] ultimately utilize the Constrained Concave-Convex Procedure (CCCP) and Cutting Plane (CP) methods to efficiently find margins in the feature space that provide good cluster separation.

The authors of [59] propose the MCIL algorithm, which utilizes principles of

the the MIL Boost algorithm [63] in a clustering framework. More specifically, a bag is classed as positive if at least one of its instances is assigned membership to one of K positive clusters, each of which has a corresponding weight function. (In comparison, the standard MIL Boost algorithm utilizes only a single weak classifier for this weight function). The boosting process is modified to include reweighting of these cluster memberships as a step prior to computing classifier error. When a positive instance is assigned membership to a cluster in any given iteration, that instance and bag then receive a reduced weight in assignment to membership in other clusters, encouraging a level of diversity in the resulting clusters. The authors in [59] report strong results in applying MCIL to accurately classify and automatically annotate colon cancer histopathology and cytology datasets.

We note that while the above approaches address the need for multiple concepts and/or clusters within the MIL domain, none attempt to incorporate explicit soft centroid-based cluster labeling to samples (i.e. directly assigning fuzzy or possibilistic labels to data samples based on cumulative bag characteristics.) In the next chapter we address the potential advantages of performing the latter, as well as introduce our contribution to MIL research in the form of a Clustering-based, multiple target concept approach to optimizing the diverse density metric.

CHAPTER 3

A CLUSTERING-BASED MULTIPLE TARGET CONCEPT APPROACH TO DIVERSE DENSITY

In section 2.2.2 we reviewed the Diverse Density (DD) algorithm. The objective of the DD approach is to locate a region in the feature space that possesses both the greatest collective proximity to positive bags in the data, as well as the least collective proximity to negative bags. The standard DD metric is optimized with respect to only a single, optimal point in the instance feature space, called the target concept. The use of a single target concept may be adequate for many applications of MIL research, such as the drug design problem [16]. However, there exist many other MIL problems with large intra-class variations where a single target concept is not sufficient to characterize all positive bags. In the rest of this chapter, we first illustrate the need to learn multiple target concepts within the MIL framework. Then, we show that existing solutions that perform the DD

with multiple different initializations cannot provide a reliable solution. Finally, we introduce our generalization of the DD approach that utilizes clustering theory to learn multiple target concepts simultaneously.

3.1 The Need for Multiple Representatives in Single Instance Learning

The term Single Instance Learning (SIL), in the context of MIL research, is used to refer to conventional machine learning approaches, where there is no ambiguity in sample labels (i.e. each object is characterized by a single instance). As with MIL problems, a principal task in the SIL context is to construct classifiers that can predict a sample's label based on its underlying features. The most basic approach to these problems uses training data to learn a single, unimodal distribution across the features for one class, and (possibly) another distribution for the second class. Then, a simple rule (e.g. Bayes) can be used to classify new data according to the way it fits the distribution(s). However, many real world data cannot be properly modeled with a simple, unimodal distribution. A well studied real-world example of this issue can be seen in statistical research correlating age and the incidence of Hodgkin's Lymphoma in populations. A sample of Hodgkins Lymphoma data from the UK is presented in Figure 3.1. Whereas some illnesses have a simple, monotonic correlation with age (generally increasing or decreasing with age), Hodgkin's Lymphoma is documented to have a distinctly bimodal distribution with respect to age, with peaks in the 20-25 and 75-80 year age ranges.

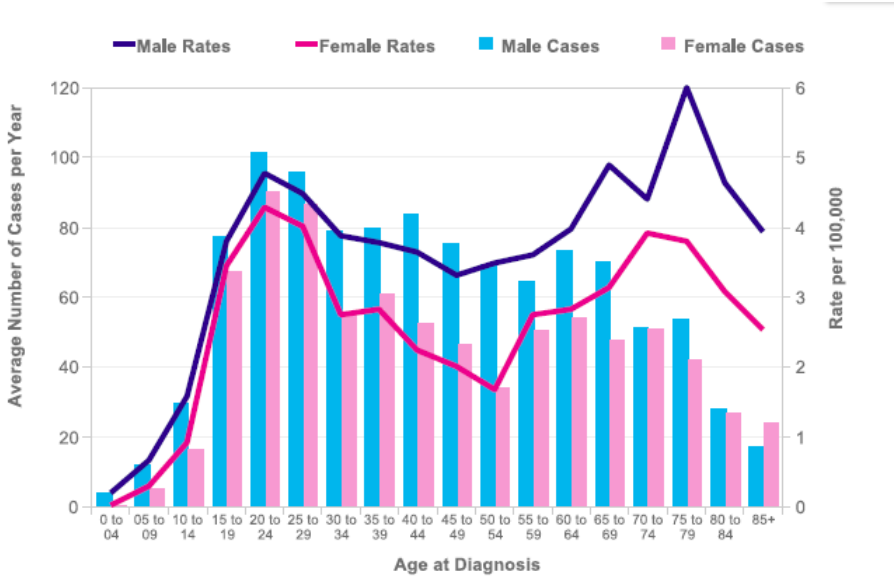


Figure 3.1: Real world bi-modal data: The Hodgkin's Lymphoma age distribution is best represented with at least two modes, one at ˜23, and one at ˜75.

An attempt to build a discriminator based on a unimodal interpretation of the age-cancer link will, at best, capture only one of the two nodes of incidence. Worse yet, if the model selects from along the continuum between two modes (e.g. the global mean of data), it is possible that an entirely false distribution will be selected to represent the data.

The prevelance of complicated distributions in real world data has led to the convention that virtually any modern approach in machine learning assumes the presence of multiple distributions per class among data. A common approach to learn such distributions is the Expectation Maximization (EM) algorithm [15]. The EM is an iterative approach that alternates between expectation and maximization steps to learn multiple components that can approximate an arbitrary probability density function. More sophisticated approaches to characterize such distributions in classification include support vector machine (SVM) [12] model-

ing, in which representatives of the dataset (called support vectors) are selected to produce a hyperplane maximizing feature space separation between the classes.

Non-parametric approaches to learning are yet another approach utilized when the distribution of the training data cannot be approximated by one (or few) known density functions. For instance, the K-nearest neighbor (KNN) classifier [13] assigns a class label to a sample based on an aggregate vote by its neighbors' labels. In other words, it assumes that the label of the test sample can be accurately predicted by the labels of the training samples in its vicinity, bypassing the need to explicitly learn distributions.

3.2 The Need For Multiple Concepts in Multiple Instance Learning

A common objective in MIL methodology is to locate points or regions in feature space heavily associated with instances from the set of positive bags, B^+. These points or regions are not technically distributions, but are instead referred to as target concepts. Nonetheless, just as real world SIL problems prompt the need for multiple distributions to model classes' probability density, real world problems in the MIL context prompt the need to consider approaches that utilize multiple target concepts.

As an illustrative example, we revisit the application of buried explosive object detection using ground-penetrating radar (GPR), referred to in shorthand as the BEO problem. Data samples collected from a GPR sensor are given a label of

"positive" when an explosive object is buried under the region of interest, and "negative" when no explosive object is present. Each sample has a corresponding GPR data cube, which corresponds to levels of intensity returned by the sensor during data collection. Cross-sections of the data cube are typically taken along the down-track and cross-track orientations, and a feature extractor is applied to quantitize the corresponding 2-dimensional images. The resulting features are then processed by an algorithm trained to distinguish between the positive and negative samples. For our purposes, we consider the edge-histogram algorithm (EHD) [57], which is an efficient and effective algorithm to BEO detection that has been implemented in a real-time system [20]. As depicted in figure 3.2, the portion of GPR data associated with the BEO signature is often small (10% or less) relative to the collected data cube and cross-sections. Furthermore, the depth at which a particular object is buried within the sample is entirely unknown, and can range from the top of the data to the bottom. To train the EHD classifier, the actual depth position is extracted manually for the BEOs; for clutter objects, multiple signatures from arbitrary depth positions are used. For testing, the EHD partitions the signature into 15 overlapping depth bins and tests each bin independently. The final confidence value is obtained by averaging the top 3 depth bin confidences.

The manual depth extraction used for the EHD training can be manageable for small data collections. However, this process has become impractical as training data have expanded considerably and are collected at a much faster pace.

If only a single object type with consistent properties in a uniform background is present in the dataset, it may be sufficient to define the problem in the context

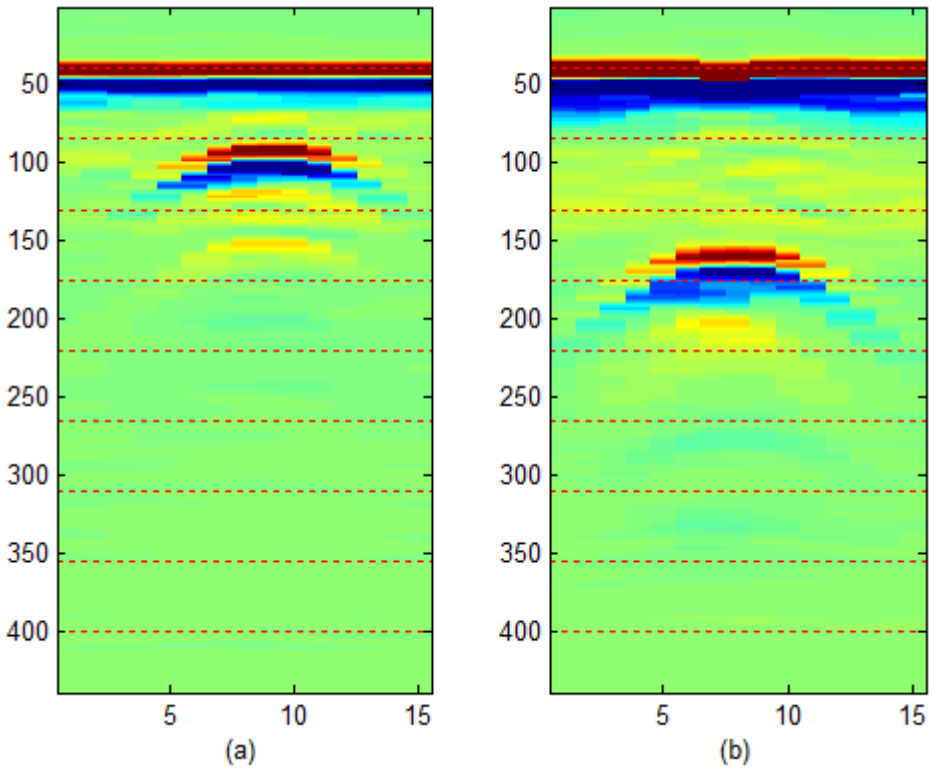


Figure 3.2: BEO Detection Example 1: Two collected positive data samples of BEOs buried at different depths. (a) BEO signature located in the 3rd depth bin. (b) BEO signature located between the 4th and 5th depth bins.

of a MIL model with a single target concept. In this case, EHD features, extracted at multiple depths, are grouped in a bag. Then, standard MIL approaches (e.g. Diverse Density) can be used to derive the appropriate single target concept that represents the positive data. However, real-world BEO data is far more complicated. The variety of explosive objects that are found beneath the surface is immense, and various factors such as weather condition and soil aridity, to name a few, can affect the data gathered by a sensor. Figure 3.3 depicts the issue with regards to two distinct object types; the first two data samples depicted are of one BEO type, while the second two are of another. Even if the "true" depth window can be identified for every data sample, it is highly likely that features extracted from the first two data samples will be very different from those extracted from the last two. This discrepancy in features implies that a well designed multi-target concept approach is more appropriate than a single target concept approach when addressing data with large intra-class variations.

3.3 Multiple Instance Diverse Density Approaches

Let $\mathcal{B} = \{B_1, \cdots, B_n, \cdots, B_N\}$ represent a set of bags. Each bag B_n represents one data object by I[1] instances of features, i.e. $B_n = \{b_{n1}, \cdots, b_{ni}, \cdots, b_{nI}\}$. Each instance, $b_{ni} = \{b_{ni1}, \cdots, b_{nif}, \cdots, b_{niF}\}$, is an F-dimensional feature vector. Recall that under the standard MIL assumption, a bag is labeled as positive (class of interest), B_n^+, if and only if at least one of its instances is positive. Similarly,

[1]It is not required that all bags have the same number of instances. Here, we assume it is the case only to simplify notation.

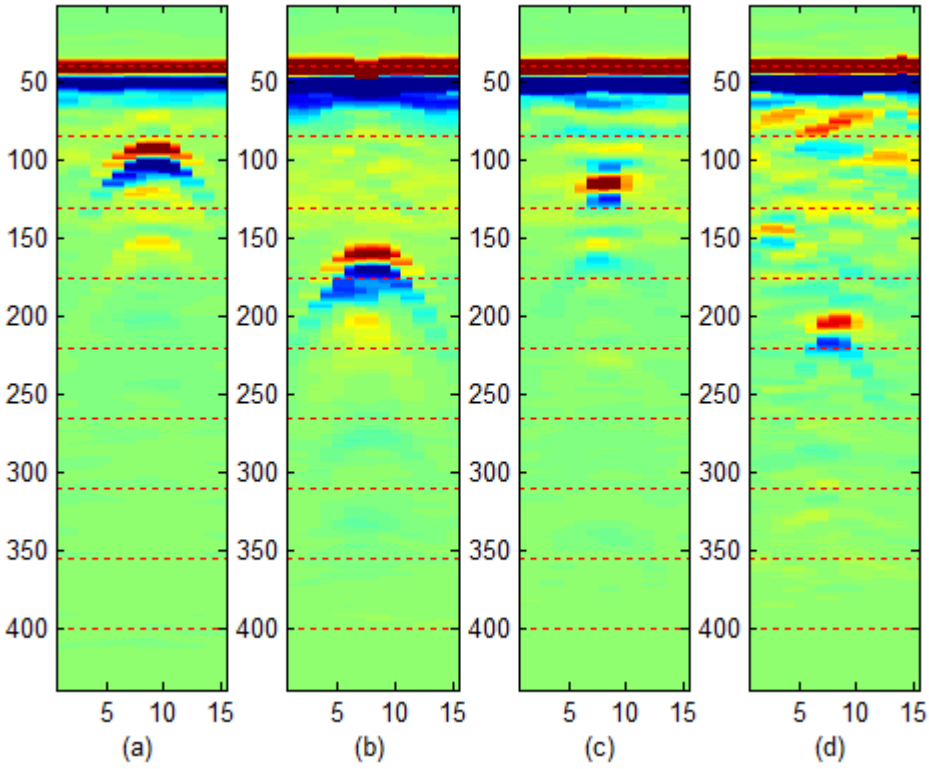


Figure 3.3: BEO Detection Example 2: The presence of diverse BEO types makes the use of a single target concept in MIL sub-optimal. Even with ideal depths selected, features extracted from the sample in (a) and (b) are likely to be extremely disparate from those extracted from the samples represented by (c) and (d).

a bag is labeled negative, B_n^-, if and only if all of its instances are negative. We assume that our data has N_{pos} positive bags and N_{neg} negative bags such that $N_{pos} + N_{neg} = N$. Let $\mathcal{B}^+ = \{B_1^+, \cdots, B_{N_{pos}}^+\}$ and $\mathcal{B}^- = \{B_1^-, \cdots, B_{N_{neg}}^-\}$ denote the subsets of positive and negative bags respectively.

The objective of the MDD approach is to identify K target concepts $\mathcal{T} = \{t_1, \cdots, t_K\}$, that describe regions in the instance feature space that include as many positive instances as possible and as few negative instances as possible. We first present our crisp representation of the MDD, called Crisp Multiple Instance Diverse Density (CMDD). Next, we describe our proposed Crisp Clustering of Multiple Instance (CCMI) approach to optimize the CMDD. Next, we highlight the advantages and disadvantages of the CMDD and motivate the need for a fuzzy or possibilistic alternative. Next, we propose the fuzzy and possibilistic alternatives, called the Fuzzy Multiple Instance Diverse Density (FMDD) and Possibilistic Multiple Instance Diverse Density (PMDD), and detail the Fuzzy Clustering of Multiple Instance (FCMI) and Possibilistic Clustering of Multiple Instance (PCMI) algorithms, which optimize the FMDD and PMDD respectively.

3.3.1 Crisp Clustering of Multiple Instance Data

The proposed Crisp Multiple Instance Diverse Density criteria are extensions to the sum of within-cluster distances, used in the K-means algorithm [38], to multiple instance data. First, we assume that the multiple instance data is partitioned into

K clusters. Each cluster Z_k includes a subset of bags such that

$$\bigcup_{k=1}^{K}\{B_{n,n=1..N}|B_n \in Z_k\} = \mathcal{B} \tag{3.3.1}$$

and

$$\bigcap_{k=1}^{K}\{B_{n,n=1..N}|B_n \in Z_k\}| = \emptyset \tag{3.3.2}$$

(3.3.1) and (3.3.2) insure that each bag is assigned to one and only one of the K clusters.

Second, we define the CMDD as the likelihood that the set of all bags $\mathcal{B}$ being generated by the set of target concepts $\mathcal{T}$:

$$CMDD(\mathcal{T}|\mathcal{B}) = \prod_{k=1}^{K}\prod_{B_{n\in Z_k}} Pr(t_k|B_n) \tag{3.3.3}$$

where t_k is the representation of partition Z_k and is alternately referred to as *target concept* t_k. $\mathcal{T} = \{t_1, ..t_K\}$ is the set of K target concepts that best describe the bags in $\mathcal{B}$. The CMDD in (3.3.3) is maximized when the bags are partitioned in such a way that each individual concept t_k best describes the bags in its associated partition Z_k.

Instead of maximizing (3.3.3), we seek the equivalent solution of minimizing

the negative log likelihood of the CMDD, i.e.

$$\begin{aligned} J(\mathcal{T};\mathcal{B}) &= -log(CMDD(\mathcal{T}|\mathcal{B})) \\ &= \sum_{k=1}^{K} \sum_{B_{n\in Z_k}} \{-log(Pr(t_k|B_n)))\}. \end{aligned} \tag{3.3.4}$$

Each bag B_n can be assigned to partition Z_k that is characterized by target concept t_k with the maximum likelihood, i.e.

$$B_n \in Z_k \, if \; k = argmax_{j=1}^{K} Pr(t_j|B_n) \tag{3.3.5}$$

The probability of bag B_n in a target concept t_k, $P(t_k|B_n)$ depends on the proximity of the instances within bag B_n to the target concept and whether or not $B_n \in \mathcal{B}^+$ or $B_n \in \mathcal{B}^-$. In our approach we use the NOISY-OR probability [49] and let

$$Pr(t_k|B_n) = \begin{cases} 1 - \prod_{i=1}^{I}(1 - Pr(b_{ni} \in t_k)) & if \; label(B_n)=1 \\ \prod_{i=1}^{I}(1 - Pr(b_{ni} \in t_k)) & if \; label(B_n)=0 \end{cases} \tag{3.3.6}$$

where $label(B_n) = 1$ for positive bags (i.e. $B_n \in \mathcal{B}^+$) , and $label(B_n) = 0$ for negative bags (i.e. $B_n \in \mathcal{B}^-$). Equations for positive and and negative data in (3.3.6) are disparate to match our interpretation that a given potential target concept t_k accurately represents positive density within a MIL dataset when at least instance within $B_n \in B^+$ is close (in terms of feature space) to t_k, and accurately represents negative density when all instances within $B_n \in B^-$ are distant from t_k.

We note that the above discrepancy has crucial ramifications for our membership assignment in (3.3.5) – specifically, positive bags will be assigned membership to target concepts with instances that are similar, while any given negative bag will be assigned membership to the target concept effectively most dissimilar. The former is reasonable and expected behavior, while the latter is aberrant – any negative bag similar to a (potentially poor) target concept will be ignored by the objective function.

One possible solution to the above issue is to assign a membership for negative bags that is complementary to that of positive bags (i.e. the target concept with minimum likelihood, in this case). In practice, however, we don't need or want negative bags to "belong" to any MDD-based target concept at all, and instead would prefer each to have the same potential impact on the objective function, regardless of relative proximity. To take this into account, we rewrite (3.3.5) as

$$B_n \in Z_k \; if \; \{(label(B_n)\text{=}1) \; AND \; (k = argmax_{j=1}^{K} Pr(t_j|B_n))\} \; OR \; \{label(B_n)\text{=}0\} \tag{3.3.7}$$

Hence, every negative bag will be considered by every target concept, while positive bags will be assigned to only one target concept. While this change does violate the standard crisp condition defined in (3.3.2) (i.e. negative bags will be assigned to more than a single target concept), we note that it does not undermine the underlying objective function in (3.3.3) or the succeeding optimization.

In (3.3.6), $Pr(b_{ni} \in t_k)$ is a metric that maps each instance b_{ni} to target concept t_k. Assuming that each t_k is characterized by a representative feature vector (e.g.

centroid), c_k and a scaling vector s_k, we use the following scaled Gaussian similarity function to represent $Pr(b_{ni} \in t_k)$:

$$Pr(b_{ni} \in t_k) = exp\{-\big(\sum_{l=1}^{L} s_{kl}(b_{nil} - c_{kl})^2\big)\} \tag{3.3.8}$$

In (3.3.8), the scaling vector s_k weights the role individual features play in defining the overall similarity [40].

Minimization of (3.3.4) involves solving 2 equations that are coupled and for which no closed form solution exists. On one hand, in (3.3.5), to assign bag B_n to one of its partitions Z_k we require its target concept probability $P(t_k|B_n)$. On the other hand, to compute $P(t_k|B_n)$ in (3.3.6), we need t_k which can only be computed after assigning all bags to their closest partition. One way to optimize (3.3.4) is to utilize alternating optimization in a form analogous to that used to derive the K-means clustering algorithm [38]. First, we assume that $P(t_k|B_n)$ is given and fixed for all $t_k, k = 1..K$ and all $B_n, n = 1..N$, and assign each B_n to the optimal partition Z_k using (3.3.7). Second, given a fixed partition $\mathcal{T} = \{t_1, ..t_K\}$, we treat each t_k independently and reduce (3.3.4) to K simpler problems:

$$J_k(t_k; \mathcal{B}) = \sum_{B_{n \in Z_k}} \{-log(Pr(t_k|B_n))\} \quad for\ k = 1..K. \tag{3.3.9}$$

Equation (3.3.9) can be minimized using a gradient descent approach as in [40]. Since t_k is characterized by centroid c_k and scaling vector s_k, we compute

$$\frac{\partial J}{\partial c_k} = -\sum_{B_{n \in Z_k}} \frac{1}{Pr(t_k|B_n)} \times \frac{\partial Pr(t_k|B_n)}{\partial c_k} \tag{3.3.10}$$

and

$$\frac{\partial J}{\partial s_k} = -\sum_{B_{n\in Z_k}} \frac{1}{Pr(t_k|B_n)} \times \frac{\partial Pr(t_k|B_n)}{\partial s_k} \tag{3.3.11}$$

As our definitions of (3.3.10) and (3.3.11) differ for positive and negative bags, we can rewrite the equations as

$$\frac{\partial J}{\partial c_k} = -\sum_{B^+_{n\in Z_k}} \frac{1}{Pr(t_k|B^+_n)} \times \frac{\partial Pr(t_k|B^+_n)}{\partial c_k} - \sum_{B^-_{n\in Z_k}} \frac{1}{Pr(t_k|B^-_n)} \times \frac{\partial Pr(t_k|B^-_n)}{\partial c_k} \tag{3.3.12}$$

and

$$\frac{\partial J}{\partial s_k} = -\sum_{B^+_{n\in Z_k}} \frac{1}{Pr(t_k|B^+_n)} \times \frac{\partial Pr(t_k|B^+_n)}{\partial s_k} - \sum_{B^-_{n\in Z_k}} \frac{1}{Pr(t_k|B^-_n)} \times \frac{\partial Pr(t_k|B^-_n)}{\partial s_k} \tag{3.3.13}$$

Using (3.3.6) and (3.3.12), it can be shown that

$$\begin{aligned}\frac{\partial Pr(t_k|B^+_n)}{\partial c_k} &= \left\{\sum_{i=1}^{I} \frac{1}{1-Pr(b^+_{ni}\in t_k)} \times \frac{\partial Pr(b^+_{ni}\in t_k)}{\partial c_k}\right\} \\ &\quad \times \prod_{i=1}^{I}(1-Pr(b^+_{ni}\in t_k))\end{aligned} \tag{3.3.14}$$

and

$$\begin{aligned}\frac{\partial Pr(t_k|B^-_n)}{\partial c_k} &= -\left\{\sum_{i=1}^{I} \frac{1}{1-Pr(b^-_{ni}\in t_k)} \times \frac{\partial Pr(b^-_{ni}\in t_k)}{\partial c_k}\right\} \\ &\quad \times \prod_{i=1}^{I}(1-Pr(b^-_{ni}\in t_k))\end{aligned} \tag{3.3.15}$$

Similarly, using (3.3.6) and (3.3.13), we obtain

$$\frac{\partial Pr(t_k|B_n^+)}{\partial s_k} = \left\{ \sum_{i=1}^{I} \frac{1}{1 - Pr(b_{ni}^+ \in t_k)} \times \frac{\partial Pr(b_{ni}^+ \in t_k)}{\partial s_k} \right\} \times \prod_{i=1}^{I} (1 - Pr(b_{ni}^+ \in t_k)) \tag{3.3.16}$$

and

$$\frac{\partial Pr(t_k|B_n^-)}{\partial s_k} = - \left\{ \sum_{i=1}^{I} \frac{1}{1 - Pr(b_{ni}^- \in t_k)} \times \frac{\partial Pr(b_{ni}^- \in t_k)}{\partial s_k} \right\} \times \prod_{i=1}^{I} (1 - Pr(b_{ni}^- \in t_k)) \tag{3.3.17}$$

Eqs (3.3.14)-(3.3.17) require derivatives for instance-concept probabilities with respect to c_k and s_k, which can be derived using (3.3.8) as follows:

$$\frac{\partial Pr(B_{ni} \in t_k)}{\partial c_{kl}} = 2(B_{nil} - c_{kl}) s_{kl}^2 e^{-\sum_{l=1}^{L} s_{kl}(b_{nil} - c_{kl})^2} \tag{3.3.18}$$

and

$$\frac{\partial Pr(B_{ni} \in t_k)}{\partial s_{kl}} = 2 s_{kl} (B_{nil} - c_{kl})^2 e^{-\sum_{l=1}^{L} s_{kl}(b_{nil} - c_{kl})^2} \tag{3.3.19}$$

Putting together Eqs (3.3.4)-(3.3.19) and our alternating optimization approach gives us the steps in Algorithm 5.

As outlined in Chapter 2, the crisp form of clustering in the SIL framework of machine learning suffers when the boundaries separating sub-classes of data are not well-defined. An instance that is almost equally similar to two distinct clusters will be assigned in a nearly arbitrary manner to the one with a marginally

Algorithm 5 The CCMI Algorithm

Inputs: $\mathcal{B}^+$ and $\mathcal{B}^-$: the complete sets of + and - bags.
K: the number of target concepts.
Outputs: $\mathcal{C}$: Centers of the K target concepts.
$\mathcal{S}$: Scales of the K target concepts.

Initialize c_k and s_k for $k = 1, .., K$
repeat
Assign all bags in $\mathcal{B}$ to to the closest target concept centroids using (3.3.5).
for k=1:K **do**
Use a few iterations of a gradient descent approach [40] to find the optimal c_k and s_k that minimizes (3.3.9) for the given partition.
end for
until centers do not change significatively or number of iterations is exceeded
return $\mathcal{C}$, $\mathcal{S}$

closer centroid. This same problem can be encountered when utilizing the CCMI algorithm in the MIL framework. In fact, there is an additional dimension of ambiguity in MIL data present. In figure 3.4, we assume that the data have two true target concepts with centers marked as TC_1 and TC_2. We display two bags that can belong to either target concept. The first bag, B_1 has five instances $\{a, b, c, d, e\}$ and one of its instances, a, is equally close to TC_1 and TC_2. This is the same scenario encountered in clustering traditional data. Another scenario, that is unique to MIL data, and makes the concept of fuzzy assignment more appropriate to MIL, is illustrated with a second bag, $B_2 = \{A, B, C, D, E\}$. In this case, one instance, A, is close to TC_1 while a different instance, B of the same bag is close to TC_2. In other words, the features that make B_2 similar to one target concept are different from the features that make the same bag similar to a different target concept. A natural alternative to this Crisp approach would be to utilize fuzzy clustering theory to permit a bag membership in more than a single

target concept to address these scenarios. The resulting Fuzzy Multiple-Concept Diverse Density metric and Fuzzy Clustering of Multiple Instance Data algorithm are presented in the next section.

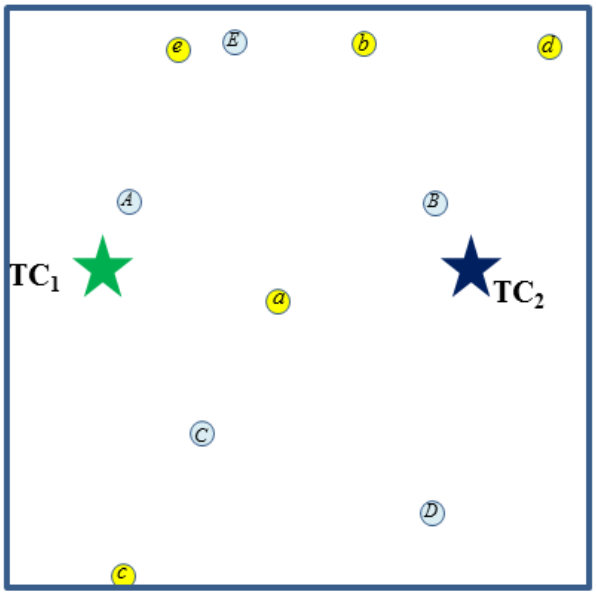


Figure 3.4: Two cases that require fuzzy assignment of a bag to multiple target concepts. The first bag, $B_1 = \{a, b, c, d, e\}$ has one instance, a, that is close to both target concepts TC_1 and TC_2. The second bag $B_2 = \{A, B, C, D, E\}$ has one instance, A, that is close to TC_1 and another instance, B, that is close to TC_2.

3.3.2 Fuzzy Clustering of Multiple Instance Data

Using a fuzzy approach, we assume that each bag, B_n, belongs to each target concept t_k with a membership u_{kn} such that:

$$u_{kn} \in [0, 1], \quad \text{and} \quad \sum_{k=1}^{K} u_{kn} = 1. \tag{3.3.20}$$

Let $\mathcal{U}$=$[u_{kn}]$ for $k = 1, \cdots, K$ and $n = 1, \cdots, N$. We define the fuzzy Multi-target concept Diverse Density (FMDD) metric as

$$FMDD(\mathcal{T},\mathcal{U}) = \prod_{n=1}^{N}\prod_{k=1}^{K} Pr(t_k|B_n)^{u_{kn}^m}. \tag{3.3.21}$$

In (3.3.21), $m \in (1, \infty)$ is a fuzzifier that controls the fuzziness of the partition as in the FCM [3]. The proposed FCMI algorithm seeks the optimal $(\mathcal{T},\mathcal{U})$ that maximize the FMDD in (3.3.21).

As with the CMDD and CCMI, we minimize the negative log-likelihood of (3.3.21) to avoid taking the products of extremely small numbers:

$$J(\mathcal{T},\mathcal{U}) = -log(FMDD(\mathcal{T},\mathcal{U})) = \sum_{n=1}^{N}\sum_{k=1}^{K} u_{kn}^m\{-log(Pr(t_k|B_n))\} \tag{3.3.22}$$

subject to the membership constraints in (3.3.20).

As with the CCMI, our approach involves an alternating optimization strategy – in this case, we alternate between fixing $\mathcal{T}$ and optimizing $\mathcal{U}$, and fixing $\mathcal{U}$ and optimizing $\mathcal{T}$. In the former case, we minimize (3.3.22) with respect to $\mathcal{U}$ by applying the method of Lagrange multipliers to obtain

$$J(\mathcal{T},\mathcal{U},\mathbf{\Lambda}) = \sum_{n=1}^{N}\sum_{k=1}^{K} u_{kn}^m\{-log(Pr(t_k|B_n))\} - \sum_{n=1}^{N}\lambda_n\big(\sum_{k=1}^{K} u_{kn} - 1\big) \tag{3.3.23}$$

As is the case for the CMDD, the FMDD adopts the NOISY-OR similarity function in (3.3.6) to model $Pr(t_k|B_n)$, and the Gaussian similarity function in (3.3.8) to model the required $Pr(b_{ni} \in t_k)$.

Assuming that the partial densities $(Pr(t_k|B_n)), n{=}1, \cdots, N$ and the columns of $\mathcal{U}$ are independent of each other, we can reduce (3.3.23) to the following N independent minimization problems:

$$J_n(\mathcal{T}, \mathcal{U}_n, \lambda_n) = \sum_{k=1}^{K} u_{kn}^m \{-log(Pr(t_k|B_n))\} - \lambda_n\Big(\sum_{k=1}^{K} u_{kn} - 1\Big), n = 1, \cdots, N \quad (3.3.24)$$

Next, we fix $\mathcal{T}$ and set the gradient of J_n to zero, we obtain

$$\frac{\partial J_n}{\lambda_n} = \sum_{k=1}^{K} u_{kn} - 1 = 0 \quad (3.3.25)$$

and

$$\frac{\partial J}{u_{qn}} = m u_{qn}^{m-1} log(Pr(t_q|B_n)) - \lambda = 0 \quad (3.3.26)$$

(3.3.25) simplifies to

$$\sum_{k=1}^{K} u_{kn} = 1 \quad (3.3.27)$$

Solving (3.3.26) for u_{qn} leads to:

$$u_{qn} = \left[\frac{\lambda}{m} \times \frac{1}{-log(Pr(t_q|B_n)}\right]^{\frac{1}{m-1}} \quad (3.3.28)$$

Substituting (3.3.28) back into (3.3.25) leads to

$$\sum_{k=1}^{K} \left[\frac{\lambda}{m} \times \frac{1}{-log(Pr(t_k|B_n)}\right]^{\frac{1}{m-1}} = 1 \quad (3.3.29)$$

, or

$$\frac{\lambda}{m} = \left[\frac{1}{\sum_{k=1}^{K} -log(Pr(t_k|B_n))^{1/(1-m)}}) \right]^{m-1} \tag{3.3.30}$$

Lastly, we can substitute (3.3.30) back into (3.3.28) to obtain an update equation for u_{qn}:

$$u_{qn} = \frac{-log(Pr(t_q|B_n))^{1/(1-m)}}{\sum_{k=1}^{K} -log(Pr(t_k|B_n))^{1/(1-m)}} \tag{3.3.31}$$

Just as membership assignment was an issue with negative bags for the CCMI, so too is the case for the FCMI. Namely, while (3.3.31) encapsulates desired behavior for positive bags (i.e. bags with instances closer to target concepts will be assigned larger values for u_{qn}), it also assigns high membership to negative bags with instances highly distant from target concepts. Once again, we assume that negative bags don't "belong" to target concepts, and so revise (3.3.31) to become

$$u_{qn} = \begin{cases} \frac{-log(Pr(t_q|B_n))^{1/(1-m)}}{\sum_{k=1}^{K} -log(Pr(t_k|B_n))^{1/(1-m)}} & if\ label(B_n)=1 \\ \\ \frac{1}{K} & if\ label(B_n)=0 \end{cases} \tag{3.3.32}$$

That is, we assume that negative bags are equally likely to belong to any of the k target concepts. We note that the formulation for negative bag membership in (3.3.32) ensures both conditions in (3.3.20) are satisfied.

Our FCMI algorithm adopts a similar approach to optimizing $\mathcal{T}$ as the CCMI. Namely, we split (3.3.22) into K independent optimizations, one for each t_k, so that

$$J(t_k;\mathcal{B},\mathcal{U}) = \sum_{n=1}^{N} u_{kn}^{m}\{-log(Pr(t_k|B_n))\} \quad for\ k = 1..K. \tag{3.3.33}$$

Each $J(t_k;\mathcal{B},\mathcal{U})$ in (3.3.33) may be optimized using a gradient descent approach as in [40]. As t_k is characterized by centroid c_k and scaling vector s_k, we require the gradients with respect to both components. Using (3.3.33), these can be derived as

$$\frac{\partial J}{\partial c_k} = -\sum_{n=1}^{N} \frac{u_{kn}^{m}}{Pr(t_k|B_n)} \times \frac{\partial Pr(t_k|B_n)}{\partial c_k} \tag{3.3.34}$$

and

$$\frac{\partial J}{\partial s_k} = -\sum_{n=1}^{N} \frac{u_{kn}^{m}}{Pr(t_k|B_n)} \times \frac{\partial Pr(t_k|B_n)}{\partial s_k} \tag{3.3.35}$$

As our definition of $Pr(t_k|B_n)$ in (3.3.34) and (3.3.35) differ for positive and negative bags, we can rewrite the equations as

$$\frac{\partial J}{\partial c_k} = -\sum_{n=1}^{N^+} \frac{u_{kn}^{m}}{Pr(t_k|B_n^+)} \times \frac{\partial Pr(t_k|B_n^+)}{\partial c_k} - \sum_{n=1}^{N^-} \frac{u_{kn}^{m}}{Pr(t_k|B_n^-)} \times \frac{\partial Pr(t_k|B_n^-)}{\partial c_k} \tag{3.3.36}$$

and

$$\frac{\partial J}{\partial s_k} = -\sum_{n=1}^{N^+} \frac{u_{kn}^{m}}{Pr(t_k|B_n^+)} \times \frac{\partial Pr(t_k|B_n^+)}{\partial s_k} - \sum_{n=1}^{N^-} \frac{u_{kn}^{m}}{Pr(t_k|B_n^-)} \times \frac{\partial Pr(t_k|B_n^-)}{\partial s_k} \tag{3.3.37}$$

From our definitions in (3.3.6) and (3.3.36), it can be shown that

$$\frac{\partial Pr(t_k|B_n^+)}{\partial c_k} = \left\{ \sum_{i=1}^{I} \frac{1}{1 - Pr(b_{ni}^+ \in t_k)} \times \frac{\partial Pr(b_{ni}^+ \in t_k)}{\partial c_k} \right\} \times \prod_{i=1}^{I} (1 - Pr(b_{ni}^+ \in t_k)) \tag{3.3.38}$$

and

$$\frac{\partial Pr(t_k|B_n^-)}{\partial c_k} = - \left\{ \sum_{i=1}^{I} \frac{1}{1 - Pr(b_{ni}^- \in t_k)} \times \frac{\partial Pr(b_{ni}^- \in t_k)}{\partial c_k} \right\} \times \prod_{i=1}^{I} (1 - Pr(b_{ni}^- \in t_k)) \tag{3.3.39}$$

Similarly, we use (3.3.6) and (3.3.37) to obtain

$$\frac{\partial Pr(t_k|B_n^+)}{\partial s_k} = \left\{ \sum_{i=1}^{I} \frac{1}{1 - Pr(b_{ni}^+ \in t_k)} \times \frac{\partial Pr(b_{ni}^+ \in t_k)}{\partial s_k} \right\} \times \prod_{i=1}^{I} (1 - Pr(b_{ni}^+ \in t_k)) \tag{3.3.40}$$

and

$$\frac{\partial Pr(t_k|B_n^-)}{\partial s_k} = - \left\{ \sum_{i=1}^{I} \frac{1}{1 - Pr(b_{ni}^- \in t_k)} \times \frac{\partial Pr(b_{ni}^- \in t_k)}{\partial s_k} \right\} \times \prod_{i=1}^{I} (1 - Pr(b_{ni}^- \in t_k)) \tag{3.3.41}$$

Eqs (3.3.38)-(3.3.41) require derivatives for instance-concept probabilities with respect to c_k and s_k, which can be derived using (3.3.8) as follows:

$$\frac{\partial Pr(B_{ni} \in t_k)}{\partial c_{kl}} = 2(B_{nil} - c_{kl}) s_{kl}^2 e^{-\sum_{l=1}^{L} s_{kl}(b_{nil} - c_{kl})^2} \tag{3.3.42}$$

and

$$\frac{\partial Pr(B_{ni} \in t_k)}{\partial s_{kl}} = 2 s_{kl} (B_{nil} - c_{kl})^2 e^{-\sum_{l=1}^{L} s_{kl}(b_{nil} - c_{kl})^2} \tag{3.3.43}$$

Putting together Eqs (3.3.22)-(3.3.43) and our alternating optimization approach, we outline the FCMI algorithm as follows:

Algorithm 6 The FCMI Algorithm

Inputs: $\mathcal{B}^+$ and $\mathcal{B}^-$: the sets of + and - bags.
K: the number of target concepts.
m: a fuzzifier parameter
Outputs: $\mathcal{C}$: Centers of the K target concepts.
$\mathcal{S}$: Scales of the K target concepts.
$\mathcal{U}$: Membership of all bags in all target concepts.

Initialize c_k and s_k for $k = 1, \cdots, K$
repeat
  Update $\mathcal{U}$ using (3.3.32).
  for k=1:K **do**
    Find optimal c_k and s_k by performing a few iterations of a gradient descent to minimize (3.3.33) for t_k.
  end for
until centers do not change significatively or number of iterations is exceeded
return $\mathcal{C}, \mathcal{S}, \mathcal{U}$

In Chapter 2, we stipulated the potential for membership ambiguities to arise in conventional fuzzy clustering theory when data points are equidistant from two or more cluster centroids. This same issue applies to the MIL domain. To that end, we consider the scenario in figure 3.5. As with figure 3.4, two true target concepts with centers marked as TC_1 and TC_2 are illustrated alongside sample bags B_1 and B_2 . Bag B_1 has 5 instances $\{a, b, c, d, e\}$ and bag B_2 has 5 instances

$\{A, B, C, D, E\}$. The fuzzy memberships, as defined in (3.3.32), consider only the relative distance from a bag to both target concepts. For example, instance $\{a\}$ of B_1 is equally close to both TCs, causing $Pr(TC_1|B_1)$ and $Pr(TC_2|B_1)$ to have comparable values (i.e. $u_{1,1} \approx u_{2,1} \approx 0.5$). Similarly, instance $\{B\}$ of B_2 is equally close to TC_1 and TC_2, making $u_{1,2} \approx u_{2,2} \approx 0.5$. Thus, despite having very different relative positions to the target concepts, the two bags will share nearly identical memberships across both target concepts. In more general terms, the fuzzy membership metric may fail to properly distinguish between bags that are equally close to and equally far from target concepts.

The above issue stems directly from the constraint that fuzzy memberships must sum to 1 across all target concepts for any given bag, regardless of bag probabilities. As with possibilistic clustering [34], we consider relaxing this constraint and propose the Possibilistic Multiple-Concept Diverse Density (PMDD) metric and the Possibilistic Clustering of Multiple Instance Data (PCMI) algorithm.

3.3.3 Possibilistic Clustering of Multiple Instance Data

As with the FMDD measure, we assume that each bag, B_n, belongs to each target concept t_k with a membership u_{kn} such that:

$$u_{kn} \in [0, 1] \tag{3.3.44}$$

However, unlike u_{kn} in (3.3.20), we do not require $\sum_{k=1}^{K} u_{kn}$ to be equal to 1.

Let $\mathcal{U}$=$[u_{kn}]$ for $k = 1, \cdots, K$ and $n = 1, \cdots, N$. We define the possibilistic

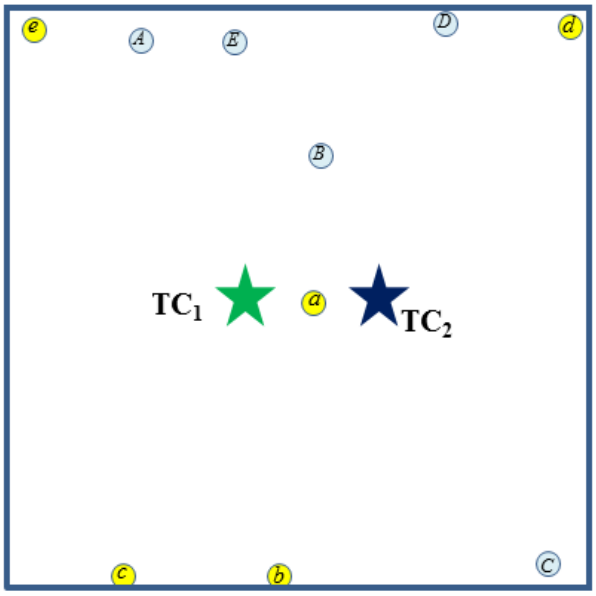


Figure 3.5: A 2 TC scenario in which fuzzy assignment fails to distinguish between the relationships of two bags to two target concepts. The first bag, $B_1 = \{a, b, c, d, e\}$ has one relevant instance, a, that is extremely close to both target concepts TC_1 and TC_2. The second bag $B_2 = \{A, B, C, D, E\}$ has one relevant instance, B, that is significantly more distant from both concepts. Memberships for both bags in both TCs will be ≈ 0.5.

Multi-target concept Diverse Density (PMDD) metric as

$$PMDD(\mathcal{T},\mathcal{U}) = \frac{\prod_{n=1}^{N}\prod_{k=1}^{K} Pr(t_k|B_n)^{u_{kn}^m}}{\prod_{n=1}^{N}\prod_{k=1}^{K} e^{\eta_k(1-u_{kn})^m}}. \tag{3.3.45}$$

The numerator in (3.3.45) is identical in form to the objective function for the FMDD, while the denominator forces u_{kn} to be as large as possible to counter the trivial solution. As with (3.3.21), m is a fuzzifier, and the proposed PCMI algorithm seeks the optimal $(\mathcal{T},\mathcal{U})$ that maximize the PMDD in (3.3.45).

Once again we minimize the negative log-likelihood of (3.3.45):

$$\begin{aligned} J(\mathcal{T},\mathcal{U}) &= -log(PMDD(\mathcal{T},\mathcal{U})) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} u_{kn}^{m}\{-log(Pr(t_k|B_n))\} - \sum_{n=1}^{N}\sum_{k=1}^{K}\{-log(e^{\eta_k(1-u_{kn})^m})\} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} u_{kn}^{m}\{-log(Pr(t_k|B_n))\} + \sum_{n=1}^{N}\sum_{k=1}^{K}\eta_k((1-u_{kn})^m) \quad (3.3.46) \end{aligned}$$

As with the FCMI, we alternate between fixing $\mathcal{T}$ and optimizing $\mathcal{U}$, and fixing $\mathcal{U}$ and optimizing $\mathcal{T}$. The latter occurs in form completely identical to that of the FCMI, and thus, (3.3.33)-(3.3.43) are also used to update the respective centroids and scales. In the former case, we assume the partial densities $(Pr(t_k|B_n)), n{=}1, \cdots, N$ and the columns of $\mathcal{U}$ are independent of each other, and reduce (3.3.46) to the following N independent minimization problems:

$$\begin{aligned} J_n(\mathcal{T},\mathcal{U}_n) &= \sum_{k=1}^{K} u_{kn}^{m}\{-log(Pr(t_k|B_n))\} \\ &+ \sum_{k=1}^{K}\eta_k((1-u_{kn})^m), n=1,\cdots,N. \quad (3.3.47) \end{aligned}$$

Computing the gradient of J with respect to the possibilistic memberships and setting it equal to zero, we obtain

$$\frac{\partial J}{u_{qn}} = m u_{qn}^{m-1} log(Pr(t_q|B_n)) - m\eta_q(1-u_{qn})^{m-1} = 0 \quad (3.3.48)$$

Isolating terms with u_{qn} in (3.3.48) yields

$$\frac{1-u_{qn}}{u_{qn}} = \{\frac{-log(Pr(t_q|B_n))}{\eta_q}\}^{\frac{1}{m-1}} \tag{3.3.49}$$

and solving directly for u_{qn} gives us the following update equation:

$$u_{qn} = \frac{1}{1-\{\frac{log(Pr(t_q|B_n))}{\eta_q}\}^{\frac{1}{m-1}}} \tag{3.3.50}$$

In a manner similar to the FCMI and CCMI, the membership update equation is well suited to positive bags. For negative bags, we assume that they belong to all target concepts with a possibility degree of 1. Thus, we replace (3.3.50) with

$$u_{qn} = \begin{cases} \frac{1}{1-\{\frac{log(Pr(t_q|B_n))}{\eta_q}\}^{\frac{1}{m-1}}} & if\ label(B_n)\text{=}1 \\ 1 & if\ label(B_n)\text{=}0 \end{cases} \tag{3.3.51}$$

The resulting PCMI algorithm is summarized below.

Algorithm 7 The PCMI Algorithm

Inputs: $\mathcal{B}^+$ and $\mathcal{B}^-$: the sets of + and - bags.
K: the number of target concepts.
m: a fuzzifier parameter.

Outputs: $\mathcal{C}$: Centers of the K target concepts.
$\mathcal{S}$: Scales of the K target concepts.
$\mathcal{U}$: Membership of all bags in all target concepts.

Initialize c_k and s_k for $k = 1, \cdots, K$
repeat
  Update u_{kn} using (3.3.51).
  for k=1:K **do**
    Find optimal c_k and s_k by performing a few iterations of a gradient descent to minimize (3.3.33) for t_k.
  end for
until centers do not change significantly or number of iterations is exceeded
return $\mathcal{C}, \mathcal{S}, \mathcal{U}$

Having examined crisp, fuzzy, and possibilistic algorithms for optimization of the MDD metric, the remaining question becomes how to identify the optimal number of target concepts K. Selecting too few target concepts in optimization will lead to a solution that fails to cover the complete distribution responsible for true positive instances. On the other hand, selecting too many will introduce target concepts that run the risk of either duplicating or splitting true concepts or failing to represent them at all. While the former problem is not easily addressed (i.e. there is no obvious means of introducing target concepts to cover the missing distribution(s) post-operation), the latter can be addressed through the selected elimiation of target concepts. We now turn to two means of selecting target concepts: the first of these utilizes possibilistic memberships to "merge" duplicate target concepts, while the second eliminates "weak" target concepts with insufficient bag probability metrics.

3.3.4 Merging Clusters Using Possibilistic Memberships

In addition to addressing the FCMI's shortcomings in distinguishing between equally close or equally distant instances as discussed in Section 3.3.2, the relaxed membership constraints for the possibilistic function offer the potential for one or more similar target concepts to merge into a single target concept during the operation of the PCMI algorithm. This is expected and useful behavior, because it permits us to start the optimization with more target concepts than are expected to fit the distribution and eliminate duplicate target concepts after convergence.

To merge similar target concepts, we follow the strategy used in [26, 53]. In particular, we use a selected merging threshold θ_M and consider two target concepts t_k and $t_{k'}$ duplicates if

$$\frac{\sum_{n=1}^{N^+} |u_{kn} - u_{k'n}|}{\sum_{n=1}^{N^+} |u_{kn}| + \sum_{n=1}^{N^+} |u_{k'n}|} < \theta_M \tag{3.3.52}$$

where u_{kn} and $u_{k'n}$ are computed using 3.3.51. Note that only positive bags are considered in this calculation, due to the uniform selection of a membership of 1 for negative bags. Assuming two concepts t_k and $t_{k'}$ meet the criteria for merging, we can arbitrarily mark for one deletion, or instead select the one with overall lesser possibilistic membership in negative bags for deletion:

$$ind_{MERGE}(t_k, t_{k'}, \mathcal{B}^+) = argmin_{q=k,k'} \sum_{n=1}^{N^+} |u_{qn}| \tag{3.3.53}$$

Merging all duplicate TCs given set $\mathcal{T}$ can be accomplished by peforming simple pairwise possibilistic membership comparisons using (3.3.52) and selecting appropriate indices for deletion using (3.3.53). A summary of this routine follows.

While merging duplicate targets can remove extraneous strong target concepts, it does not address the issue of target concepts that fail to model true positive distributions. As the number of target concepts increases relative to the true generative concepts, we expect to see more of these weak target concepts. In the following, we detail a simple means of target concept elimination that relies on bag probabilities.

Algorithm 8 Merging TCs Using Possibilistic Memberships

Inputs: $\mathcal{B}$: a MI Dataset
$\mathcal{T}$: a set of target concepts synthesized using the PCMI.
Outputs: $\mathcal{T}'$: A set of target concepts with duplicate entries eliminated

for k=1:K **do**
  for q=k+1:K **do**
    Determine if target concepts t_k and t_q are candidates for merging using (3.3.52)
    If merging is required, add index k or q to a pool $\mathcal{T}_{merge}$ for elimination using (3.3.53).
  end for
end for
$\mathcal{T}'$ is defined as the set difference between $\mathcal{T}$ and $\mathcal{T}_{merge}$
return $\mathcal{T}'$

3.3.5 Eliminating Weak Target Concepts

A simple way to quantize a target concept's strength is through aggregate positive bag probability, which we define as

$$\phi_{QN}(t_k, \mathcal{B}) = \frac{1}{N^+}\sum_{n=1}^{N^+} Pr(t_k|B_n^+) < \theta_{QN} \tag{3.3.54}$$

where $Pr(t_k|B_n^+)$ can be computed with the NOISY-OR or MLCE metric using (2.2.2) or (2.2.4) respectively. While (3.3.54) provides an idea of how many positive bags respond to a target concept, it ignores the fact that a weak target concept may simply respond to a much smaller number of positive bags than negative bags, and that a strong target concept may respond to a much larger number of positive bags than negative bags. To that end, we utilize a second measure based on the ratio of aggregate positive bag probabilities and aggregate negative bag

complementary probabilities as follows:

$$\phi_{QL}(t_k, \mathcal{B}) = \frac{\frac{1}{N^+}\sum_{n=1}^{N^+} Pr(t_k|B_n^+)}{1 - \frac{1}{N^-}\sum_{n=1}^{N^-} Pr(t_k|B_n^-)} < \theta_{QL} \tag{3.3.55}$$

.

ϕ_{QN} can be seen as a quantitative validity metric – the larger the value, the more relevant the target concept is to the positive data. ϕ_{QL} can be seen as a quality validity metric. Larger values indicate that a target concept strongly distinguishes between positive and negative bags, while a value close to 1 indicates it cannot distinguish between the two classes. We can specify two minimum quality thresholds θ_{QN} and θ_{QL} and remove target concept t_k if either $\phi_{QN}(t_k, \mathcal{B}) < \theta_{QN}$ or $\phi_{QL}(t_k, \mathcal{B}) < \theta_{QL}$.

Up to this point, we have addressed MDD analysis with the goal of discovering and refining multiple distinctive target concepts in a MIL dataset. However, we have not yet broached the subject of using these metrics and algorithms in an effort to accurately classify prospective data as positive or negative based on the target concepts. In the next chapter we examine both simple approaches to classifying data using optimized target concepts, as well as more sophisticated techniques such as employing embedded feature space and negative target concepts.

CHAPTER 4

MIL CLASSIFICATION STRATEGIES USING MULTIPLE TARGET CONCEPTS

In Chapter 3, we considered strategies geared towards unsupervised learning in the MIL setting. We have formulated the Multiple Diverse Density (MDD) objective function and derived crisp, fuzzy, and possibilistic algorithms to optimize it (CCMI, FCMI, and PCMI respectively). In this chapter, we focus on supervised learning for multiple instance data. We show that target concepts learned during multiple instance clustering can be used to derive a robust and efficient classification algorithm. Just as training samples in a MIL dataset face notable issues of ambiguity, so too do testing samples. Specifically, the issue arises as to how one should aggregate individual instance labels or information into a cumulative label for a bag – making classification within a MIL context a challenging task.

The testing sample ambiguity problem is often addressed as a natural extension of assumptions made in the training process. Algorithms which predicate themselves on the single-instance confirmation assumption generally assign a label

according to whether at least one instance meets "positive" criteria in a prospective bag (in which case the bag is deemed positive) or none meet said criteria (in which case the bag is deemed negative). For example, as outlined in Chapter 2, an APR classifier [16] selects a label based on whether or not any instances fall within the hyper-rectangle constructed during training. Similarly, the original DD classifier [40] selects a class label for a testing sample according to whether or not at least one prospective instance in a bag is within a pre-trained distance threshold of the optimized target concept. In the following, we first examine a simple classification strategy relying on the learned target concepts.

4.1 MDD-Based Bag Probability Classification of MIL Data

The simplest approach to classification using the learned target concepts utilizes the closest (or most probable) target concept. Given a testing bag B_{te} and a set of K optimized target concepts $\mathcal{T} = \{t_1 \cdots t_k\}$, we first compute $Pr(t_k|B_{te})$ for each target concept t_k using

$$Pr(t_k|B_{te}) = 1 - \prod_{i=1}^{I} (1 - Pr(b_{te,i} \in t_k)) \tag{4.1.1}$$

Higher values of $Pr(t_k|B_{te})$ in (4.1.1) indicate a strong correlation with the target concept and, therefore, high likelihood that the testing sample is positive. As an alternative to the NOISY-OR metric in (4.1.1), and in accordance with classification strategies commonly utilized in MIL research [1, 40], we also consider the most-likely cause estimate as a viable metric for testing bag probability – that

is,

$$Pr(t_k|B_{te}) = \max_{i=1}^{I} Pr(b_{te,i} \in t_k) \tag{4.1.2}$$

Individual instance probability, $Pr(b_{te,i} \in t_k)$, in (4.1.1) and (4.1.2) can be computed using a form similar to (3.3.8), or

$$Pr(b_{te,i} \in t_k) = exp\{-\big(\sum_{l=1}^{L} s_{kl}(b_{te,i,l} - c_{kl})^2\big)\} \tag{4.1.3}$$

Where c_k and s_k are the centroid and scales for target concept t_k respectively, learned using one of our clustering algorithms (CCMI, FCMI, or PCMI).

Once a set of bag probabilities $Pr(t_1|B_{te}) \cdots Pr(t_K|B_{te})$ is computed for the K target concepts, we can use a simple aggregation operator to assign a net confidence to bag B_{te}. A reasonable choice would be to assume that a reliably positive bag should be as close as possible to at least one target concept, so we use the max operator as follows:

$$Conf(B_{te}, \mathcal{T}) = \max_{k=1}^{K} Pr(t_k|B_{te}) \tag{4.1.4}$$

The probability-based classifcation procedure is summarized below:

While the approach above is appealing from the standpoint that it is simple and can be derived naturally from the optimization process, it faces at least two critical issues. For one, taking the maximum of target concept probabilities as a net confidence for a sample ignores differences in the reliability of distinct con-

Algorithm 9 Simple MDD Probability Classifcation

Inputs: B_{te}: a prospective testing sample
$\mathcal{T}$: a set of target concepts synthesized using the CCMI,FCMI, or PCMI.
Outputs: $Conf_{te}$: A probability-based confidence that B_{te} is positive.

for k=1:K **do**
Compute bag probability in the k_{th} target concept $Pr(t_k|B_{te})$ using either (4.1.1) or (4.1.2).
end for
Compute net confidence $Conf_{te} = Conf(B_{te}, \mathcal{T})$ using (4.1.4).
return $Conf_{te}$

cepts. Secondly, this procedure eliminates the possibility for a bag's relationship to more than one of the target concepts to play defining roles in producing a label. Fortunately, the embedded space paradigm introduced in Chapter 2 provides a convenient means of mapping relationships between bags in an MIL dataset and derived target concepts to construct a set of representative features, permitting classification to take into account these two nuances.

4.2 Embedded Feature Space Transformation using Target Concepts

In Chapter 2, we presented examples of how MILES and other MIL algorithms use an embedded feature space approach to transform the bag-of-instance representation for each sample into a one-dimensional embedded space feature vector, thereby allowing standard classifiers to be constructed and applied to the remapped feature vectors. Instead of considering all available instances in a dataset for mapping, a similar approach can be considered by utilizing only the

target concepts synthesized by the CCMI,FCMI, and PCMI algorithms in the feature embedding process.

Assume we have a set of bags $\mathcal{B}$ and a set of K target concepts $\mathcal{T}$. Each bag is mapped to each target concept using either a similarity (e.g. probability) or dissimilarity (e.g. distance) metric. While similarity metrics have value from the standpoint of a narrow, controlled interval for each feature, they may result in substantial information loss, as all dissimilar bags (with potentially very different distance values) will map to values close to zero. We therefore choose to utilize a most-likely distance estimate as follows:

$$dmin(B_n, t_k) = \min_{i=1}^{I} d(b_{ni}, t_k), for\, t_k \in \mathcal{T} \tag{4.2.1}$$

Using (4.2.1), each bag is mapped to a feature vector with K dimensions based on the minimum distance across all instances within the given bag. Appropriate choices for $d(b_{ni}, t_k)$ are examined later in this section. The assumption is that at least one distance between one instance and one target concept is discriminatively small. However, as discussed by the authors of [11], this measure may not best measure the distinction between positive and negative data, depending upon the dataset. For example, in the so-called "distribution" dataset paradigm, a general trend for multiple or all instances in a bag towards a positive distribution is the more reliable measure. To that end, we choose to consider a mean distance estimate as follows:

$$dmean(B_n, t_k) = \frac{\sum_{i=1}^{I} d(b_{ni}, t_k)}{I}, for\, t_k \in \mathcal{T} \tag{4.2.2}$$

In addition to (4.2.1) and (4.2.2), we also consider an aggregation-based feature that consolidates information from multiple TCs to a single bag at once. For example, a minimal overall TC distance mapping can be computed as

$$dmin(B_n, \mathcal{T}) = \min_{k=1}^{K} \min_{i=1}^{I} d(b_{ni}, t_k), for\, t_k \in \mathcal{T} \tag{4.2.3}$$

The metric in (4.2.3), which mirrors the minimum Hausdorff distance [11], represents the overall proximity of a given bag to its nearest TC, which can be a strong indicator of the bag's label as positive, regardless of other distance measures.

Thus far, we have addressed metrics that produce features reflecting a bag's similarity to (or lack of dissimilarity to) to standard target concepts (i.e. those representative of positive data density). As [11] notes, however, there is value both in considering a bag's similarity to positive data, as well as considering a sample's dissimiliarity to negative data. Henceforth, we will refer to the target concepts we have constructed using the MDD to address the former consideration as "positive target concepts." In order to address the latter consideration, we define "negative target concepts" below, and explore a simple means of deriving them that complements our FCMI and PCMI approaches.

4.3 Deriving Negative Target Concepts in Multiple Instance Data

A negative target concept can be defined as a point or region in the feature space densely populated by instances from negative bags. The importance of negative target concepts is best viewed through the prism of the MIL concept datasets defined in [11]. Specifically, only the "Concept" dataset – which relies on traditional MIL assumptions of a single or few true instances per positive bag being responsible for its label, provides little incentive to utilize negative target concepts. Meanwhile, the "Distribution" dataset – which assumes multiple instances from positive bags trend towards the "true" target concept point or region, have potentially great use for negative target concepts, since multiple instances from positive bags will be expected to have aggregately larger dissimilarity metrics to these concepts relative to instances from negative bags. Lastly, the "Multi-Concept" dataset – which often has outlying data points for representative instances in positive bags, will also likely see substantially increased dissimilarity metrics between negative target concepts and positive bags, but potentially no or little correlation to learned positive concepts.

Because we have no information as to the true label of instances in positive bags, they are largely irrelevant in defining the adequacy of a negative target concept. The main advantage of this observation is that negative target concepts need not be defined through the bag-of-instance mechanism of the multiple in-

stance framework. Instead, we can consider any instance from any negative bag to play an equal potential role in shaping negative target concepts. To that end, we pool all instances from all negative bags and perform conventional clustering (e.g. Kmeans) to obtain a set of K^- negative target concepts (defined by the cluster centroids). If needed, we can then estimate scales comparable to those acquired with the CCMI, FMCI, or PCMI algorithms by computing values inversely proportional to the standard deviation of feature values for the members of each derived cluster. A summary of these steps is provided in Algorithm 10.

Algorithm 10 Negative TC Clustering

Inputs: $\mathcal{B}^-$: the set of - bags.
K^-: the number of target concepts.
Outputs: $\mathcal{C}^-$: Centers of the K^- negative target concepts.
$\mathcal{S}^-$: Scales of the K^- negative target concepts.

Construct I^- by pooling all instances from all elements of $\mathcal{B}^-$
Use any conventional clustering algorithm to partition I^- into K^- clusters.
for $k = 1 : K^-$ **do**
  Define negative concept centroid c_k^- as the centroid of the k^{th} cluster.
  Find scale vector s_k^- using the standard deviations in each feature dimension for the members of the k^{th} cluster.
end for
return $\mathcal{C}^-$, $\mathcal{S}^-$

4.4 Embedded Feature Space Transformation using Negative Target Concepts

Our major goal in deriving negative target concepts is to augment positive target concepts in an effort to accurately classify Multiple Instance data. One clear means of doing so is to perform the feature embedding transformation described

in section 4.2 using the set of negative TCs $\mathcal{T}^-$ as follows:

$$dmin(B_n, t_k^-) = \min_{i=1}^{I} d(b_{ni}, t_k^-), for\, t_k^- \in \mathcal{T}^- \tag{4.4.1}$$

As can be seen, our formulation for embedding features utilizing negative TCs in (4.4.1) is identical in form to the equation for positive TCs in (4.2.1). In the case of our positive target concept embedding in (4.2.1), each distance provides information relevant to the "concept" dataset paradigm outlined earlier, in that at least one representative instance from each positive bag should be close to at least one positive TC. For negative target concepts, the information we glean from (4.4.1) mirrors the "mean distance" measure calculated for positive target concepts using (4.2.2) in that it is more relevant to the "distribution" dataset paradigm, as we expect negative bags to have more instances close to negative TCs than positive bags.

Additionally, we consider the potential value of mapping the maximum dissimilarity from negative target concepts to bags:

$$dmax(B_n, t_k^-) = \max_{i=1}^{I} d(b_{ni}, t_k^-), for\, t_k^- \in \mathcal{T}^- \tag{4.4.2}$$

Equation (4.4.2) is expected to be useful for datasets which meet the multi-concept dataset criteria, as outlying positive bag instances should demonstrate relatively large distance values.

The distance $d(b_{ni}, t_k)$ in any of our feature-embedding measures may be computed using any standard distance metric. Here, we use the scaled Euclidean

function:

$$d(b_{ni}, t_k) = \sum_{j=1}^{F} s_{kj}(b_{nij} - c_{kj})^2 \tag{4.4.3}$$

In (4.4.3), $j = 1, \cdots, F$ are the feature space dimensions. When Positive TCs are being evaluated (i.e. $t_k \in \mathcal{T}$) we use values for c_k and s_k obtained through CCMI,FCMI, or PCMI optimization. For negative target concepts (i.e. $t_k \in \mathcal{T}^-$) the values we use for c_k and s_k are based on the cluster centroids and standard deviation as outlined in algorithm 10.

The mechanics defined above cumulatively map each bag B_n in prospective MIL datasets to a simple feature vector with up to $(2K + 2K^- + 1)$ dimensions and include

$$\begin{aligned} \nu(B_n, \mathcal{T}^+, \mathcal{T}^-) \;=\; & [dmin(B_n, t_1) \cdots dmin(B_n, t_K), dmean(B_n, t_1) \cdots dmean(B_n, t_K), \\ & dmin(B_n, t_1^-) \cdots dmin(B_n, t_K^-), dmax(B_n, t_1^-) \cdots dmax(B_n, t_K^-), \\ & dmin(B_n, \mathcal{T}] \end{aligned} \tag{4.4.4}$$

After embedding maps these bags to feature vectors, traditional classification algorithms such as the SVM [12] can be used to classifiy the data.

CHAPTER 5

EXPERIMENTAL RESULTS AND ANALYSIS

The proposed algorithms were evaluated using multiple synthetic and real data collections, both to analyze the efficacy of the algorithms in producing robust, meaningful clusters and target concepts (TCs), as well as to produce a competitive vehicle for classification of MI data. The former criteria are addressed through two synthetic data experiments and one real data experiment. The first set of synthetic data is relatively simple, and intended to illustrate the idea of clustering multiple instance data, and address how it can be used to more reliably find a set of true target concept distributions (hereafter referred to as "true concepts") than the Diverse Density (DD) [40] algorithm. In addition, these data are used to illustrate how the TC merging approach outlined in Section 3.3.4 and weak TC removal approach outlined in Section 3.3.5 may be used to select the optimal number of TCs. The second synthetic data experiment analyzes the sensitivity of the proposed algorithms to various parameters. Lastly, the real data experiment employs a buried explosive object (BEO) dataset to examine our algorithms' ability to locate distinctive BEO clusters by virtue of object composition, shape,

depth, and other defining characteristics.

Multiple Instance Classification utilizing our approaches is evaluated through two real world data experiments. The first of these employs several benchmark datasets and compares the performance of our approach to other classic embedded-feature space MIL approaches. The second once again utilizes a BEO dataset as before – in this case, our ability to distinguish BEOs from so-called false alarm, or clutter objects, is compared to approaches currently deployed in the field for BEO detection.

5.1 Illustration of the Proposed Multiple Instance Clustering Algorithms

5.1.1 Illustrative Synthetic Datasets

We generated 2 simple synthetic datasets for the purposes of establishing the utility of our approach, as well as comparing it to the baseline Diverse Density (DD) algorithm [40]. The first one, Data-1TC, has only one true concept. It is used mainly to establish the fact that our proposed approach effectively generalizes the DD when only true concept is present in the data. The second dataset, called Data-2TC, has 2 true concepts and is used to illustrate the need for our approach and to analyze the performance and compare the crisp (CCMI) and fuzzy (FCMI) algorithms.

Data-1TC onsists of 20 positive bags and 20 negative bags. All bags contain

between 2 and 6 instances (the number chosen randomly), and the instance feature space is 2-dimensional, with dimensions denoted as x and y. The actual instances for each bag are generated using the following strategy:

1. The first instance from each positive bag is generated randomly within a region of radius of 0.1 from the location (0.3,0.5). This point corresponds to the datset's true concept.

2. The remainder of instances from each bag have a random location (x,y) such that $x \in [0, 2]$ and $y \in [0, 2]$.

3. All instances from negative bags are generated within the same interval ((0,0 to (2,2)), subject to the constraint that they cannot fall within a buffered radius of 0.15 within the true concept coordinate at (0.3,0.5).

Figure 5.1(a) depicts this dataset. Positive bags are denoted by an asterix, while negative bags are denoted by dots. The shaded region corresponds to the true concept region. As can be seen, the true concept region includes instances from positive bags but not from negative bags. The location of this region is entirely unknown to the algorithms, and is included purely for validation purposes.

The second synthetic dataset used in this experiment, Data-2TC, is nearly identical to Data-1TC, but includes 20 additional bags corresponding to a second true concept. This data was generated using the following strategy:

1. 20 positive bags were generated from the first true concept located at (0.3,0.5). As in Data-1TC, the first instance of each of these bags is generated randomly within a region of radius of 0.1 from this point.

2. 20 positive bags were generated from a second true concept located at (0.5,1.5). Here also the first instance for each of these bags was generated randomly within a region of radius of 0.1 from this point.

3. The remainder of instances for both the first and second true concepts were generated at random points between (0,0) and (2,2). We made the additional requirement that these points could not fall within a buffered radius of 0.15 of the alternate true concept centroid (i.e. instances from the first true concept bag do not fall within a radius of 0.15 of (0.5,1.5). Without this constraint the two true concepts may have been less distinct, making the evaluation less reliable.

4. Negative bags were generated within the same interval, subject to the constraint that they cannot fall within a buffered radius of 0.15 of either true concept point.

Figure 5.1(b) displays Data-2TC. Instances from positive bags generated from the first true concept are depicted by an asterix, those from the second true concept are depicted as diamonds, and instances from negative bags are depicted as dots. The two shaded areas represent the two true concept regions.

5.1.2 Results and Analysis using Data-1TC

We use Data-1TC to evaluate and compare the performance of the proposed CCMI and FCMI algorithms to the Diverse Density aligorithm [40] when only one true concept is present in the data. To that end, we ran each algorithm 50 times with

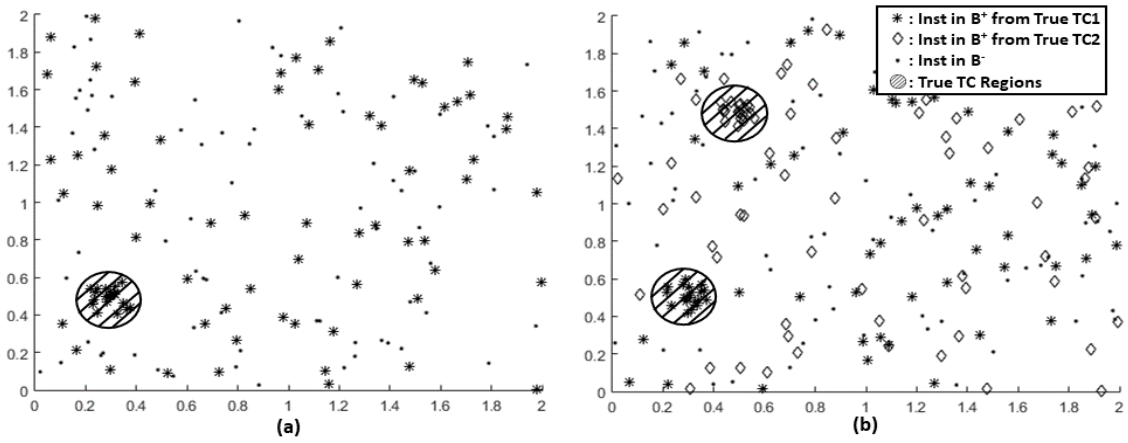


Figure 5.1: Illustrative Synthetic Datasets. Shaded regions denote true concepts. Negative bags are denoted by dots. (a) Data-1TC: Positive bags are denoted by an asterix. (b) Data-2TC: Positive bags generated with true concept 1 are denoted by an asterix, while positive bags generated with true concept 2 are denoted by a diamond.

the following parameters:

1. The starting point for the target concept was chosen randomly from among all instances in positive bags in our data.

2. The scaled Gaussian similarity function in (3.3.8) was used to compute instance-concept similarity. The scales of both dimensions were initialized to 1 at first, but later initialized at 2 and the runs repeated due to the poor performance with the former value.

3. We let each algorithm run for 400 iterations. Even though the DD and our algorithms often stabilize after a few iterations (less than 50), we wanted to make sure that a stable state was reached in each case.

For the DD algorithm, only 50% of the 50 runs converged correctly to the true concept at (0.3,0.5) when scales were initialized to 1. By contrast, 74% of the runs (based on the same initial TC centroids) converged to the true concept when

scales were initialized to a value of 2. This emphasizes both the role scales play in DD optimization as well as the potential importance of good initialization.

A typical, successful run of the DD algorithm is shown in figure 5.2(a). The path taken by the DD algorithm is shown by the black line. In remaining 26% of the DD runs, the target concept converged to a sub-optimal location. An example of this behavior is shown in figure 5.2(b), where the run ended outside the instance feature space. It is worth noting that in our experiments, the more distant the initial target concept is from the true one, the more likely the DD is to fail to reach the optimal TC.

Figure 5.4(a) depicts an error bar plot of the negative log of the Diverse Density objective function for the 50 runs of the DD algorithm on Data-1TC as it decreases over the operation of the first 25 iterations. It is worth noting that while the objective function does indeed gradually decrease, the runs which fail to converge to the true concept add a substantial level of volatility to the function even in later iterations.

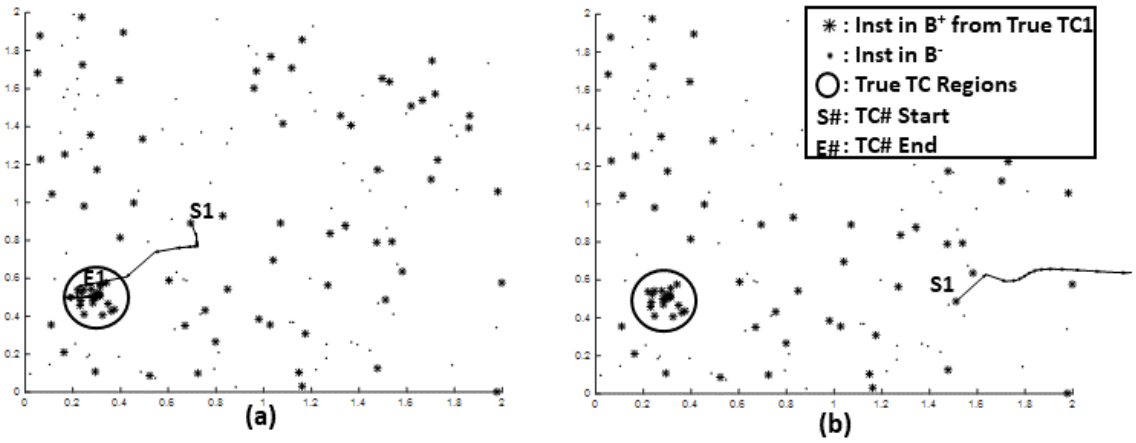


Figure 5.2: Sample runs of the DD algorithm on Data-1TC (a) A sample run that successfully converges to the true concept. (b) An alternate DD run in which the algorithm fails to locate the true concept.

Next, we ran the CCMI and FCMI on Data-1TC while fixing the number of target concepts, K, to 1. All other parameters for this experiment, as well as initial target concepts for the 50 runs, were identical to those used in the standard DD. To provide roughly the same number of the gradient descent steps (400) to our optimization, 100 "outer" loops were made for our alternating optimization, with 4 iterations of the line search per loop. In the case of the FCMI, the fuzzifier value was set to 1.0625. The results of these two experiments proved to be identical, run-for-run, to those of the standard DD, thus confirming that our proposed crisp and fuzzy algorithms effectively generalize the DD algorithm when K is set to 1.

Two major, related issues arise in standard single instance clustering: (1) How does the algorithm behave when the actual number of clusters is unknown? and (2) Does using an overspecified number of clusters improve the likelihood of detecting the true clusters? To investigate these issues for the case of multiple instance data, we ran the CCMI and FCMI algorithms on Data-1TC using $K = 2$ TCs, keeping all other parameters identical. For all runs of the CCMI and FCMI, we observed two successful behaviors. In the first case, one of the two TCs converges to the true concept location, and the other reaches a suboptimal location in or outside the feature space. In the second case, both TCs reach the true concept location. Figure 5.3 illustrates each of these outcomes.

Technically speaking, none of the above scenarios utilizing two TCs correctly describes the data, because there is only a single true concept present in the data. As discussed in Chapter 3, however, we can utilize the possibilistic clustering algorithm (PCMI) and bag probabilities to merge target concepts and selectively

eliminate "weak" clusters in an effort to obtain the correct true concept.

To that end, we ran the PCMI at the conclusion of each CCMI and FCMI run specified above for an additional 100 outer iterations using identical parameters, and a constant value of $-log(.75)$ for all η. We then applied the merging criteria outlined in Section 3.3.4 using $\theta_M = 0.05$, and the weak TC elimination approach outlined in Section 3.3.5 using a value of 0.5 for ϕ_{QN} and 5 for ϕ_{QL}. After the selection process, we noticed a substantial improvement to the rate of locating the true concept, with an 82% success rate for the CCMI, and an 86% success rate for the FCMI. 14% of the remaining runs ended with a sub-optimal target concept for both the CCMI and FCMI. The lone remaining 4% of CCMI runs resulted in a failure to eliminate one of the two target concepts.

Table 5.1 summarizes the results of our experiments on Data-1TC. Additionally, figure 5.4(b) shows an error-bar depiction of the objective function for the 50 runs of the CCMI for the first 25 (inner-loop) iterations of the algorithm. We make two observations regarding this figure. First, we note that the relatively stable decrease in the objective function indicates that our alternating optimization approach to the Multiple Concept Diverse Density does not appear particularly volatile between the membership assignment step and target concept optimization step. Secondly, while the mean objective function value across iterations is similar to that observed for the DD in figure 5.4(a), the decrease in standard deviation is indicative of the algorithm's consistency in finding solutions with similar objective function metrics.

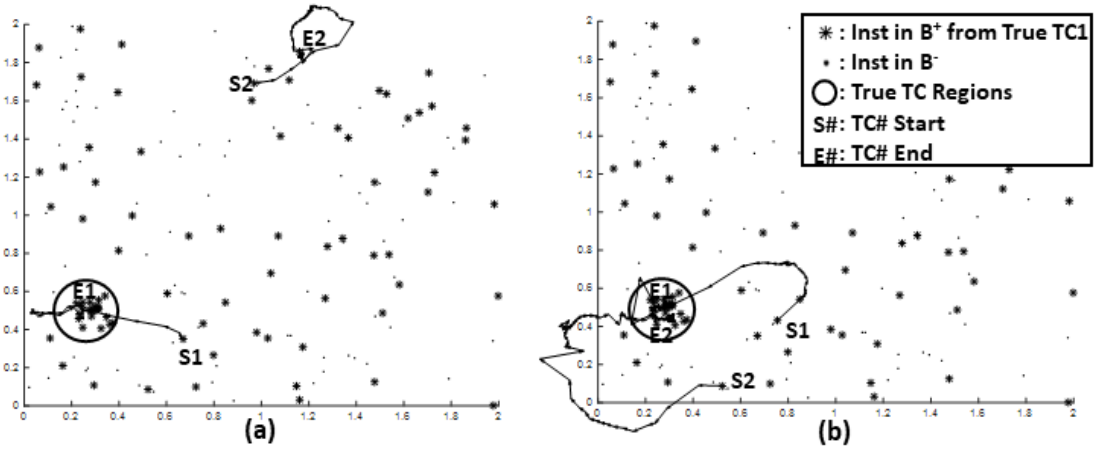


Figure 5.3: Sample runs of the FCMI algorithm on Data-1TC: (a) A run where one TC successfully converges to the true concept. (b) A run where both TCs converge to the true concept.

Table 5.1: Data-1TC Results

	DD (Sc=1)	DD (Sc=2)	CCMI (K=1)	FCMI (K=1)	CCMI (K=2)*	FCMI (K=2)*
TrueC detected	50%	74%	74%	74%	82%	86%
TrueC not detected	50%	26%	26%	26%	18%	14%

*After PCMI, Merging, and Weak TC Removal

The results of the above experiments verify that the proposed CCMI and FCMI are equivalent to the DD when the number of target concepts is set to 1. However, the CCMI and FCMI have the advantage of seeking multiple target concepts simultaneously. Thus, as we have illustrated, even if there is only one true concept, running CCMI or FCMI with $K > 1$ increases the likelihood of identifying the correct region. When paired with the PCMI and selection criteria, we can pinpoint this region and discard the remainder.

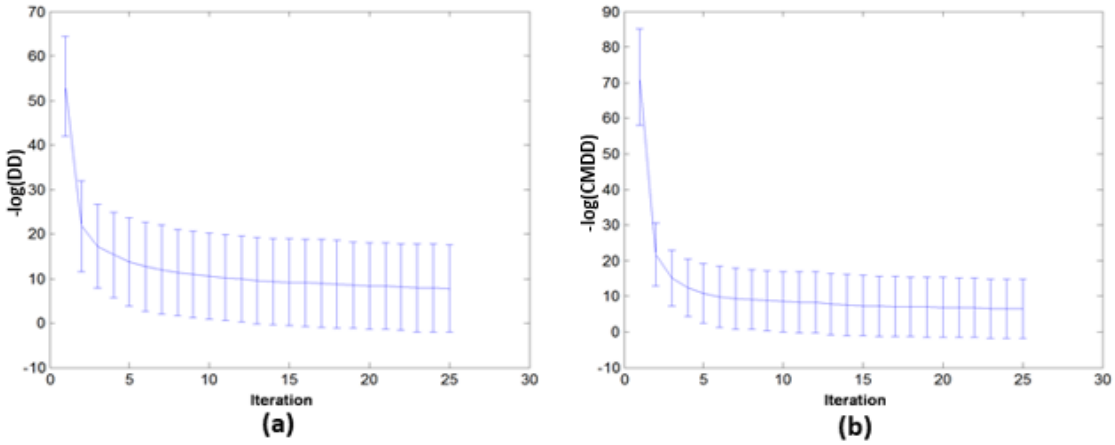


Figure 5.4: Evolution of the Objective Function for The DD and CMDD Algorithm (K=2) across 50 runs. (a) The DD Algorithm mean reaches a stable value after several iterations, but the results are somewhat volatile across runs. (b) The CMDD objective function metric is significantly less volatile in later iterations.

5.1.3 Results and Analysis using Data-2TC

As we outlined in Section 2.2, existing MIL algorithms cannot identify multiple target concepts simultaneously. If the data is expected to include multiple true concepts, the author of [39] recommend two approaches to finding them; the first of these is to utilize a disjoint diverse density function that uses a max function to select the highest bag-to-target concept mapping when computing the diverse density. Beyond the question of whether or not this approach is a logical extension to the DD, the author of [39] acknowledges that it is computationally unfeasible even with data of a moderate size and more than a few true concepts. The second recommendation made is to run the DD from multiple starting points and to select the most common set of concepts learned across the runs. To validate this hypothesis, we performed 50 runs of the DD algorithm on Data-2TC using parameters identical to those utilized with Data-1TC. The results do not support

this hypothesis at all. In fact, none of the 50 runs converged to one of the true concepts. On 62% of the runs, the DD algorithm converged to a point mid-way between true concept 1 and true concept 2. This behavior is depicted in figure 5.5(a). On the remaining 38% of the runs, the algorithm moves to different, but still sub-optimal location, as depicted in figure 5.5(b). Neither of these cases is satisfactory – prompting the need for simultaneous TC discovery such as that permitted by the FCMI and CCMI.

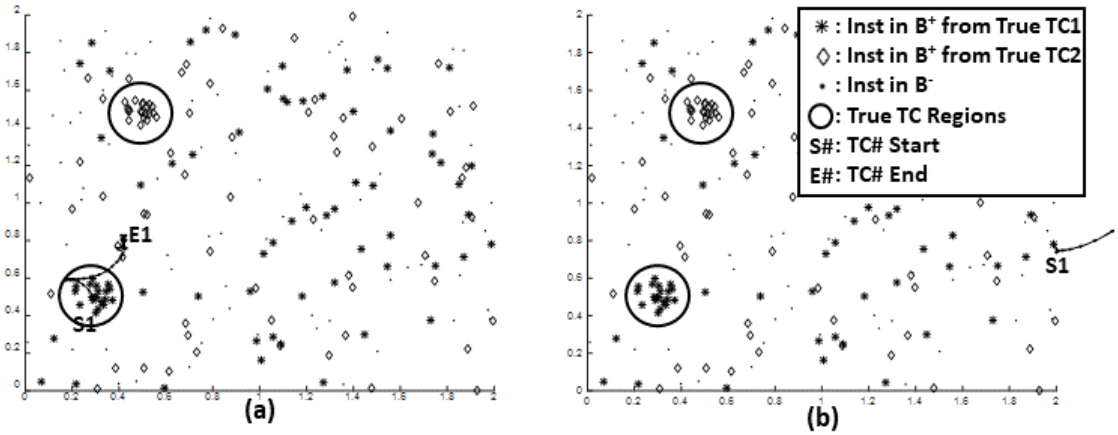


Figure 5.5: Sample runs of the DD algorithm on Data-2TC (a) A sample run that converges a point between the two true concepts. (b) An run in which the algorithm fails to locate either true concept or a mid-point.

To evaluate the CCMI and FCMI algorithms on Data-2TC, we first fixed the number of target concepts, K, to 2, and ran each algorithm 50 times using different initializations and the same parameters as those used for Data-1TC. For both FCMI and CCMI, 46% of the runs correctly located the two true concepts. A single one of the two true concepts is located correctly on 18% of the CCMI runs and 20% of the FCMI runs. In a similar manner to the DD algorithm, the CCMI and FCMI located a point between the two true concepts at a rate of 18% and

22% respectively. The remaining scenario – in which neither true concept nor mid-point is located, occurs on 18% of the CCMI runs and 12% of the FCMI runs. All four of these scenarios are illustrated in figure 5.6. Figure 5.8(a) depicts the objective function for the CCMI over the first 25 iterations of the algorithm. Again, we note the relatively smooth decrease in the objective function over time.

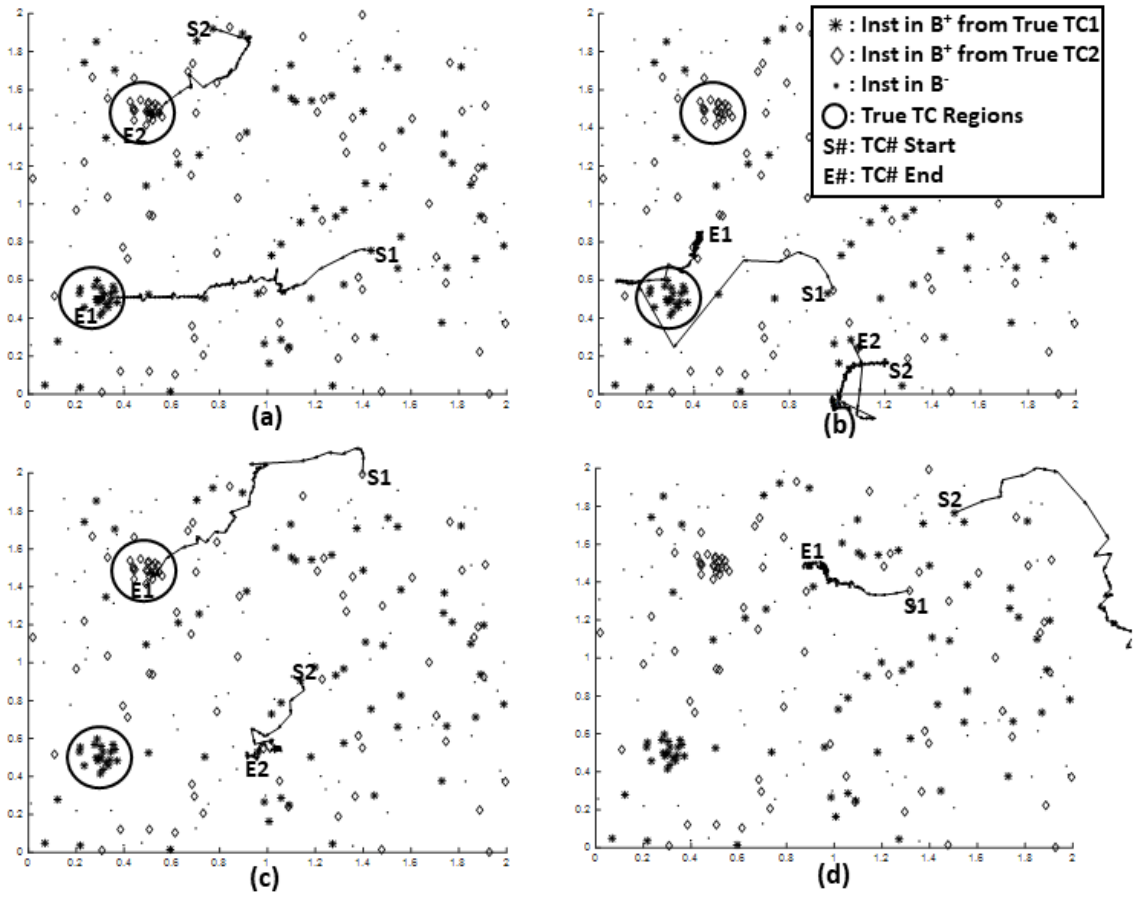


Figure 5.6: Sample runs of the FCMI algorithm on Data-2TC. (a) The two TCs correctly locate the true concepts. (b) One of the two TCs locates a mid-point between the two TCs while the other converges to a sub-optimal location. (c) One of the two true concepts is located by a TC. (d) Neither true concept nor the mid-point is located.

As with the Data-1TC dataset, we sought to improve the performance of our attempts by utilizing additional target concepts. To that end, we repeated 50 runs of the CCMI and FCMI on Data-2TC using K=4 target concepts instead of K=2,

with all other parameters were the same as in our initial experiments. The PCMI merging and selection criteria were also applied in an identical manner. Results were significantly better with this change. The CCMI saw convergence to both true concepts at a rate of 84%, while the FCMI saw convergence to both true concepts at a rate of 88%. Only on a single run of the CCMI did the merging and selection fail to prune the number of TCs below 3, while the remaining runs (14% for the CCMI and 12% for the FCMI) locate only one of the two true concepts. The scenarios encountered when utilizing 4 TCs and the CCMI or FCMI are illustrated in figure 5.7. Figure 5.8(b) depicts the objective function for the CCMI over the first 25 iterations of the algorithm with 4 targets. As with the Data-1TC objective functions, we note that the volatility of the objective function is significantly decreased when a larger value for K is utilized.

While these results are a substantial improvement over those obtained with $K = 2$ target concepts, it was conjectured that changing the initialization used could provide even more consistent results. This belief arose due primarily to two observations made during earlier experiments on Data-1TC and Data-2TC. First, we noticed that initial centroids highly distant from the true concept(s) were significantly more likely to converge to local minima during optimization. Second, we noticed with the standard DD algorithm that initial scales played a major role in the algorithm's ability to converge to the correct region. Based on these observations, the goal was set to find an alternative form of initialization that would select centroids more reliably near the true concept, along with scales that better fit the distribution of positive instances in the dataset.

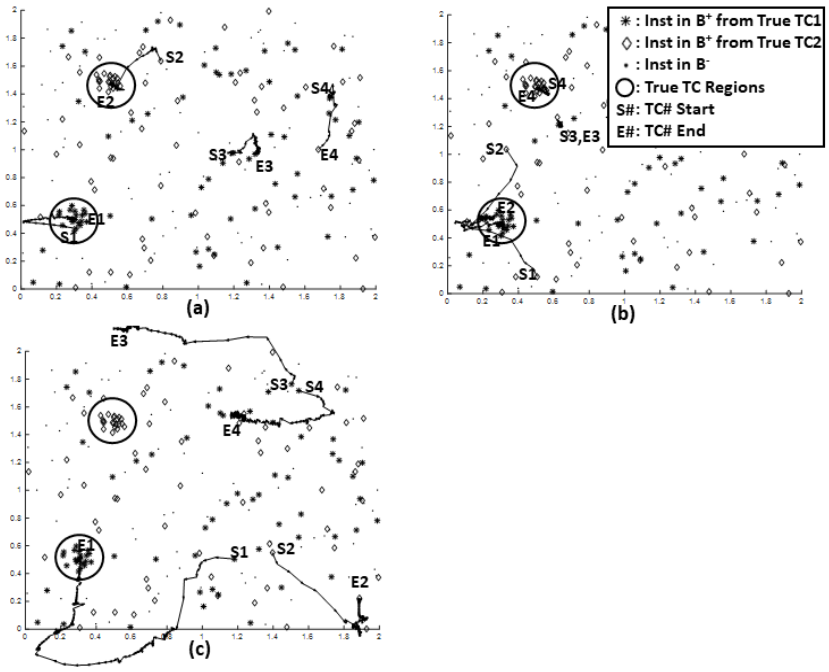


Figure 5.7: Sample runs of the FCMI algorithm on Data-2TC with 4 TC. (a) Two TCs correctly locate the true concepts, while the others are eliminated. (b) Two TCs located a true concept and are merged, while a third locates the second true concept, and another is eliminated. (c) Only one of the true concepts is located.

The alternative initialization we forumlated rests highly in Kernel Density Estimation (KDE) theory [47]. In short, KDE relies on the aggregation of several simple kernel functions to approximate a more complex probability distribution. For our purposes, application of KDE is impractical for the positive distribution due to the ambiguity of which instance(s) in each positive bag are resposible for its label. On the other hand, we do know that every instance in every negative bag is definitively negative. Hence, we can apply a KDE model to the negative instances in the dataset to approximate the distribution of negative data. In turn, we can use simple dissimilarity metrics to select the instance in each positive bag that are most distinct from the negative density. These instances provide a reasonable, albeit imperfect estimate for the true positive instances in the dataset. We then

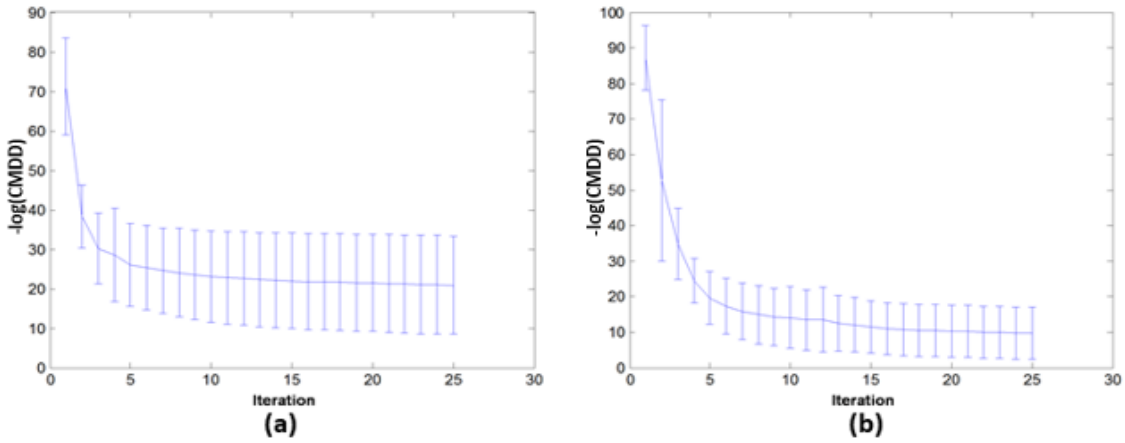


Figure 5.8: Evolution of the Objective Function for The CMDD Algorithm with K=2 and K=4 TCs. We note that (b) CMDD with K=4 TCs reaches a level of significantly less volatility in the objective function across runs than (a) CMDD with K=2 TCs, despite similar mean values.

cluster these "likely positive" instances and use their centroids and standard deviation values as starting points for the CCMI/FCMI algorithm. These steps are summarized below.

1. All instances from all negative bags were clustered using a simple K-means approach into 100 distinct clusters. Centroids (based on cluster means) and scales (based on cluster standard deviation) were computed for each cluster.

2. For each positive bag, a kernel density estimate was generated for every negative cluster. The instance most distant from its closest negative neighbor was estimated as the most likely positive instance, and added to a pool of "likely positive" instances.

3. All "likely positive" instances were clustered using the same K-means approach (given a selected number of target concepts), and from these clusters a set of initial TC centroids (based on the cluster means) and scales (based

on the cluster standard deviations) are generated.

We ran the CCMI and FCMI again with $K = 2$ target concepts and this new KDE initialization. Results proved significantly more reliable than even those using a larger number for K and the merge/selection method. For all 50 runs [1] of both algorithms, the two target concepts converged to the true concept locations. Table 5.2 summarizes the results of the KDE initialization and other experiments involving Data-2TC.

Table 5.2: Data-2TC Results

.;

	DD	CCMI (K=2)	FCMI (K=2)	CCMI (K=4)*	FCMI (K=4)*	CCMI (K=2) KDE	FCMI (K=2) KDE
2 Correct TC	N/A	46%	46%	84%	88%	100%	100%
1 Correct TC	0%	18%	20%	14%	12%	0%	0%
2 TC Midpoint	62%	18%	22%	0%	0%	0%	0%
All TC Fail	38%	18%	12%	0%	0%	0%	0%
Select Fail	N/A	N/A	N/A	2%	0%	N/A	N/A

*After PCMI, Merging, and Weak TC Removal

In this section, we used two simple datasets to illustrate the behavior of the proposed algorithms and compare them to the DD algorithm. In the next section,

[1] We note that instead of 50 different initial TC centroids, we randomize the initial partitions for the K-means clustering of negative and "likely positive" instances. Fewer than ten unique configurations for TC centroids and scales actually appeared given the 50 initial partitions.

we use multiple synthetic datasets with various levels of difficulty to analyze the sensitivity of the proposed algorithm to various parameters.

5.2 Sensitivity of the Proposed Algorithm to Various Parameters

5.2.1 Data Generation

The second set of synthetic datasets were generated with two goals in mind. First, we wanted to generate a large number of sets with varying parameters to test the robustness of our approaches in addressing specific distribution shapes and imbalances. Secondly, we wanted to better emulate often noisy or imperfectly partitioned real world data by allowing a small level of overlap between positive and negative classes. To that end, we first describe our general generation process for the data, and then describe the parameters we vary in constructing the datatsets.

A total of 1100 datasets were generated, with 11 parameters to vary, 10 distinct values per parameter, and 10 randomly generated datasets per combination. In every case, the parameter in question is chosen from 10 uniformly spaced values (e.g. when varying the number of negative bags, for example, we generate from 100 to 1000 bags, with an interval of 100 bags between at each step.) In most cases, all but the specific parameter being tweaked is fixed for dataset generation.

Except when otherwise specified, the following strategy was adhered to in generating each dataset (values with an asterix will vary only according to the

parameter experiment in question):

1. For each of 2* true concepts in the dataset, 200* positive bags are generated within a 2-dimensional* instance feature space.

2. Each true concept has a selected coordinate centroid ((0.5,0.5)* and (1.5.1.5)*) and true scale σ value (0.01,0.01)* – the latter of which controls distribution shape in each dimension.

3. All bags were generated with a random number of instances from 2 up to 20*.

4. The first (true positive) instance in each positive bag is generated from a Gaussian distribution based on the concept's centroid and σ values.

5. All positive remaining instances were generated randomly within the interval ((0,0) to (2,2)) subject to the constraint that they cannot fall within the 90% confidence interval of a gaussian distribution for any other true concept for the dataset – again based on each one's associated centroid and σ values.

6. 200* negative bags are generated for each dataset, again with 2 to 20* instances per bag.

7. All negative instances were generated randomly within the interval ((0,0) to (2,2)), subject to the constraint none can fall within the 90% confidence interval of a normal data distribution to any true concept based on its true concept centroids and σ values.

The use of a confidence-interval controlled distribution in the procedure above allows for some overlap between positive classes and the negative class, requiring that algorithms used to address the generated datasets will require some robustness to noise. Having described the general procedure by which data are generated, we now turn to the manner in which parameters vary for the assessment of our algorithms. Below we examine some fundamental questions to ask of our algorithms' performance, and break apart the dataset experiments into 3 distinct categories that address these questions, as well as provide specific details with respect to the 11 specific parameters we test. As a reminder, the fixed parameter values outlined above are used in generating every dataset unless otherwise noted.

As mentioned, the defined categories cover three distinct tests for our algorithm. The first set of experiments (positive and negative discrepancy) should help test whether our approach is susceptible to bias in one class when there is a substantial bias in quantity of data samples. The second set of experiments (distribution shape) should determine the extent to which the target concept centroid and scales we learn during optimization can appopriately adjust to larger or irregularly shaped data distributions. The third category (dimensionality) datasets examine whether increasing the data dimensions or number of distributions that generate the data lead to side effects that hamper performance (e.g. the curse of dimensionality, etc.)

1. Bag Quantity Parameters: Our first performance question relates to whether or not our approach is susceptible to bias in one class or another when the number of samples varies between them. To that end, we varied the propor-

tion of positive bags (in either or both concepts), as well as the proportion of negative bags.

(a) Positive Bag Quantity: These datasets vary the number of positive bags generated for both true concepts from a value of 100 to 1000.

(b) Unbalanced Positive Bag Quantity: These datasets vary the number of positive bags generated for the first true concept (with centroid at (0.5,0.5)) from a value of 100 to 1000.

(c) Negative Bag Quantity: The Negative Bag Quantity datsets vary the number of negative bags generated from 100 to 1000.

2. Positive Instance Distribution Parameters: Our second performance question addresses the extent to which true concept distribution shape and proximity negatively impact our algorithms' ability to locate them and define an appropriate set of scales to model the true concept distributions. The parameters varied to this end are the relative distance between the true concept centroids, along with the σ parameters that control dispersion in each dimension during generation.

(a) Target Concept Radius: The Target Concept datasets vary the σ value (in both dimensions) for both true concepts in the dataset from (0.0025,0.0025) to (0.025,0.025).

(b) Unbalanced Target Concept Radius: These datasets vary the σ value (in both dimensions) for the first true concept (centroid (0.5, 0.5)) in the dataset from (0.0025,0.0025) to (0.025,0.025).

(c) Target Concept Ellipse Single: These datasets vary σ for the first true concept (centroid (0.5, 0.5)) in the dataset from (0.005, 0.01) to (0.05,0.01). Larger values for σ is this case imply an elliptically shaped distribution with a larger degree of eccentrity.

(d) Target Concept Ellipse Double: These datasets vary the first dimension of σ for the first true concept and second dimension of σ for the second true concept. Specifically, σ values for (centroid (0.5, 0.5)) vary from (0.005, 0.01) to (0.05,0.01), while σ values for centroid (1.5, 1.5) vary from (0.01, 0.005) to (0.01, 0.05) for the same datasets, respectively.

(e) TargetConceptDistance: These datasets vary the distance between the true concepts in each feature dimension from 0.1 to 1. For the former generation the true centroids are at positions (0.95,0.95) and (1.05, 1.05), while the latter true centroids are at positions (0.5,0.5) and (1.5,1.5) (the default values).

3. Data Dimensionality Parameters: Our third performance concern relates to issues of dimensionality. More specifically, do additional true concept distributions, number of instances per bag, or feature space dimensions result present side effects that hamper performance of our algorithms (e.g. the curse of dimensionality)? To that end, we varied the above three parameters.

(a) Target Concept Quantity: The Target Concept Quantity datasets vary the number of true concepts from 1 to 10. σ values for these true concepts are as default, while new centroids are appended as each ad-

ditional target concept is added.

(b) Max Instance Quantity: These datasets vary the maximum number of instances in each bag from 10 to 100. The minimum number remains 2 in each case.

(c) Feature Space Dimension: These datasets vary the dimensionality of the instance feature space from 1 to 10. Additional dimensions for the true concepts and σ take uniform values in accordance with the default, 2-d dataset. For example, the third Feature Space Dimension dataset has 2 true concepts with centroids (0.5,0.5,0.5) and (0.5,0.5,0.5) and σ values of (0.01, 0.01, 0.01), and instances generated in the bounds ((0,0,0) to (2,2,2)). In addition to dimension changes, we also vary the number of positive bags per true concept from 100 to 1000 to counteract the sparsity of the feature space as the dimensionality increases.

5.2.2 Sensitivity Data Algorithm Setup

With our datasets defined, we now turn to the setup we use for our algorithms. Because we found FCMI to outperform the CCMI slightly with our first set of synthetic data experiments, we chose to utilize the former in performing these experiments. In all cases, we rely on the assumption that we have an equal number of target concepts to the true concepts present (2 in all cases except for the "TargetConceptQuantity" datasets).

Operation of our FCMI algorithm is relatively straightforward, following the KDE initialization and parameters we used for Data-2TC in Section 5.1.3, with

the exception that we utilize 500 outer-loop iterations to be certain the algorithm converges to a stable solution.

5.2.3 Sensitivity Data Performance Measures

We use three measures to assess the validity of the clustering results and their deviation from the true partitions. The first of these, "centroid error" measures the error of final centroid locations relative to the true concept centroids used in generation of the data. The second of these, "class overlap" measures the overlap between positive and negative bag probabilities across the target concepts. Hence, it roughly represents the ability of the learned target concepts to distinguish between positive and negative bags. The last of these, "cluster purity" measures the degree to which our optimized target concepts incorrectly mix bags generated from different true concept distributions.

The first metric, "centroid error" is simple to calculate. For each dataset, we compute the mean error of each target concept and its closest true concept, or $\sum_{k=1}^{K} ||c_{k,TRUE} - c_{k,TC}||$, where $||c_{k,TRUE} - c_{k,TC}||$ is the Euclidean distance between the k^{th} true concept and its nearest target concept.

Our second metric, "class overlap" requires a bit more consideration. While the most-likely cause estimate (MLCE) provides a reasonable means by which to judge a bag's probability in each target concept (4.1.2), the question becomes how to compare the multiple probabilities for positive and negative bags across multiple target concepts. We choose to rely on a "max bag probability" approach and histogram overlap scheme with the following steps:

1. For each of our K target concepts, we compute a set of probabilities for all N bags $\{Pr(t_k|B_1) \cdots Pr(t_k|B_N)\}$ using (4.1.2)

2. We compute a single "max bag probability" value for each bag as

$$MaxBP(B_n, \mathcal{T}) = \max_{k=1}^{K} Pr(t_k|B_n) \tag{5.2.1}$$

3. We then construct a cumulative probability density histogram for the positive bags in the dataset, and overlay it with a cumulative inverse probability density histogram for the negative bags in the dataset. The maximum overlap between these two functions becomes the reported class overlap error.

Figure 5.9 depicts a sample set of cumulative probability density histograms computed for the first generated dataset for the smallest parameter value in the Positive Bag Quantity experiment (i.e. 100 positive bags per true concept). As this is a relatively simple dataset (no complex shape, extra TC or dimensions, etc.), the class overlap (region indicated by an arrow) is relatively small.

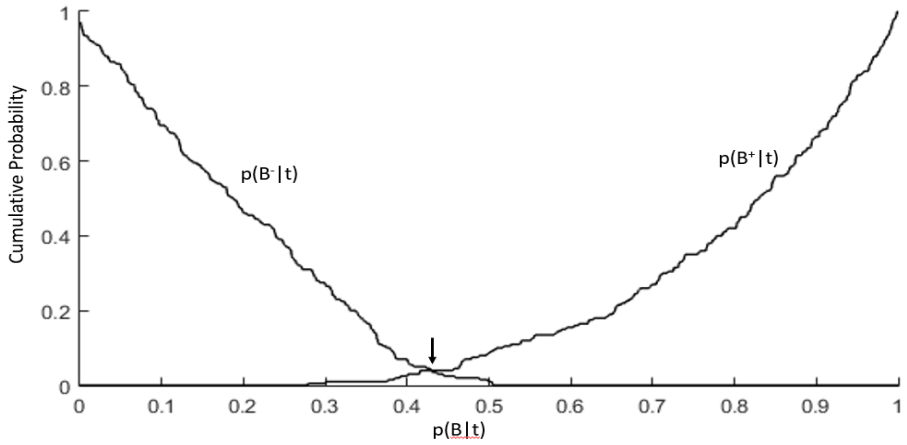


Figure 5.9: Positive Bag Quantity DS-100 Cumulative Probability Class Overlap.

The last metric by which we assess performance, "cluster purity" is comparatively simple. We assign each positive bag a cluster according to the target concept in which it has the highest bag probability. We then compare this set of computed partitions with a ground truth one generated with the IDs of true concepts from which each bag was generated using the RAND index [42].

As noted, we generate 10 random datasets per parameter and parameter value combination. The measures outlined above are computed for each of these datasets, and aggregated to produce an error-bar plot for each parameter. These plots give us a strong idea of how our three metrics change as a given parameter increases or decreases. We note that in the case of the first measure (centroid error), assessment of algorithm performance is straightforward. While some small centroid error is expected for any optimization, a noticeable trend or jump as a parameter is changed requires scrutiny. However, the second measure (class overlap) and third measure (cluster purity), are less straightforward. Because we permit a level of overlap in data generation, a level of confusion between positive and negative bags is expected. And as parameters controlling the shape of the generative concepts changes, we likewise expect bags to be assigned to concepts other than those from which they were generated. In light of these observations, we generate a set of estimated "true scales" for each target concept to complement the true centroids (in this case, each estimated scale is directly proportional to the σ used in data generation). We then generate class overlap and cluster purity metrics using the true centroids and scales, and plot them alongside the optimized values to provide a reasonable performance baseline.

5.2.4 Sensitivity Data Results

Below we present our findings and their evaluation for the Sensitivity Data.

1. Bag Quantity Parameter Results

   (a) Positive Bag Quantity Results: The solid red line in figure 5.10(a) indicates the mean centroid error averaged across the 10 runs of the FCMI for each option of the number of positive bags per true concept, while the error-bars indicate the standard deviation across the same runs. These values demonstrates that regardless of the number of bags per target concepts, the difference between the derived and true concept centroids is quite small (no more than 0.03 on average per parameter choice). For figure 5.10(b), the solid red line and bars are indicative of mean and standard deviation for positive-negative bag overlap using the FCMI, while the dashed blue line indicates the same measures calculated using true concept estimates for centroids and scales. The fact that this overlap is quite small regardless of bag quantity (approximately 2-3% per run) is an indication that positive bags in the dataset have appropriately higher bag probabilities than the negative bags, regardless of positive bag quantity. In addition, the fact there is little difference between the overlap measures when using using the true concepts versus the FCMI is a strong indication that our algorithm is not introducing outside error. Figure 5.10(c) details the RAND index of clusters when calculated using the approach described in Section 5.2.2,

again with the solid red line and bars indicating FCMI measure, and the dashed blue line and bars indicating results with true concept assignment. These values are similar to each other across parameter selection and close to 1 across the board, implying that FCMI-derived TCs can appropriately distinguish between bags generated by the first and second true concept regardless of the number of positive bags present. We note that the small amount of class overlap present and slight fluctuations in RAND index common to both the FCMI and true concept computations are expected due to our tolerance for overlap in the process of generating the data.

(b) Unbalanced Positive Bag Quantity Results: Figure 5.11 depicts error measures when the number of bags generated by only a single true concept are varied. The small amount of centroid error, small bag probability overlap, and high RAND index are similar to those observed when the quantity of bags varies for both true concepts. Coupled with the similar performance between FCMI and true concept calculations, it can be reasonably concluded that our algorithm can still locate target concepts with a disproportionately small number of bags relative to another.

(c) Negative Bag Quantity Results: Figure 5.12 depicts error measures when the number of negative bags is varied. As with the results for parameters related to positive bag quantity, true concept results and FCMI results are extremely similar, and none of our error metrics is

unexpectedly high or low. This indicates that the FCMI can tolerate a substantially unbalanced ratio of negative to positive bags in the dataset.

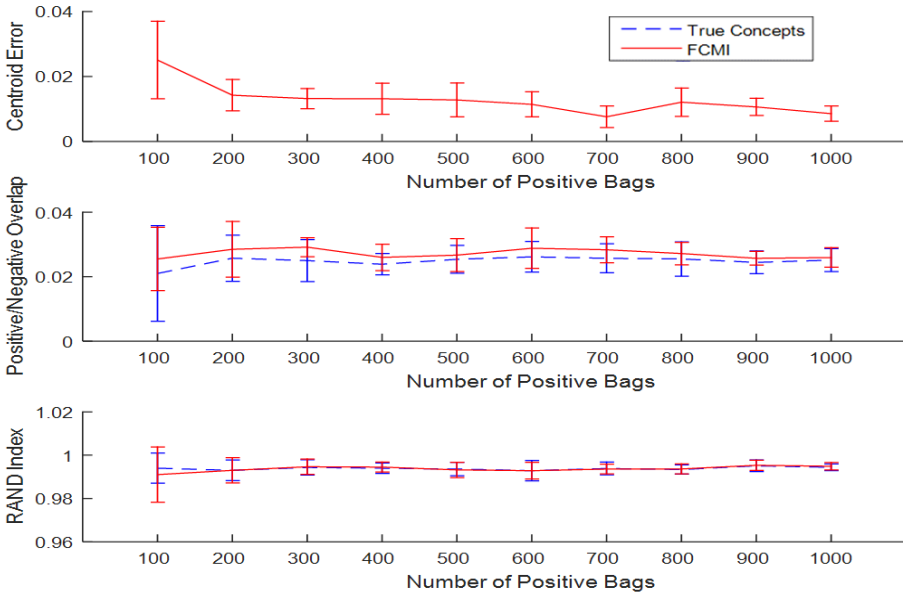


Figure 5.10: Positive Bag Quantity Performance: (a) The number of positive bags does not appear to heavily influence the centroid error between target concepts and true concepts. (b) The number of positive bags does not appear to heavily influence the overlap between positive bag probabilities and negative bag probabilities in the dataset. (c) The number of positive bags does not influence the RAND index of our assigned clustering.

2. Positive Instance Distribution Parameter Results

(a) Target Concept Radius Results: As figure 5.13 depicts, changing the σ used in true concept bag generation does have a noticeable impact on performance measures – unlike parameters from our prior category. Figure 5.13(a) indicates a small but perceptible climb in centroid error as the value of σ increases. Figure 5.13(b) likewise demonstrates larger overlap between the classes as σ increases, and figure 5.13(c) indicates a

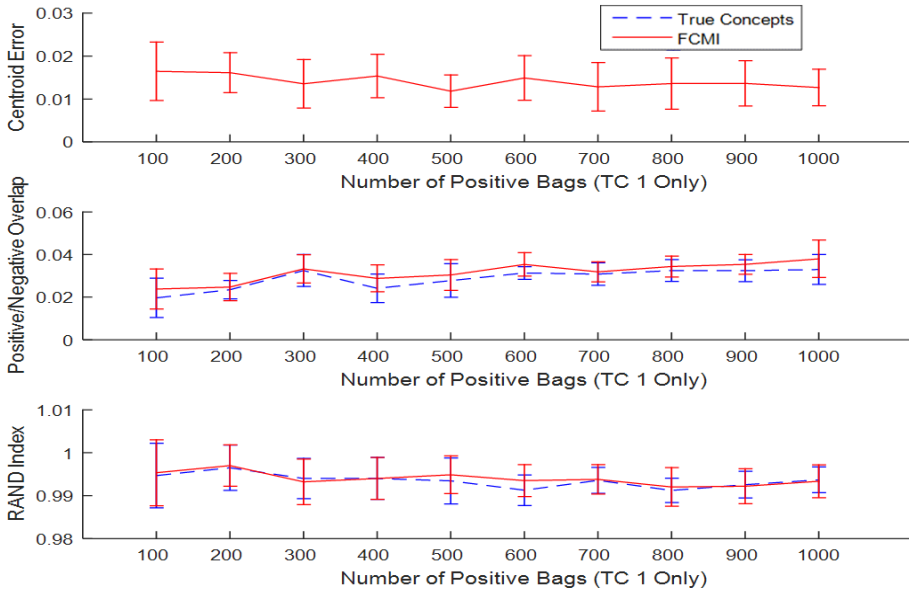


Figure 5.11: Unbalanced Positive Bag Quantity Performance: (a)(b)(c) Our error measures are unaffected by having a disproportionate number of positive bags generated for one true concept.

slight trend down in the RAND index as σ increases. These phenomena are all related – as the dispersion of true positive instances from the centroid increases, a larger proportion of negative instances or instances generated from other true concepts will appear in overlapping spaces between the outlying true positive instances. The key observation to make here, however, is that measures computed using the true centroids and scales do not substantially underperform those computed using the FCMI generated TCs – indicating a level of adaptability to simple changes in true concept shape.

(b) Unbalanced Target Concept Radius Results: The performance measures assessed for the datasets where only a single true concept's σ varies are depicted in figure 5.14. The trends here are virtually identi-

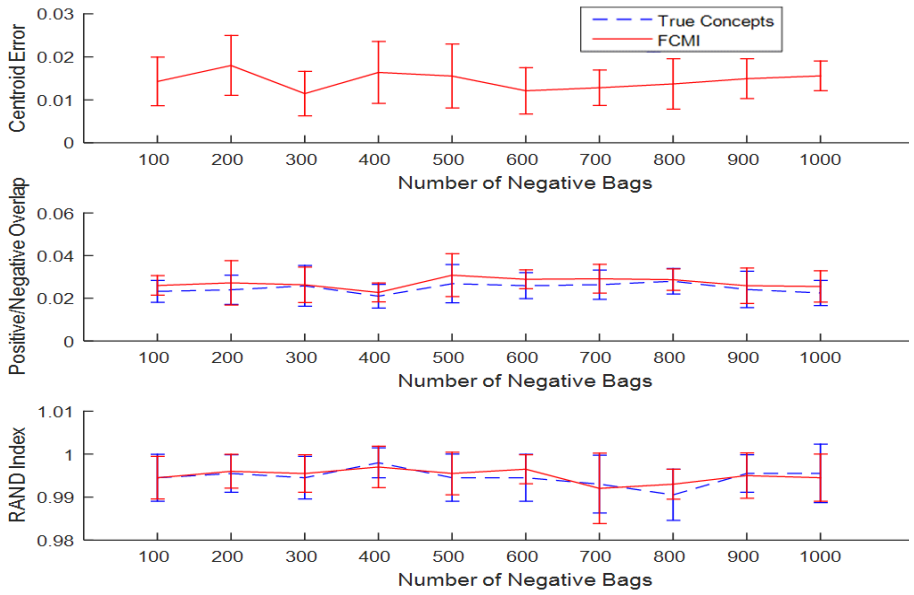


Figure 5.12: Negative Bag Quantity Performance: : (a)(b)(c) The number of negative bags generated does not appreciably impact our observed error measures.

cal as those observed for data with two varying true concept σ values – specifically, a gradual increase in the centroid error and overlap, as well as gradual decrease in RAND index, as σ increases. The FCMI's error measures are again in line with those computed used the true centroid and scale, indicating appropriate, variable scales can be learned simultaneously using our algorithm.

(c) Target Concept Ellipse Single Results: Figure 5.15(a) depicts the mean centroid error for the first elliptical true concept distribution. These results demonstrate only a small change to centroid error as σ increases. Figure 5.15(b) depicts positive and negative probability overlap for the same sets of data. We note here the somewhat surprising observation that the FCMI overlap outperforms the true concept at larger values of σ in terms of overlap. Figure 5.16(c) likewise demonstrates a very

slight performance decline for true concept computation with respect to the RAND index as σ increases. These discrepancies in performance as σ increases are likely an indication that the calculations we used to synthesize "true" scales are flawed, and not an indication that the FCMI actually better models the true concept distributions than the generative parameters themselves.

(d) Target Concept Ellipse Double Results: For the most part, figure 5.16, which depicts results for two elliptical distributions as one of their dimension's σ varies, matches what was observed the Target Concept Ellipse Single datasets. Namely, we see reasonable performance overall by the FCMI with slight discrepancies in true concept-based performance as σ increases – again likely due to an error in computing true scale values. However, we note the extremely large standard deviation when σ is 0.02 for all three error measures, despite having means fairly in-line with the general trend. This anomaly is the result of the fact that on a single run of the FCMI for one of these datasets, neither TC succeeded in locating the true concept – each one instead arrived at a local objective minimum in between the two true concepts.

(e) Target Concept Distance Results: The impact of the centroid distance parameter on performance is among the most nuanced for our datasets. As figures 5.17(a) and 5.17(b) depict, mean centroid error and positive-negative overlap measures are relatively high at a distance of 0.1, but still not at a peak. This is explained by the fact that the generated

distributions virtually merge into a single, almost circular single true concept distribution. As long as one of the true concepts is located, most positive bags from the other true concept stand a good chance of being correctly classified as positive. While this has only moderate impact on the mean centroid and overlap measures, it has devasting impact on the RAND index, as depicted in figure 5.17(c), as it becomes virtually impossible to distinguish between bags generated from one true concept to another. As the distance parameter is increased slightly (e.g. to a value of 0.3), we see less confusion between bags generated by the true concepts as measured by the RAND index, but more overlap between positive and negative bags. This can be attributed to the fact that the shapes of the distributions at this distance no longer resemble a single circular distribution, yet are still in extreme proximity – making KDE-based clustering and FCMI convergence to the true concept (as opposed to a point in between) more difficult. Furthermore, any failure to locate either true concept becomes significantly more costly, as instances at the margins of one of the two distributions may no longer be correctly classified by a single TC. As the parameter increases beyond 0.3, the performance of the FCMI rapidly returns to its default performance with simpler datasets as the distributions become distinct.

3. Data Dimensionality Parameter Results

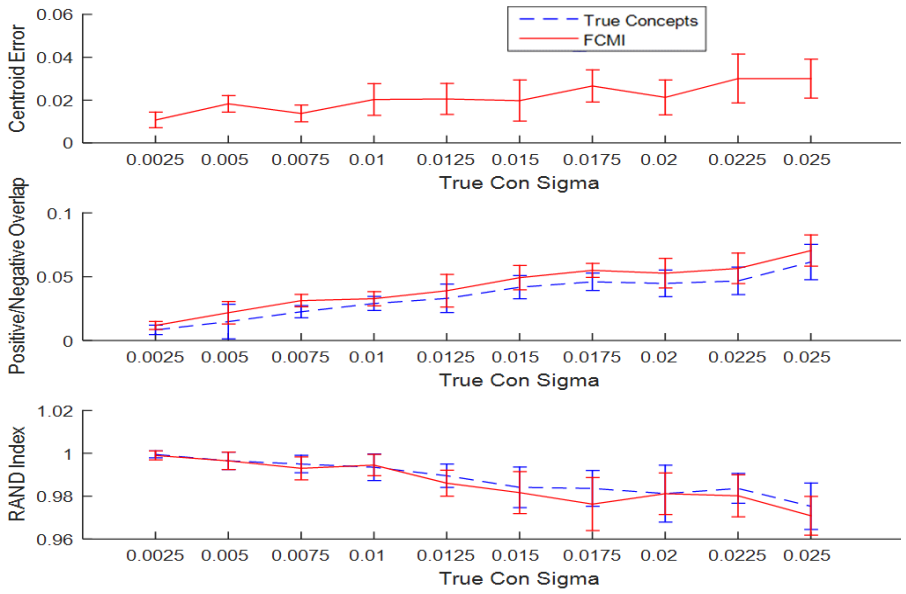


Figure 5.13: Target Concept Radius Performance: Increasing the true concept sigma results in gradually (a) greater centroid error, (b) greater positive-negative overlap, and (c) a decreased RAND index. FCMI performance is in line with the true concept performance.

(a) Target Concept Quantity Results: Figure 5.18(a) depicts mean centroid error as the number of true concepts is varied for each dataset. We note that FCMI error is generally close to 0 for small numbers of true concepts, but then rapidly increases as more than six are utilized. Figures 5.18(b) and 5.18(c) depict similar behavior, with overlap increasing substantially and RAND index falling as more than 6 true concepts are utilized. In comparison, these measures remain close to perfect when using the true concept centroids and scales. It is likely that, as more true concept distributions are introduced in closer proximity within the instance feature space, opportunities for both the KDE initialization and FCMI optimization to fail to locate and distinguish between different true concepts increases substantially. The fact that

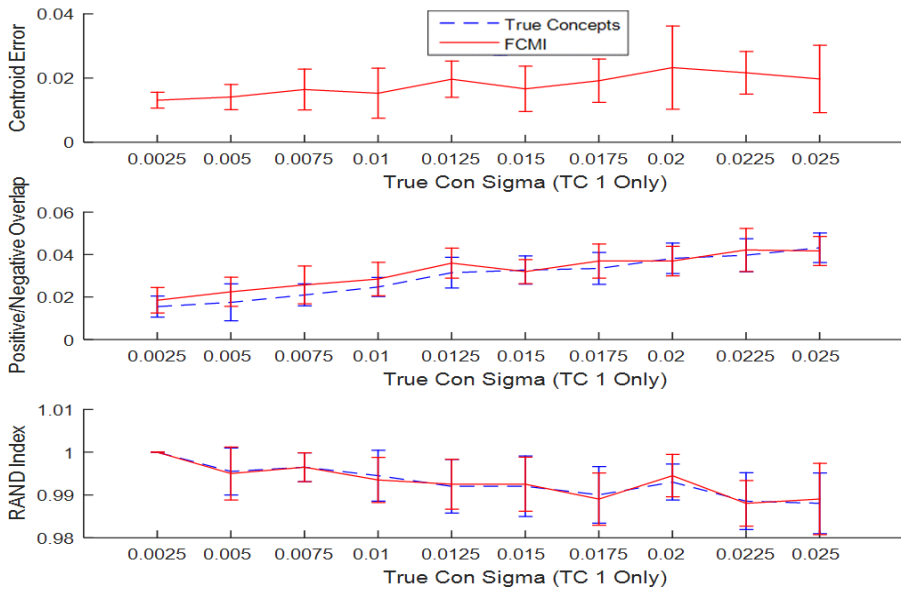


Figure 5.14: Unbalanced Target Concept Radius Performance: Increasing a single true concept's sigma results in gradually (a) greater centroid error, (b) greater positive-negative overlap, and (c) a decreased RAND index. FCMI performance is in line with the true concept performance.

the region for instance generation ((0,0) to (2,2)) does not change as a function of the number of target concepts compounds the potential for dense, locally minimal regions to be located.

(b) Max Instance Quantity Results: Figure 5.19(a) depicts centroid error as the maximum number of instances per bag changes, and is relatively flat, indicating the number of instances does not greatly influence location of the correct true centroids. By contrast, figure 5.19(b) demonstrates different and unanticipated results. Namely, we notice an overall decrease in the overlap between positive and negative bags as the number of instances increases. This observation are best explained through analysis of how we generated the data. Specifically, we permitted secondary instances in positive bags (i.e. not "true instances")

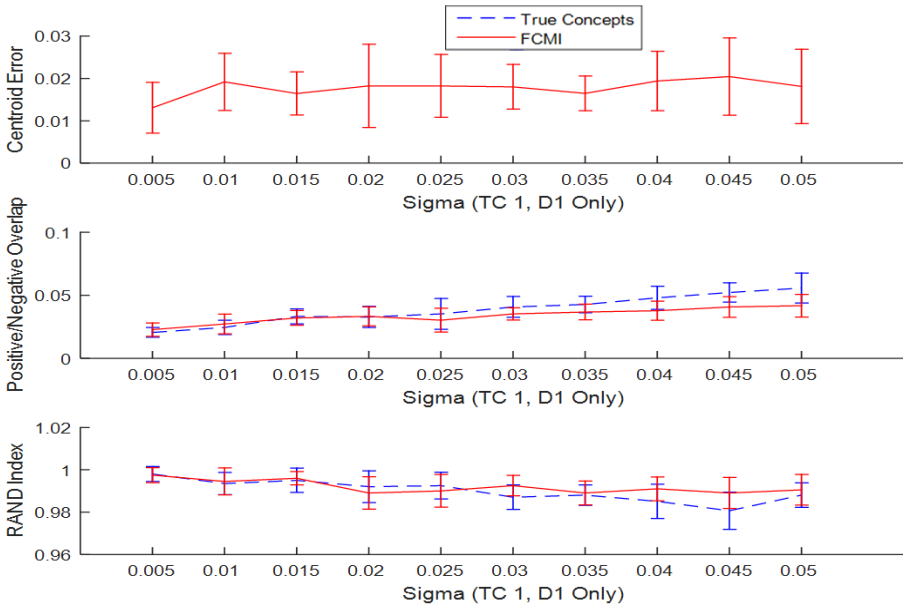


Figure 5.15: Target Concept Ellipse Single Performance: Increasing the eccentricity of a single elliptical true concept gradually (a) greater centroid error, (b) greater positive-negative overlap, and (c) a decreased RAND index. FCMI performance is in line with the true concept performance.

to be generated within the true concept distribution, unlike negative bags and positive bags generated from other true concepts. As a result, increasing the number of maximum instances proportionally increases the likelihood that a bag with a "bad" or outlying true positive instance may have a secondary instance more in line with the distribution.

(c) Feature Space Dimension Results: Figure 5.20(a) depicts centroid error as the number of instance space feature dimensions increases. While its gradual increase alongside an increasing number of dimensions can be explained to an extent by the "curse of dimensionality," the larger issue is the decreasing overlap between classes seen in figure 5.20(b), and the stabilization of the RAND index to a near perfect value with a larger number of dimensions, as seen in figure 5.20(c). This is almost

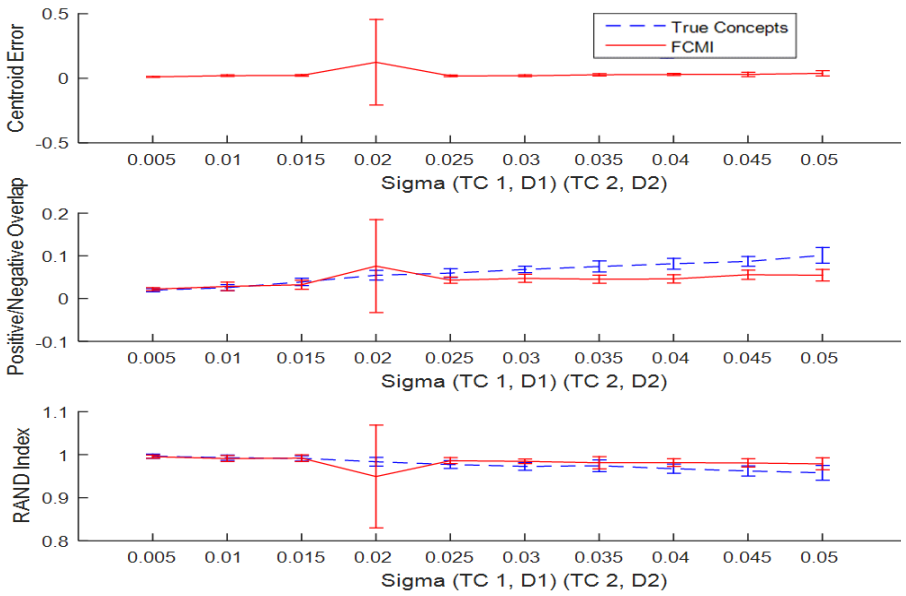


Figure 5.16: Target Concept Ellipse Double Performance: Failure to locate tue concepts can substantially impact overall performance. (a) (b) (c) Standard deviation is substantially increased for all error measures when sigma is 0.02 due to a single failed run.

certainly a side-effect of our data becoming exponentially sparse within the feature space as the number of dimensions increases, and simply multiplying the number of bags in the dataset by proportional values to the dimension parameter did not solve problem. It can likely be concluded that carrying out synthetic experiments based on our generative approach will likely be infeasible – or at least provide misleading analysis, in higher dimensional feature space.

Based on the above results, we can pinpoint several strengths and weaknesses of our approach with respect to parameter changes. For one, the FCMI seems to adjust reasonably well to different quantities of data being utilized at either the bag or instance level (as explained above, the observed changes in performance due to number of instances are a side-effect of the manner in which we generated

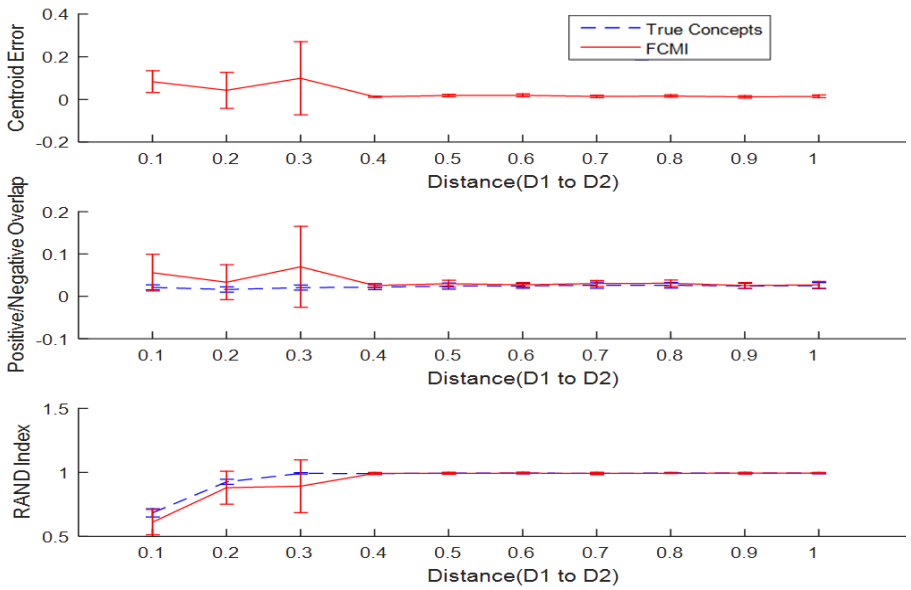


Figure 5.17: Target Concept Distance Performance: (a) (b) Centroid error and overlap between positive and negative probabilities peaks at a distance of 0.3 (c) RAND index bottoms out at a distance of 0.1, when bags from the two true concepts are virtually indistinguishable.

the data), including an unbalanced number of bags generated by one true concept in comparison to another. Similarly, the FCMI seems to respond well to changes in the shape and size of the concept dispersion via the σ parameter, indicating a good set of scales are being learned, alongside the concept centroids.

On the other hand, as evidenced by the experiments varying the distance between target concepts and the number of target concepts, the FCMI appears to have issues with true concept distributions that are close in proximity and/or have heavy overlap.

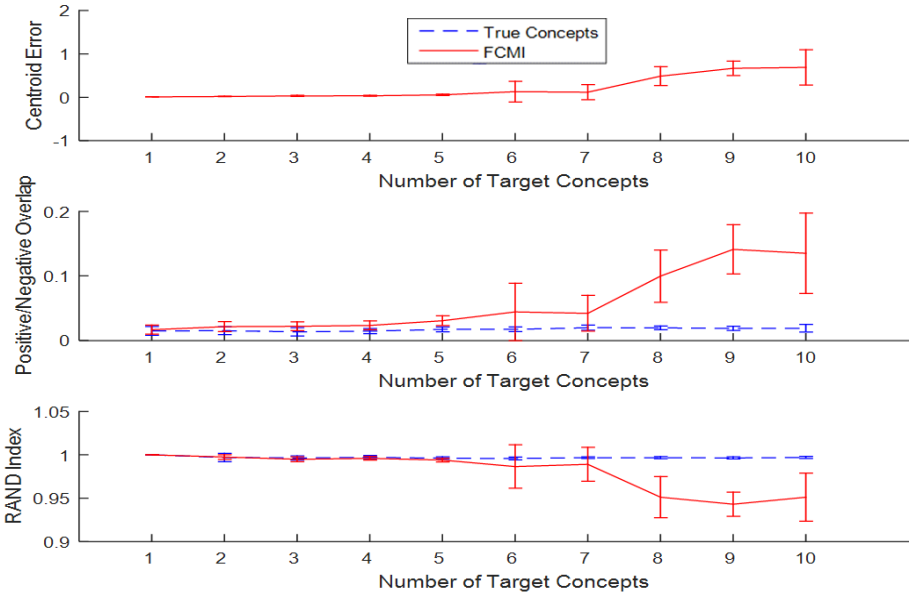


Figure 5.18: Target Concept Quantity Performance: The number of target concepts is largely irrelevant for lower values. However, values above 6 substantially (a) increase mean centroid error (b) increase positive-negative overlap (c) decrease the RAND index.

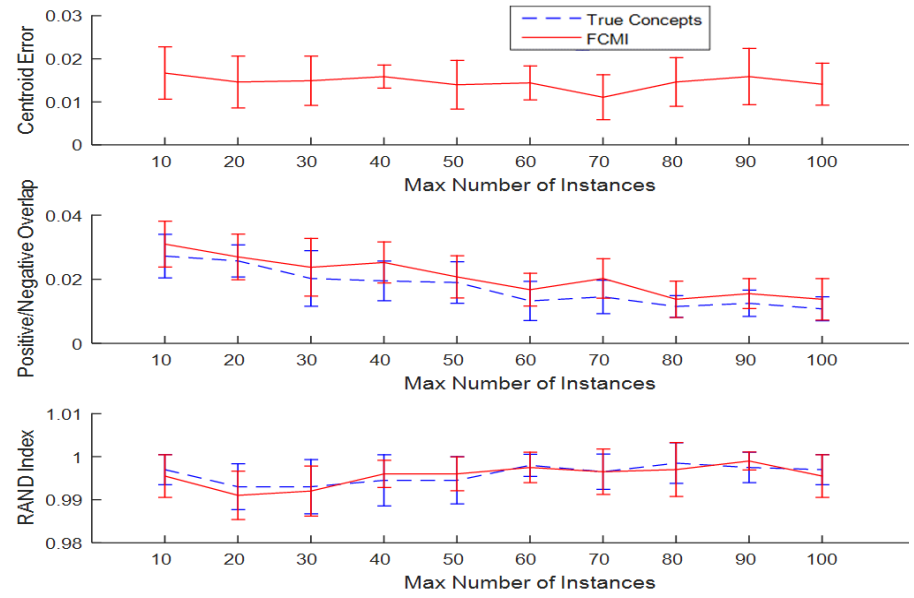


Figure 5.19: Max Instance Quantity Performance: Increasing the maximum number of instances per bag (a) (c) has little impact on centroid error or the RAND index (b) substantially decreases positive-negative overlap.

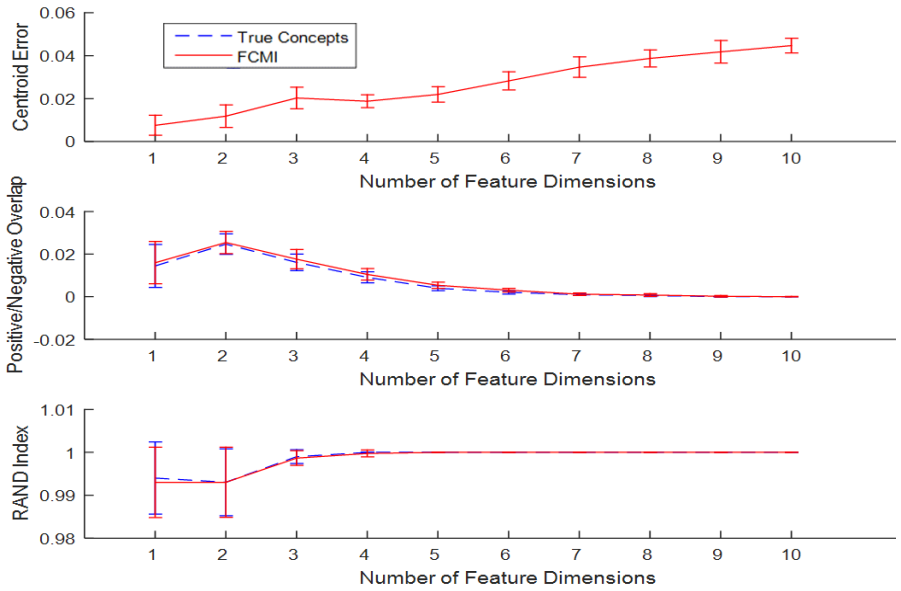


Figure 5.20: Feature Space Dimension: Increasing the feature space dimensionality (b) decreases positive-negative overlap and (c) quickly stabilizes the RAND index to a perfect 1, despite (a) increasing overall centroid error

5.3 Application of FCMI and PCMI to BEO Clustering

5.3.1 BEO Clustering Data Description

Our first real-world data analysis was geared towards knowledge discovery through Multiple Instance Clustering of buried explosive object (BEO) data. These data were collected using a Ground Penetrating Radar (GPR) sensor by a NIITEK vehicle-mounted GPR system [22] from outdoor test lanes at two different locations.

The data collection process works roughly as follows: a vehicle with a sensor array at its head such as the one depicted in Figure 5.21 drives along a road or other path and emits 51 cross-track GPR signals into the ground, each of which returns a seperate *channel* waveform signal of 416 time sampled values. The time sample information is then converted to an estimate of *depth*-based intensity returns for each channel. As the vehicle drives along its path, it continuously collects new information from its sensors along the down-track direction which we refer to as *scans*. The net result of data collection is therefore a $N_d \times N_c \times N_s$ data cube consisting of N_d depth values, N_c channel values, and N_s scan values. Figure 5.22 depicts a sample data-cube collected by the GPR at the location of a BEO.

The raw GPR data collected in the field require a significant amount of preprocessing to clean hardware-generated noise, as well as to properly align the depth-dimension of the data cube. After this initial preprocessing, a computationally inexpensive prescreening step is used to identify segments of the data

Figure 5.21: NIITEK Data Collection Vehicle with Mounted GPR Sensor Array

cube associated with anomalous activity for further processing. The specific pre-screening performed on our data used a simple adaptive least mean squares (LMS) algorithm[51], which ultimately flags individual locations at which buried explosive objects *might* be present. At each such location an "alarm" is generated that consists of a subset of the complete data cube taken at the location (scan) of interest.

Data used in our experiment were collected at test lanes across two locations. The first location was a temperate region with significant rainfall, whereas the second collection was a desert region. The lanes in both locations are simulated roads with known BEO locations. BEOs include conventional Anti-Tank mines (AT) and Anti-Personnel mines (AP), along with Improvised Explosive Devices (IEDs) that consist of either high metal content (classed as M), or low metal content (classed as LM). All BEOs are buried at various depths under the surface.

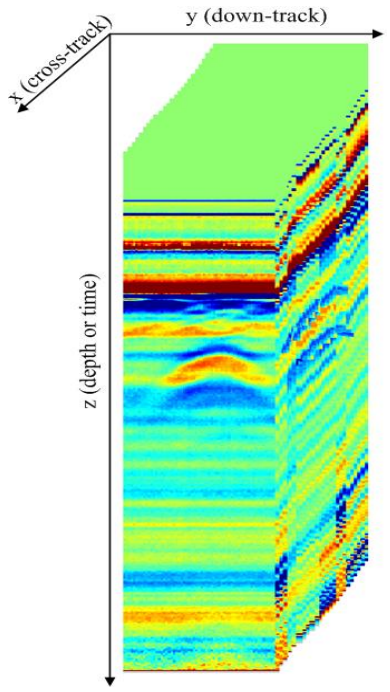
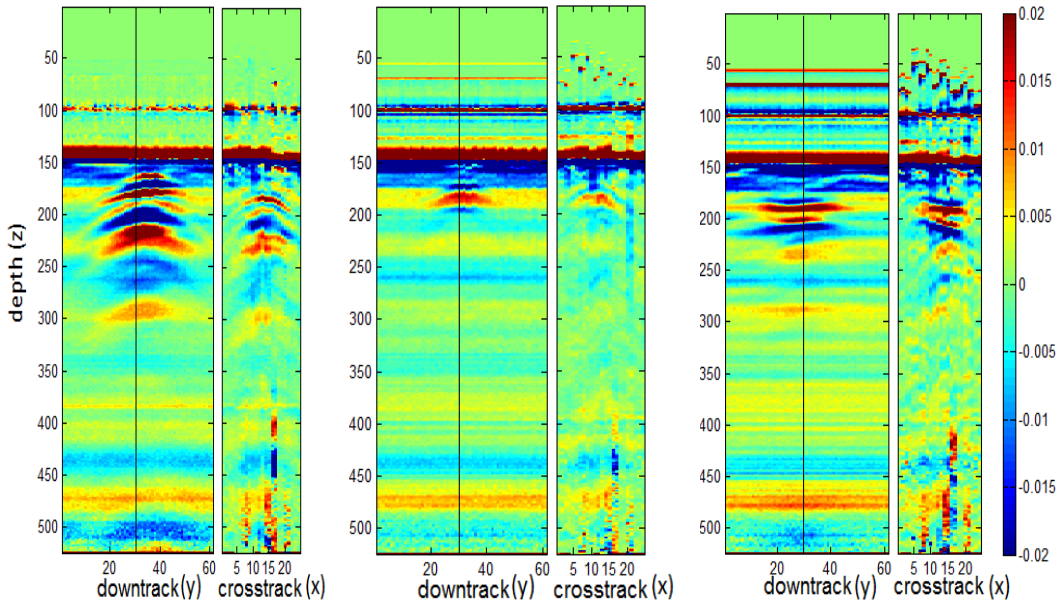


Figure 5.22: Sample GPR-collected data cube

Multiple data collections were performed at each site at different dates resulting in a large and diverse collection of BEO and false alarm signatures. False alarms arise as a result of radar signals that present BEO-like character. Such signals are generally said to be a result of clutter. After the prescreening process, each alarm in the dataset has a corresponding data cube with dimensions representing the depth (500 depth bins), down-track (61 frames or scans), and cross-track (51 channels). Using the ground truth, each sample is labeled as BEO or clutter. The true depth location is unknown. A few sample BEO and clutter alarms are depicted in figure 5.23. For our experiment, we use a subset of the complete data that has 500 BEO samples and 500 clutter samples. For this experiment, the 500 BEO samples were drawn entirely at random, while the clutter samples were selected to exclude stronger candidates (i.e. those with particularly high prescreening values).



(a) High Metal Anti-Tank Mine (b) High Metal Anti-Personnel Mine (c) Clutter Object

Figure 5.23: Information regarding an alarm's signature is present in both downtrack and crosstrack directions.

5.3.2 BEO Feature Extraction

After prescreening the data and obtaining individual alarms, the next step in processing the data is to represent the data in a quantitized form via a feature extraction procedure. For BEO data, this poses a non-trivial problem, because the actual signature of a BEO spans a variable (but typically small) subset of depths within the associated alarm data cube. An illustration of this is shown in Figure 5.24, which depicts slices from two alarms' data cubes taken from the middle channel. The hyperbolic shape for each alarm corresponds to the actual BEO signature within the entire data cube – the remainder of the data cube corresponds to background. Despite the fact that these alarms correspond to the same BEO type, it is clear that the actual depth values corresponding to the BEO signature cannot be reliably predicted. As a consequence, a method that extracts

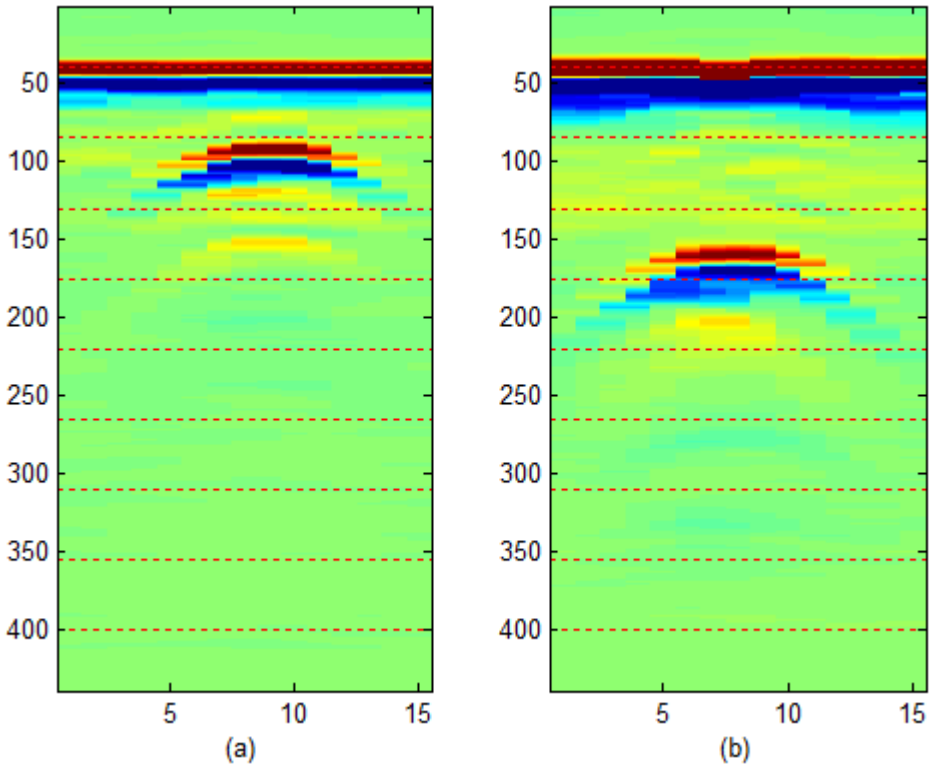


Figure 5.24: BEO Data Representation Issue 1: Two collected positive data samples. The target type is identical for both, but the depth at which the mine signature (hyperbolic shape with high intensity) is present is ambiguous.

a global feature vector representing an individual alarm would be an inadequate choice for machine learning.

One option to dealing with the ambiguous nature of the depth location for the signatures is to frame the BEO problem from a MIL standpoint. In this case, we choose to divide each individual alarm into 15 overlapping windows along the depth. Each individual 60 depth x 61 scan x 51 channel GPR depth-window datacube is a candidate for the actual location of the BEO signature, and thus can be considered an "instance" within the entire alarm data cube, or "bag."

A complicating factor in the application of BEO detection is the diverse set of BEO signatures that may encountered in the field. Different sensors, different soil

types, and most importantly, different types of BEOs, can all lead to vastly different GPR signature being generated during data collection. Figure 5.25 depicts a simple example of this, in which two disparate BEO types possess very distinct signatures. Hence, the BEO detection problems faces not only the problem of ambiguity in true positive depth, but a complicating factor in the form of a large intra-class variation within the correct BEO signature. From the viewpoint of our research, multiple-instance learning approaches to BEO detection that ignore the potential for mulltiple target concepts will be sub-optimal. Given the above complications to the application, it was deemed that the MDD framework would be an ideal one for locating regions within the feature space corresponding to distinct BEO types.

5.3.3 The EHD Feature

As mentioned, the 15 overlapping windows form the set of 15 instances per data sample. For each individual instance, we chose to utilize the Edge Histogram Descriptior (EHD) feature extractor [20] to quantize the GPR data. The EHD is a well-studied technique for representing frequency and direction of intensity changes within an image. The detected edges are pooled into five distinct histogram bins according to their orientation: vertical (V) , horizontal (H), diagonal (D, 45° ascending), and anti-diagonal (A, 45° descending), and isotropic (N, non-edged). As the EHD feature operates by sliding an edge-mask along a 2-dimensional image and vectorizing accumulated results according to the edge orientations, it must be adjusted to function with a 3-dimensional data-cube. We accomplish this by

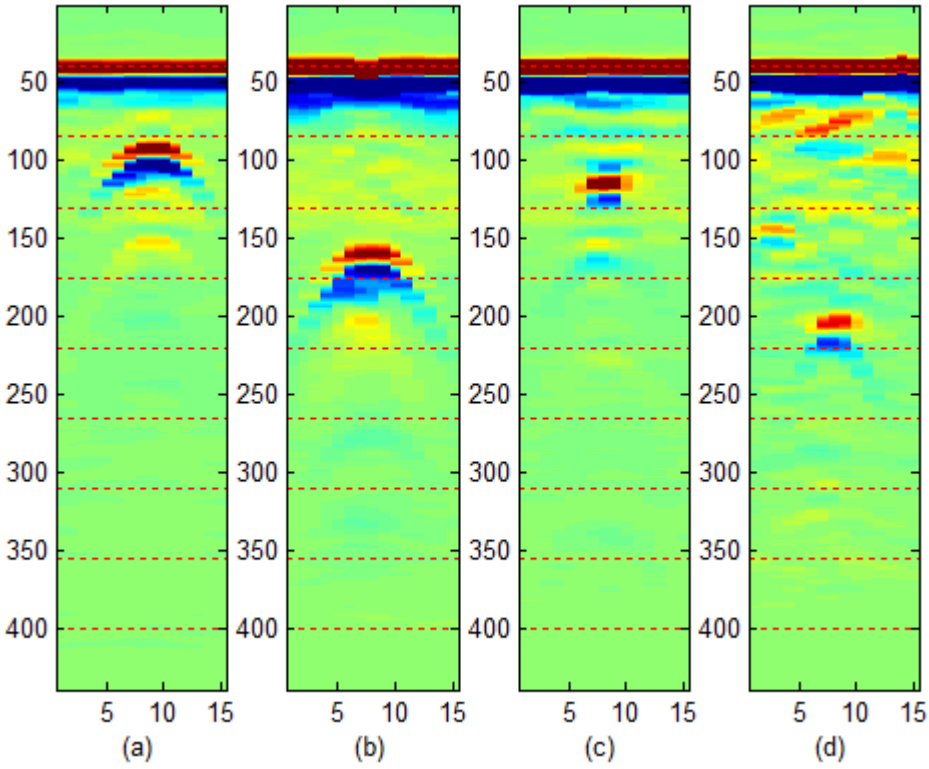


Figure 5.25: BEO Data Representation Issue 2: The presence of diverse target types in BEO data makes the use of a single target concept in MIL sub-optimal. Even with ideal depths selected, features extracted from the BEO signatures in (a) and (b) will be very different from those in (c) and (d).

extracting two smaller sub-cubes from each depth-window data-cube representing both down-track (DT) and cross-track (CT) directions.

For the down-track direction, we utilize a 60 depth by 7 channel by 15 scan sub-cube. For the cross-track direction, we utilize a 60 depth by 15 channel by 7 scan sub-cube. Let each 3-dimensional sub-cube $S(x, y, z)$ be represented by a set of sub-images $S^x_{z,y}, x = 1 \cdots 7$, where z corresponds to the depth dimension, y corresponds to scans or channels for the DT or CT direction respectively, and x corresponds to channels or scans for the DT or CT direction respectively. The above edge-masks are processed for each pixel of sub-image $S^x_{z,y}$ and each is labeled according to the dominant edge (H,V,D,A, or N – if none of the edges exceeds threshold parameter θ_G). The quantized pixel edges are then partitioned across 7 overlapping windows along the sub-images $S^x_{z,y}, x = 1 \cdots 7$, and for each of these partitions we average the accumulated pixel values (across all dimensions) into a five-dimensional partial edge feature $H_{y_i} = [E^H_{y_i}, E^V_{y_i}, E^D_{y_i}, E^A_{y_i}, E^N_{y_i}] for i = 1 \cdots 7$, where $E^H_{y_i}$ corresponds to the average horizontal edge of pixels in the i^{th} partition, and $E^V_{y_i}$ corresponds to the average vertical edge for pixels in partition i, etc. The 35-dimensional EHD feature in either direction is then accumulated as $EHD(x, y, z) = [H_{y_1}, H_{y_2}, \cdots, H_{y_7}]$. As mentioned before, the only differences between EHD calculation for DT and CT features are the dominant plane from which sub-images are extracted (scans for DT, channels for CT).

Several experiments were performed to determine which of the 35-dimensional EHD features are most discriminative between BEO varieties. In the vast majority of experiments, the ones providing the most discrimination were determined to be

the diagonal edges from the first, second, and third window partitions, as well as the anti-diagonal edges from the fifth, sixth, and seventh partitions – regardless of whether the down-track or cross-track direction was being evaluated. We refer to these edges as $D_1^{DT}, D_2^{DT}, D_3^{DT}, A_3^{DT}, A_2^{DT}, A_1^{DT}$ for the down-track direction, and $D_1^{CT}, D_2^{CT}, D_3^{CT}, A_3^{CT}, A_2^{CT}, A_1^{CT}$ for the cross-track direction. We chose to restrict our analysis to this subset of discriminative features, which is ultimately given as $EHD_{cl} = [D_1^{DT}, D_2^{DT}, D_3^{DT}, A_3^{DT}, A_2^{DT}, A_1^{DT}, D_1^{CT}, D_2^{CT}, D_3^{CT}, A_3^{CT}, A_2^{CT}, A_1^{CT}]$. Figure 5.26 provides a visual summary of how these particular features are extracted.

$$EHD_{cl} = D_1^{DT} D_2^{DT} D_3^{DT} A_3^{DT} A_2^{DT} A_1^{DT} D_1^{CT} D_2^{CT} D_3^{CT} A_3^{CT} A_2^{CT} A_1^{CT}$$

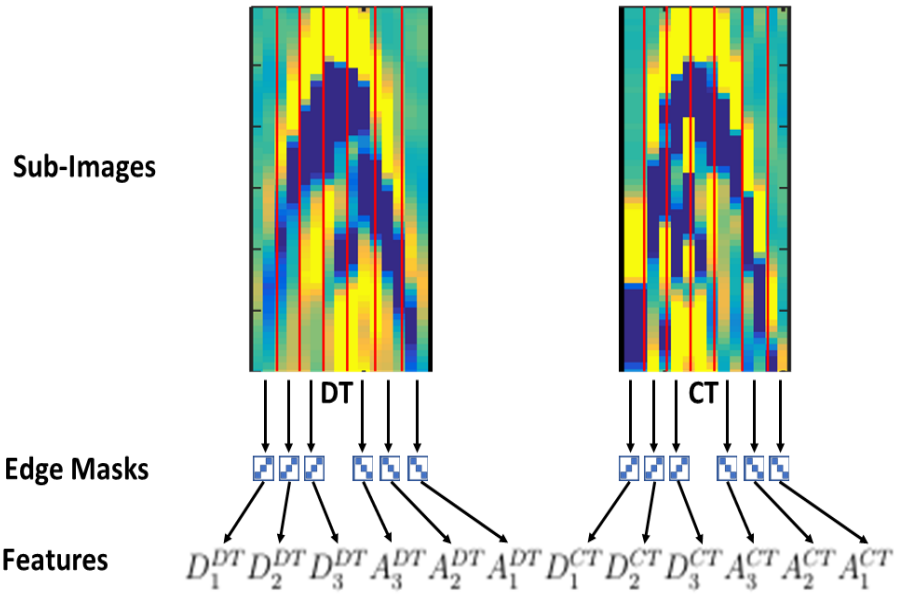


Figure 5.26: Extraction of the most discriminative EHD features.

5.3.4 BEO Clustering Setup

We applied our proposed FCMI algorithm to the BEO data using 10 target concepts. For the purposes of this clustering experiment, neither merging nor removal of weak target concepts was desired, so we did not utilize the PCMI. Unless otherwise noted, the FCMI was run with identical parameters to those used in the Sensitivity Analysis Experiment in Section 5.2.2.

5.3.5 BEO Clustering Results and Analysis

The results produced by running the FCMI were ten target concepts with substantial diversity in target depth, size, and shape. In figures 5.27-5.31 we display samples from five of the most distinctive BEO clusters, along with their respective EHD features. In figure 5.32 we display a single example from every cluster along with a plot of all respective features. Bags with high probability in TC 1 come predominantly from large, shallow targets, and have strong features across the board. By contrast bags with high probability in TC 2 come from relatively small and weak targets from variable depths, or as a response in deeper windows to strong, shallow targets. In all cases they have extremely weak features across the board. Bags with high probability in TC 3 have much stronger features along the inner diagonal and anti-diagonal features ($D_3^{DT}, A_3^{DT}, D_3^{CT}, A_3^{CT}$) than outer-diagonal and anti-diagonal features ($D_1^{DT}, A_1^{DT}, D_1^{CT}, A_1^{CT}$). Most of these come from small but dense targets at shallow depths. TC 4 is represented by bags with the opposite characteristics – the outer-diagonal and anti-diagonal

features $(D_1^{DT}, A_1^{DT}, D_1^{CT}, A_1^{CT})$ are much stronger than those features in middle $(D_3^{DT}, A_3^{DT}, D_3^{CT}, A_3^{CT})$, and most correspond to large and deep targets. Finally, bags with high probability in TC 5 have noticeably stronger CT features $(D_1^{CT}, D_2^{CT}, D_3^{CT}, A_3^{CT}, A_2^{CT}, A_1^{CT})$ than DT features $(D_1^{DT}, D_2^{DT}, D_3^{DT}, A_3^{DT}, A_2^{DT}, A_1^{DT})$. These tend to show up in deep windows for large targets from a variety of depths.

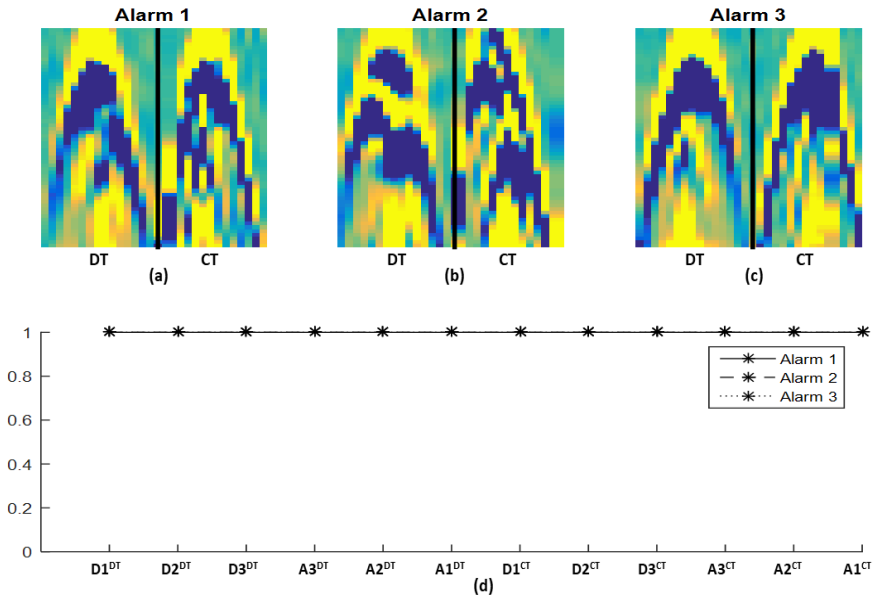


Figure 5.27: Target Concept 1 BEO samples. (a) (b) (c) Signatures taken from shallow, large targets. (d) Features are strong across the board.

One of the advantages to the FCMI is that the fuzzy memberships permit bags to share memberships across very distinct target concepts, and correspondingly high bag probabilities in the same. Of the 500 positive bags used in this experiment, over 100 had high (>0.9) bag probability in at least two target concepts. Figure 5.33 presents a few diverse examples of this behavior. In figure 5.33(a), a bag correpsonding to a large, shallow target has strong response to TC 1 in an early depth, and a strong response to TC 2 at a later depth, where the signature has substantially weakened. In figure 5.33(b), a bag correpsonding to a compact,

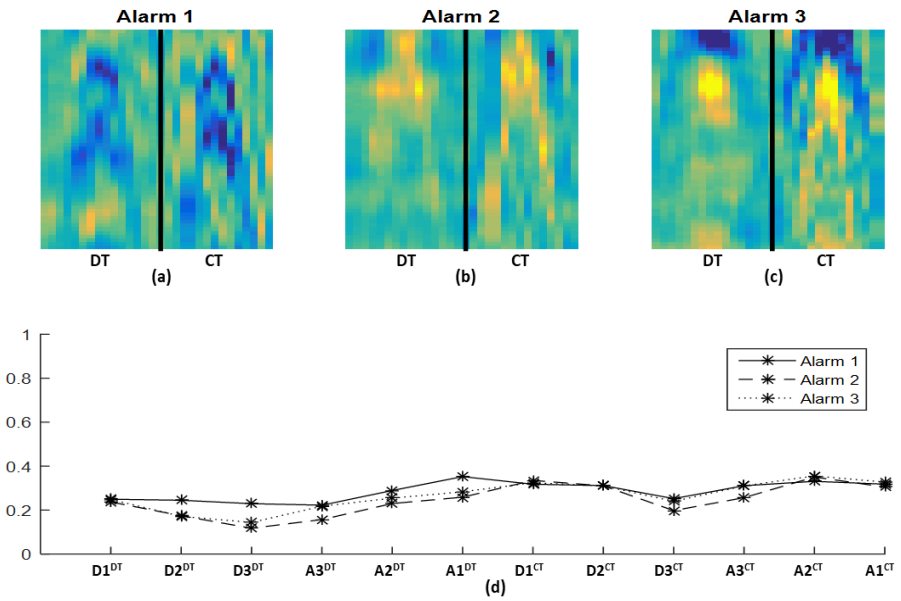


Figure 5.28: Target Concept 2 BEO samples. (a) (c) Signatures taken from shallow, small targets. (b) Signature taken from deep, small target. (d) Features are weak across the board.

shallow target has strong response to TC 3 in an early depth, and a strong response to TC 4 at a later depth, where the signature has expanded. Lastly, figure 5.33(c) depicts another shallow, compact target with high response to TC 3 at an early depth, but in this case a strong response at a later depth to TC 5, where only the CT feature still has a substantial profile.

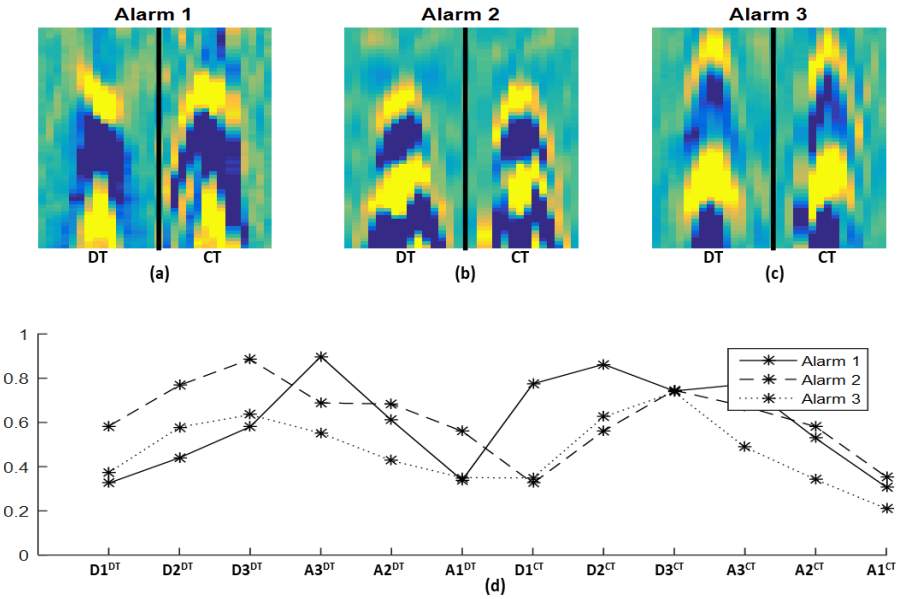


Figure 5.29: Target Concept 3 BEO samples. (a) (b) Signatures taken from shallow, compact targets. (c) Signature taken from deep, compact target. (d) Inner features $(D_3^{DT}, A_3^{DT}, D_3^{CT}, A_3^{CT})$ are stronger than outer features $(D_1^{DT}, A_1^{DT}, D_1^{CT}, A_1^{CT})$.

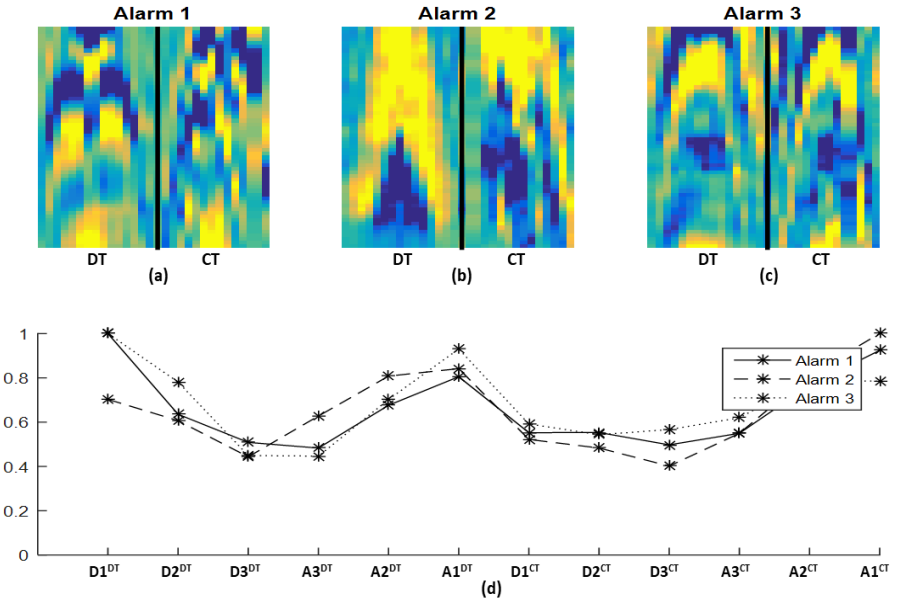


Figure 5.30: Target Concept 4 BEO samples. (a) (b) (c) Signatures taken from deep, large targets. (d) Outer features $(D_1^{DT}, A_1^{DT}, D_1^{CT}, A_1^{CT})$ are stronger than inner features $(D_3^{DT}, A_3^{DT}, D_3^{CT}, A_3^{CT})$

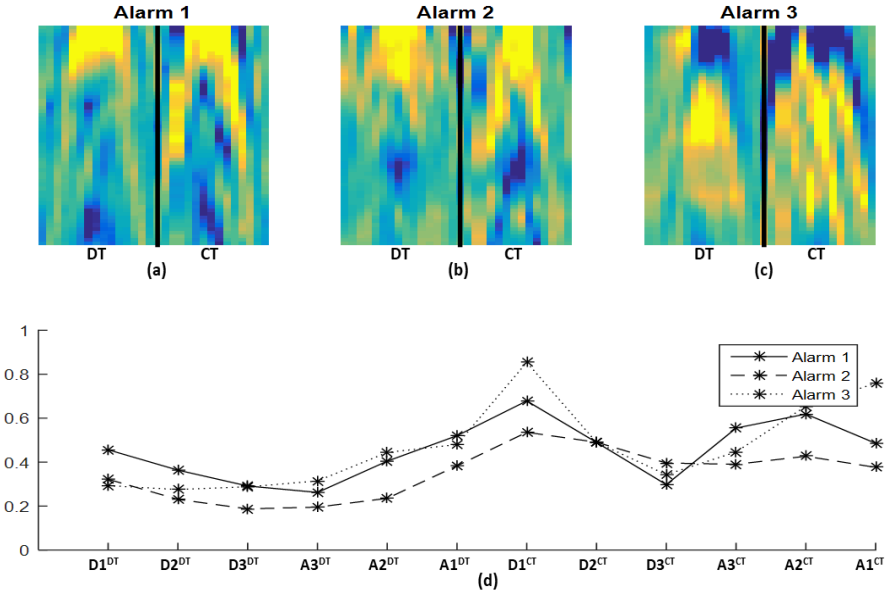


Figure 5.31: Target Concept 5 BEO samples. (a) (c) Signatures taken from shallow, compact targets. (b) Signatures taken from deep, compact target. (d) CT features ($D_1^{CT}, D_2^{CT}, D_3^{CT}, A_1^{CT}, A_2^{CT}, A_3^{CT}$) are stronger than DT features ($D_1^{DT}, D_2^{DT}, D_3^{DT}, A_1^{DT}, A_2^{DT}, A_3^{DT}$)

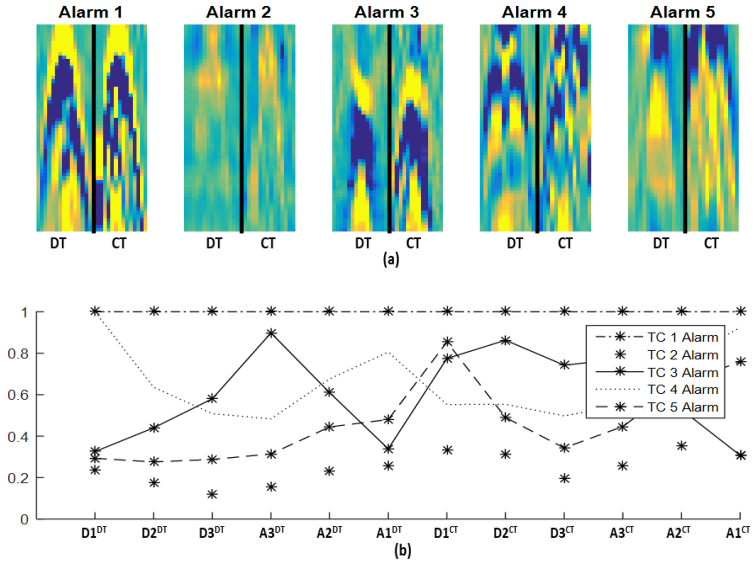


Figure 5.32: (a) One sample from every Target Concept and (b) their features.

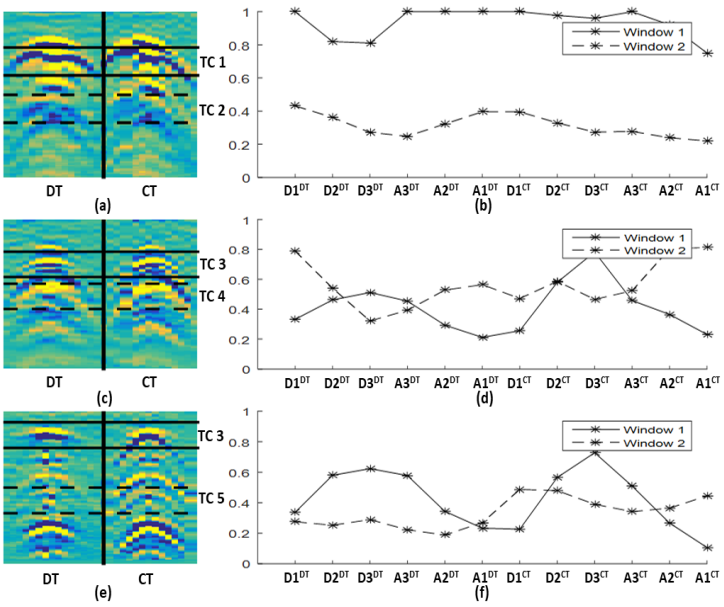


Figure 5.33: BEOs with multiple High Bag Probabilities. (a)(b) Large probability in TC 1 and TC 2 at distinct depths. (c)(d) High probability in TC 3 and TC 4 at different depths. (e)(f) High probability in TC 3 and TC 5 at different depths.

5.4 Application of FCMI to Benchmark Multiple Instance Data Classification

5.4.1 Benchmark Datasets

The first of two classification experiments we performed involves five datasets widely utilized when conducting MIL [2, 1, 48]. The first two datasets, MUSK1 and MUSK2 were adopted by the authors in [16] to closely mirror the drug-activity-prediction process. To be specific, each dataset consists of a set of distinct molecules (bags) each with a varying number of molecular conformations (instances). Each conformation is characterized by a set of surface measurement features. A portion of these molecules exhibit a musky smell (positive bags), while the remainder do not (negative bags).

The three remaining datasets, ELEPHANT, FOX, and TIGER, are COREL-based datasets adopted by the authors in [2] to evaluate an image annotation task. Each dataset consists of images (bags) broken into segments (instances), each of which of which is characterized by a combination of color, texture, and shape features. Positive bags were drawn directly from the COREL category in question, while negative bags were randomly generated from a set of images of other animals. Table 5.3 describes key characteristics for each dataset, while [16] and [2] can be consulted for a full description of the MUSK and COREL datasets respectively.

Table 5.3: Benchmark Dataset Summary Statistics

.;

	MUSK1	MUSK2	ELEPHANT	FOX	TIGER
# Pos. Bags	47	39	100	100	100
# Neg. Bags	45	63	100	100	100
Avg. # Inst Per Bag	5.17	64.69	6.60	6.96	6.10
# Features	166	166	230	230	230

5.4.2 Benchmark Classification Algorithm Setup

Classification was assessed for a total of six approaches. The first of these is the simple MDD-probability based approach described in Algorithm 9. Next, we assessed the utility of three embedded feature space appoaches using target concepts as outlined in 4.2 . Lastly, for the sake of comparison, we ran two implementations of classic embedded feature approaches in MIL approaches to gauge the efficacy of our performance.

All training and testing of classifiers was performed using a 10-fold cross-validation scheme. For FCMI-related algorithms, we applied the FCMI using 20 target concepts, a fuzzifier value of 1.5, 250 outer-loop fuzzy membership assignment steps, and 3 inner-loop target concept optimization steps. While merging of target concepts was not utilized [2], we did use the weak-TC removal scheme outlined in Section 3.3.5, using values of 0.025 and 2 for ϕ_{QL} and ϕ_{QL} respectively. All

[2]The use of the PCMI did not boost performance in our classification experiments.

other parameters were identical to those used with the Sensitivity Data Analysis Experiment as described in Section 5.2.2.

Embedding of features for the three relevant algorithms was performed with several combinations from those outlined in 4.2 during our research, but our analysis includes only three main embedding configurations. The first of these solely utilizes the positive TC MinDist as defined in 4.2.1, and includes all positive TCs (i.e. no weak TC selection was utilized). This is intended as a baseline result for our algorithms. The second is the same as the first, but only embeds features for target concepts that are not pruned by our selection approach. This is functionally similar to embedding performed by other MIL applications (e.g. the DD-SVM as described below). The third includes the Positive TC MinDist, the Positive TC MeanDist defined in (4.2.2), and the Negative TC MaxDist defined in (4.4.2). The rationale behind this setup was to include one set of features to potentially satisfy each of the Concept, Distribution, and Multi-concept dataset paradigms outlined in [11].

Below, we provide a full description of the six approaches we used, along with any relevant parameters.

1. FCMI-Prob: The simplest approach we utilized simply computes bag probability in each target concept for each testing bag, and then takes the overall maximum across target concepts for each bag as the net confidence. Training bags in this case are solely used for the purpose of removing weak target concepts as described above before these bag probabilities are computed. Negative target concepts play no role in this classifier.

2. EFCMI-CKNN: Our first embedded features approach is a modified approach based on an standard k-Nearest Neighbors [13] classification routine. For the standard KNN approach, each sample's embedded feature vector is compared to its closest k (in our case 20) neighbors in the training set using a simple Euclidean distance operation, and the proportion of these neighbors that are positive becomes our net confidence that the sample is positive. Our classifier includes a additional mechanic based on the Citation-kNN [55] algorithm that also includes "citers" in the confidence calculation. The "citer" operation performs a kNN test for all training samples against all training samples plus the prospective testing instance. The label of any training sample that includes the testing sample in its closest c (in our case 3) neighbors is added to the pool for confidence calculation.

3. EFCMI-SVM: Our second approach relies on construction of a support vector machine (SVM), trained on the embedded feature vectors. The implementation we use is packaged with the "Open Source Pattern Recognition Toolbox" [41]. All parameters were set to default, with the RBF kernel being used.

4. EFCMI-ONS: Our last approach relies on construction of a One-norm support vector machine (ONS), trained on the embedded feature vectors. The ONS is optimized by solving a linear programming problem as defined in [9]. We use values of 1.2 for λ and 2 for σ^2.

5. DD-SVM: The first embedded features algorithm to which we compare our

performance is the DD-SVM [10]. In short, the DD-SVM finds a set of local maximizers (centroids and scales) using some variation of the standard diverse density (DD) function using a number of starting points based on instances from positive bags. After merging similar maximizers, bags are then remapped using the remainder in a form virtually identical to the Positive TC Mindist (4.2.1)) and then a standard SVM is trained on the data. We use the EM-DD (as suggested by the authors) packaged with the MILL toolkit (toolkit ref) with 50 starting points per cross-validation (i.e. 50 potential points for feature embedding), as well as the same Pattern Recognition Toolbox SVM implementation defined above for the EFCMI-SVM algorithm. We note that the discrimination approach used for classification with the DD-SVM is functionally similar to that utilized by the EFCMI-SVM.

6. MILES: The second algorithm to which we compare our approaches is Multiple-Instance Learning via Embedded Instance Selection (MILES) [9]. In short, MILES pools every instance in the dataset (positive or negative) as a potential target concept, and after remapping the bags to these points using a scaled MLCE distance, constructs a One-Norm SVM to separate positive from negative data. A full description of MILES and its embedding approach can be found in 2.2.4. For the MUSK datasets, we used values of 2 for λ and 10^5 for σ^2. For the three COREL datasets, we used values of 1.2 for λ and 600 for σ^2. A link to a MILES implementation can be found in [9]. We note that the discrimination approach used for classification with the MILES algorithm is functionally similar to that utilized by the EFCMI-ONS.

5.4.3 Benchmark Classification Results and Analysis

Receiver operator characteristic curves (ROC) were generated for each of the above approaches, and the area under the curve (AUC) was computed in each case. Table 5.4 below depicts the complete results. As noted above, the entries for the EFCMI approaches include separate entries depending upon whether the weak TC were pruned and which embedded features were included.

Table 5.4: Benchmark Dataset Results (AUC)

.;

	MUSK1	MUSK2	ELEPHANT	FOX	TIGER
FCMI-Prob	.9031	.8083	.8426	.6403	.7871
EFCMI-CKNN (No Sel)*	.8844	.9078	.8758	.6819	.8403
EFCMI-CKNN (Simple Dist)**	.8903	.9023	.8777	.6822	.8332
EFCMI-CKNN	.9279	.9078	.9074	.6654	.8281
EFCMI-SVM (No Sel)*	.9314	.9373	.8869	.6902	.8332
EFCMI-SVM (Simple Dist)**	.9371	.9389	.8909	.6902	.8318
EFCMI-SVM	.9338	.9341	**.9203**	.6761	.8172
DD-SVM	**.9626**	**.9683**	.8877	.6564	.7635
EFCMI-ONS (No Sel)*	.9305	.9101	.8475	.6606	.8103
EFCMI-ONS (Simple Dist)**	.9182	.8962	.8475	.6612	.8067
EFCMI-ONS	.9480	.9263	.8808	**.6955**	.7962
MILES	.9428	.9560	.9018	**.6955**	**.9172**

*No Weak TC Removed, and only Pos TC Dist Embedding Used

**Only Pos TC Dist Embedding Used

Several observations can be made with respect to the AUC values in Table 5.4. As expected, the FCMI-Prob is the least effective in discriminating between data, and falls well short of other approaches. Secondly, we note that the two steps we took to improve FCMI classification had diverse effects depending upon the data

in question. In the case of weak TC removal, the net performance improvement is small across datasets and algorithms. On the other hand, the addition of the Positive TC MeanDist and Negative TC MaxDist embedded features has substantially more variation in its impact. For three of the datasets (MUSK1, MUSK2, and ELEPHANT) it confers a clear advantage overall for each of the algorithms. On the other hand, performance is decreased for the EFCMI-CKNN for both TIGER and FOX dataset, and decreased for the TIGER dataset for the EFCMI-ONS.

Next, we consider the two existing embedded classifiers we ran for the sake of comparison. For the COREL data, the EFCMI-SVM outperforms the DD-SVM to varying degrees, while the opposite holds true for the MUSK datasets. Coupled with the fact that the EM-DD steps used in computing target concepts for the DD-SVM are substantially more expensive computationally, the EFCMI-SVM holds several apparent advantages. The MILES algorithm is a different matter. In the case of the TIGER data, it clearly outperforms any of our approaches, as well as the DD-SVM, and in two other cases provides comparable performance to one of our algorithms (MUSK1 and FOX, for EFCMI-ONS). While the EFCMI-SVM performance on the ELEPHANT data was found to be the best overall, MILES performance on the MUSK2 data was also substantially better than that of any of our approaches. On the one hand, this may demonstrate the potential value in relying on a "brute force" method to select target concepts and embedded feature weights for classification. On the other hand, the MILES algorithm can become computationally expensive for larger datasets. As well, target concept centroids and scales obtained using the FCMI approach have a level of interpretability that

those obtained with MILES lack, since the latter rely on a uniform kernel for embedding and the generated weights can be difficult to analyze or justify.

5.5 Application of FCMI to BEO Classification

5.5.1 BEO Classification Dataset

Our second classification task revisits the BEO application for which clustering analysis was performed in Section 5.2. Instead of clustering the BEO data, our task in this case was to evaluate the efficacy of our algorithms in discriminating between BEOs and clutter object, and to compare their performance to existing algorithms. The dataset was drawn from the same two sites as outlined in Section 5.2, but in this case no restriction was made on the prescreening confidence for the false alarms, therefore allowing for a realistic dataset involving BEOs and potentially high-confidence clutter objects.

5.5.2 BEO Features and Classification Setup

We utilized the same FCMI-based embedded classifiers (EFCMI-CKNN, EFCMI-SVM, EFCMI-ONS) for this experiment as were described for the Benchmark datasets in Section 5.4.2, as well as the EHD features as described in Section 5.3.3. Several crucial differences exist for this task from those two, however. For one, we utilized all 35 down-track and all 35 cross-track features in this experiment, because other features (e.g. non-edge features, for example) play a vital role in discriminating BEOs from clutter. Secondly, because prior experiments have

demonstrated confidences from down-track and cross-track features to be best acquired though individual DT and CT classifier operation and then combined through a fusion mechanism, we ran the FCMI separately for DT features and CT features. Hence, we generated 20 DT target concepts, 20 CT target concepts, and generated a separate pair of confidences for each sample in testing. These confidences are then combined through a simple weighted arithmetic mean operation (i.e. $Conf_{NET} = (Conf_{DT} + 0.5(Conf_{CT}))$)

We compared the performance of these algorithms to both the LMS prescreener used to generate the alarms, as well as two existing BEO classifiers based on the EHD feature, one of which has been employed in the field for some time, and one of which has emerged as a prospect for effective BEO discrimination.

1. EHD-KNN: The Edge Histogram Descriptor k-nearest-neighbors approach (EHD-KNN) [20] is an algorithm that has been deployed in the field to combat the threat of BEOs for a substantial number of years. It relies on the construction of a set of prototypes using training EHD feature vectors and a self-organizing map (SOM) [31] approach. A possibilistic-KNN approach is used to assign confidence to prospective testing samples based on their proximity to a set of nearest neighbors. Separate confidences are generated for down-track and cross-track features, which are then combined using a simple geometric mean. The EHD-KNN algorithm has demonstrated superiority in its capability to discriminate between BEOs and clutter in many comparative studies.[28, 43, 23].

2. EHD-ONS: A recently introduced approach to BEO discrimination relies on

construction of a linear, one-norm SVM using EHD features. The EHD-ONS ultimately produces a simple set of classification weights and bias, allowing for an almost instantenous transformation from EHD feature to BEO confidence. This speed in testing is an important consideration in deployment.

Both EHD-KNN and EHD-ONS operate at the instance-level, producing an individual confidence per window for both DT and CT directions. DT and CT confidences for each window are then fused through a simple geometric mean, and the max window confidence overall becomes the net alarm confidence.

5.5.3 BEO Classification Results and Analysis

Figure 5.34 depicts the performance of our best-performing algorithm on the dataset (EFCMI-CKNN), alongside the LMS prescreener, EHD-KNN, and EHD-ONS algorithms, in the form of a Receiver Operating Curve (ROC). As can be seen, it is competitive or better than the two existing approaches for this small dataset. Further testing will need to be performed on larger BEO datasets to see if this trend holds. In addition, because the embedded features approach operates on a different mechanic for discrimination (at the bag level, instead of instance-level), and therefore encodes information differently, the EFCMI algorithms hold the potential to be effective in discriminator fusion going forward.

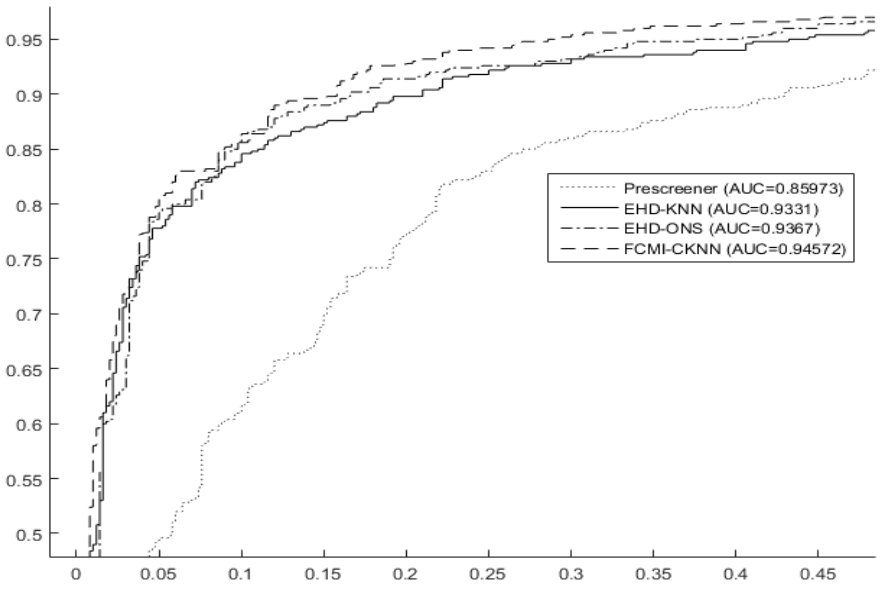


Figure 5.34: BEO Classification Results. ROC depicting the FCMI-CKNN algorithm to have comparable performance for the BEO dataset to two established algorithms (EHD-KNN and EHD-ONS).

CHAPTER 6

CONCLUSIONS

We proposed three measures that combine concepts from data clustering and multiple instance learning to analyze Multiple Instance data. These measures generalize the existing Diverse Density (DD) metric to include support for multiple target concepts. The first metric is the Crisp Multiple Concept Diverse Density (CMDD). Given a set of target concepts $TC_1, .., TC_k$, the CMDD assigns each bag to the closest target concept and computes the diverse density of each concept. The DD of the kth target concept, TC_k, is defined as the cumulative probability that all positive bags assigned to it are correlated with TC_k, and the cumulative probabilty that all negative bags assigned to it are *not* correlated with TC_k. We also developed an algorithm to maximize the CMDD criteria and learn the optimal target concepts. This algorithm, called the Crisp Clustering of Multiple Instance Data (CCMI), utilizes an alternating optimization technique that rotates between finding the best partition of the dataset into K groups, and using an iterative line search to optimize the location and scale of the target concept of each group.

Our second proposed metric is the Fuzzy Multiple-Concept Diverse Density

(FMDD). The FMDD criterion relaxes the constraint of the CMDD that each bag is assigned to one and only one target concept. Instead, the FMDD uses a fuzzy membership matrix that maps each bag to all target concepts, with the simple requirement that the membership values for each bag must sum to 1. After defining the FMDD metric, we proposed the Fuzzy Clustering of Multiple Instance Data (FCMI) algorithm to optimize the FMDD. Similar to the CCMI, the FCMI utilizes an alternating approach to optimization; in this case, the algorithm alternates between updating the fuzzy membership matrix and optimizing the locations and scales of the target concepts.

Our third proposed metric is the Possibilistic Multiple-Concept Diverse Density (PMDD). The PMDD removes the constraint of the FMDD that all memberships must sum to 1, permitting bags in equally close proximity to multiple target concepts to have a distincively higher set of memberships than with an extremely large, but equal, distance to the same target concepts. We then proposed the Possibilistic Clustering of Multiple Instance Data (PCMI) algorithm to optimize the PMDD, and then demonstrated how the PMDD memberships and bag probabilities could be used to merge similar target concepts and select weak ones for removal.

After defining our metrics, we provided two major approaches to classification that utilize a set of target concepts. The first of these relies on a simple bag probability calculation and the most-likely cause estimate (MLCE), while the second relies on the use of an embedded feature-space to map bags in the MIL dataset to a simple vector appropriate for any standard machine learning algorithm. We

further proposed three classifiers that rely on our FCMI optimization and these embedded features. EFCMI-CKNN combines our embedded features approach with a modified KNN classifier, EFCMI-SVM does the same but with the construction of a standard kernel-based support-vector machine, and the EFCMI-ONS does the same but utilizes a sparse, one-norm support-vector machine.

We applied our algorithms to synthetic and real world data. First we demonstrated that the CCMI and FCMI perform as well or better than the standard DD algorithm for a dataset with a single TC. Secondly, we demonstrated that, on a dataset with multiple TC, that the CCMI and FCMI were capable of learning the correct TC while the DD algorithm failed to do so. We also demonstrated that the PCMI could be used to merge similar targets with these synthetic datasets, resulting in better accuracy in learning the TC. We then demonstrated the the FCMI algorithm responds robustly to a number of parameter changes for the synthetic data, as well as provided an analysis of those changes for which it fails to respond well.

The FCMI was next validated using datasets taken from the real world application of buried explosive object (BEO) detection. We showed that the FCMI was capable of effectively partitioning the data into multiple, distinctive target concepts, each of which captures a distinct set of BEO types and properties. We further demonstrated through a simple classification task that our embedded approaches are competitive with both a deployed and an emergent BEO discrimination algorithms in discriminating between BEOs and clutter objects. In addition, we demonstrated that these same approaches are competitive with other mature

embedded feature space approaches in addressing five benchmark MIL datasets.

REFERENCES

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

[2] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann, *Support vector machines for multiple-instance learning*, Advances in neural information processing systems, 2002, pp. 561–568.

[3] James C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[4] Christopher M Bishop et al., *Pattern recognition and machine learning*, vol. 4, springer New York.

[5] Jeremy Bolton, Paul Gader, Hichem Frigui, and Pete Torrione, *Random set framework for multiple instance learning*, Information Sciences **181** (2011), no. 11, 2061–2070.

[6] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon, *Multiple instance learning: A survey of problem characteristics and applications*, Pattern Recognition **77** (2018), 329–353.

[7] E. Chang, Kingshy Goh, G. Sychay, and G. Wu, *Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines*, Circuits and Systems for Video Technology, IEEE Transactions on **13** (2003), no. 1, 26–38.

[8] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik, *Support vector machines for histogram-based image classification*, Neural Networks, IEEE Transactions on **10** (1999), no. 5, 1055–1064.

[9] Yixin Chen, Jinbo Bi, and James Z. Wang, *Miles: Multiple-instance learning via embedded instance selection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 12, 1931–1947.

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

[11] Veronika Cheplygina, David MJ Tax, and Marco Loog, *Multiple instance learning with bag dissimilarities*, Pattern Recognition **48** (2015), no. 1, 264–275.

[12] Vladimir Vapnik Corinna Cortes, *Support-vector networks*, Machine Learning **20** (1995), no. 3, 273.

[13] Thomas M. Cover and Peter E. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory **13** (1967), no. 1, 21–27.

[14] Gordon M Crippen and Timothy F Havel, *Distance geometry and molecular conformation*, vol. 74, Research Studies Press Taunton, England, 1988.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society, SeriesB **39** (1977), no. 1, 1–38.

[16] T.G. Dietterich, R.H. Lathrop, and Tomas Lozano-Perez, *Solving the multiple instance problem with axis-parallel rectangles*, Artificial Intelligence **89** (1997), 31–71.

[17] Harold Edson Driver and Alfred Louis Kroeber, *Quantitative expression of cultural relationships*, University of California Press, 1932.

[18] James R Foulds and Eibe Frank, *Speeding up and boosting diverse density learning*, Discovery Science, Springer, 2010, pp. 102–116.

[19] Yoav Freund, Robert E Schapire, et al., *Experiments with a new boosting algorithm*, Citeseer, 1996.

[20] Hichem Frigui and Paul, *Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors*, IEEE Transactions on Fu **17** (2009), no. 9, 185–199.

[21] Hichem Frigui, Lijun Zhang, Paul Gader, and Dominic Ho, *Context-dependent fusion for landmine detection with ground-penetrating radar*, Defense and Security Symposium, International Society for Optics and Photonics, 2007, pp. 655321–655321.

[22] K. J. Hintz, *Snr improvements in niitek ground penetrating radar*, in Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets IX, Orlando, FL, USA, April (2004).

[23] KC Ho, Paul D Gader, and Joseph N Wilson, *Improving landmine detection using frequency domain features from ground penetrating radar*, IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium, vol. 3, IEEE, 2004, pp. 1617–1620.

[24] Maximilian Ilse, Jakub M Tomczak, and Max Welling, *Attention-based deep multiple instance learning*, arXiv preprint arXiv:1802.04712 (2018).

[25] Nathan Ing, Jakub M Tomczak, Eric Miller, Isla P Garraway, Max Welling, Beatrice S Knudsen, and Arkadiusz Gertych, *A deep multiple instance model to predict prostate cancer metastasis from nuclear morphology*, (2018).

[26] M Maher Ben Ismail and Hichem Frigui, *Possibilistic clustering based on robust modeling of finite generalized dirichlet mixture*, 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 573–576.

[27] Melih Kandemir and Fred A Hamprecht, *Instance label prediction by dirichlet process multiple instance learning.*

[28] Andrew Karem, Aleksey Fadeev, Hichem Frigui, and Paul Gader, *Comparison of different classification algorithms for landmine detection using gpr*, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XV, vol. 7664, International Society for Optics and Photonics, 2010, p. 76642K.

[29] Leonard Kaufman and Peter J Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, vol. 344, John Wiley & Sons, 2009.

[30] James D. Keeler, David E. Rumelhart, and Wee-Kheng Leow, *Integrated segmentation and recognition of hand-printed numerals*, Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3 (San Francisco, CA, USA), NIPS-3, Morgan Kaufmann Publishers Inc., 1990, pp. 557–563.

[31] Teuvo Kohonen, *Exploration of very large databases by self-organizing maps*, Proceedings of International Conference on Neural Networks (ICNN'97), vol. 1, IEEE, 1997, pp. PL1–PL6.

[32] Tomáš Komárek and Petr Somol, *Multiple instance learning with bag-level randomized trees*, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 259–272.

[33] R. Krishnapuram, H. Frigui, and O. Nasraoui, *Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. i*, Fuzzy Systems, IEEE Transactions on **3** (1995), no. 1, 29–43.

[34] R. Krishnapuram and J.M. Keller, *A possibilistic approach to clustering*, Fuzzy Systems, IEEE Transactions on **1** (1993), no. 2, 98–110.

[35] Xu Liu, Licheng Jiao, Jiaqi Zhao, Jin Zhao, Dan Zhang, Fang Liu, Shuyuan Yang, and Xu Tang, *Deep multiple instance learning-based spatial–spectral*

classification for pan and ms imagery, IEEE Transactions on Geoscience and Remote Sensing **56** (2018), no. 1, 461–473.

[36] Stuart Lloyd, *Least squares quantization in pcm*, Information Theory, IEEE Transactions on **28** (1982), no. 2, 129–137.

[37] Jun Lou, Tian Jin, Fulai Liang, and Zhimin Zhou, *A novel prescreening method for land-mine detection in uwb sar based on feature point matching*, Geoscience and Remote Sensing, IEEE Transactions on **51** (2013), no. 6, 3706–3714.

[38] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics (Berkeley, Calif.), University of California Press, 1967, pp. 281–297.

[39] Oded Maron, *Learning from ambiguity*, Ph.D. thesis, Massassachusetts Institute of Technology, MIT, 1998.

[40] Oded Maron and Tomás Lozano-Pérez, *A framework for multiple-instance learning*, Advances in Neural Information Processing Systems **10** (1998), no. 1, 570–576.

[41] Kenneth D Morton Jr, Peter Torrione, Leslie Collins, and Sam Keene, *An open source pattern recognition toolbox for matlab*, arXiv preprint arXiv:1406.5565 (2014).

[42] William M Rand, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical association **66** (1971), no. 336, 846–850.

[43] Christopher Ratto, Peter Torrione, Kenneth Morton, and Leslie Collins, *Context-dependent landmine detection with ground-penetrating radar using a hidden markov context model*, 2010 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2010, pp. 4192–4195.

[44] Soumya Ray and Mark Craven, *Supervised versus multiple instance learning: An empirical comparison*, Proceedings of the 22nd international conference on Machine learning, ACM, 2005, pp. 697–704.

[45] David E Rumelhart and David Zipser, *Feature discovery by competitive learning**, Cognitive science **9** (1985), no. 1, 75–112.

[46] Eli Saber and A Murat Tekalp, *Region-based shape matching for automatic image annotation and query-by-example*, Journal of Visual Communication and Image Representation **8** (1997), no. 1, 3–20.

[47] Bernard W Silverman, *Density estimation for statistics and data analysis*, Routledge, 2018.

[48] Hanqiang Song, Zhuotun Zhu, and Xinggang Wang, *Bag reference vector for multi-instance learning*, arXiv preprint arXiv:1512.00994 (2015).

[49] S. Srinivas, *A generalization of the noisy-or model*, Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (1993), 208–218.

[50] Yijun Sun and Jian Li, *Adaptive learning approach to landmine detection*, Aerospace and Electronic Systems, IEEE Transactions on **41** (2005), no. 3, 973–985.

[51] P. A. Torrione, C. S. Throckmorton, L. M. Collins, F. Clodfelter, S. Frasier, and I Starnes, *Application of the lms algorithm to anomaly detection using the wichmann/niitek ground-penetrating radar*, Proceedings of the SPIE: Detection and Remediation Technologies for Mines and Minelike Targets VIII, vol. 5089, 2003, pp. 1127–1136.

[52] Peter A Torrione, Chandra S Throckmorton, and Leslie M Collins, *Performance of an adaptive feature-based processor for a wideband ground penetrating radar system*, Aerospace and Electronic Systems, IEEE Transactions on **42** (2006), no. 2, 644–658.

[53] Mohamed Trabelsi and Hichem Frigui, *Robust fuzzy clustering for multiple instance regression*, Pattern Recognition (2019).

[54] Robert Choate Tryon, *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*, Edwards brother, Incorporated, lithoprinters and publishers, 1939.

[55] Jun Wang, *Solving the multiple-instance problem: A lazy learning approach*, In Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, 2000, pp. 1119–1125.

[56] J.N. Wilson, P. Gader, Wen-Hsiung Lee, H. Frigui, and K.C. Ho, *A large-scale systematic evaluation of algorithms using ground-penetrating radar for landmine detection and discrimination*, Geoscience and Remote Sensing, IEEE Transactions on **45** (2007), no. 8, 2560–2572.

[57] Chee Sun Won, Dong Kwon Park, and Soo-Jun Park, *Efficient use of mpeg-7 edge histogram descriptor*, ETRI Journal **24** (2002), no. 1, 23–30.

[58] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu, *Deep multiple instance learning for image classification and auto-annotation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3460–3469.

[59] Yan Xu, Jun-Yan Zhu, I Eric, Chao Chang, Maode Lai, and Zhuowen Tu, *Weakly supervised histopathology cancer image segmentation and classification*, Medical image analysis **18** (2014), no. 3, 591–604.

[60] Changbo Yang, Ming Dong, and Jing Hua, *Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning*, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, 2006, pp. 2057–2063.

[61] Mina Yousefi, Adam Krzyżak, and Ching Y Suen, *Mass detection in digital breast tomosynthesis data using convolutional neural networks and multiple instance learning*, Computers in biology and medicine **96** (2018), 283–293.

[62] Lotfi A Zadeh, *Fuzzy sets*, Information and control **8** (1965), no. 3, 338–353.

[63] Cha Zhang, John C Platt, and Paul A Viola, *Multiple instance boosting for object detection*, Advances in neural information processing systems, 2006, pp. 1417–1424.

[64] Dan Zhang, Fei Wang, Luo Si, and Tao Li, *M 3 ic: maximum margin multiple instance clustering*, Twenty-First International Joint Conference on Artificial Intelligence, 2009.

[65] Deyuan Zhang, Bingquan Liu, Chengjie Sun, and Xiaolong Wang, *Random sampling image to class distance for photo annotation.*

[66] Qi Zhang and Sally A. Goldman, *Em-dd: An improved multiple-instance learning technique*, In Advances in Neural Information Processing Systems, MIT Press, 2001, pp. 1073–1080.

[67] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li, *Multi-instance learning by treating instances as non-i.i.d. samples*, Proceedings of the 26th Annual International Conference on Machine Learning (New York, NY, USA), ICML '09, ACM, 2009, pp. 1249–1256.

CURRICULUM VITAE

NAME: Andrew D. Karem

ADDRESS: Department of CECS
JB Speed School of Engineering
University of Louisville
132 Eastern Parkway
Louisville, KY 40292

DOB: Louisville, KY - October 28, 1979

EDUCATION
& TRAINING: B.S., Computer Science
Rice University
Houston, Texas
1998 - 2002

M.S., CECS
University of Louisville
Louisville, KY
2003 - 2007

Ph.D., CSE
University of Louisville
Louisville, KY
2013 - 2019