View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Louisville

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2018

Robust fuzzy clustering for multiple instance regression.

Mohamed Trabelsi

University of Louisville

Follow this and additional works at: https://ir.library.louisville.edu/etd

Part of the Robotics Commons

Recommended Citation

Trabelsi, Mohamed, "Robust fuzzy clustering for multiple instance regression." (2018). *Electronic Theses and Dissertations.* Paper 2975.
https://doi.org/10.18297/etd/2975

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

ROBUST FUZZY CLUSTERING FOR MULTIPLE INSTANCE REGRESSION

By

Mohamed Trabelsi
B.E.,Tunisia Polytechnic School, 2016

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

Master of Science in Computer Science

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2018

Copyright 2018 by Mohamed Trabelsi

All rights reserved

ROBUST FUZZY CLUSTERING FOR MULTIPLE INSTANCE REGRESSION

By

Mohamed Trabelsi
B.E.,Tunisia Polytechnic School, 2016

A Thesis Approved On

April 19th, 2018

Date

By the following Thesis Committee:

Hichem Frigui, Ph.D., Thesis Director

Zhu Xuwen, Ph.D.

Amir A. Amini, Ph.D.

Olfa Nasraoui, Ph.D.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Hichem Frigui for his guidance, assistance and encouragements during both planning and fulfillment of this work.

I thank Dr. Olfa Nasraoui, Dr. Zhu Xuwen and Dr. Amir A. Amini for accepting to serve on my thesis committee and being a part of this special milestone.

I would like to thank my colleagues in the Multimedia Research Laboratory, and the Computer Engineering and Computer Science Department for their friendship and support.

Finally, I want to thank my family for their continuous support and unconditional love.

ABSTRACT

ROBUST FUZZY CLUSTERING FOR MULTIPLE INSTANCE REGRESSION

Mohamed Trabelsi

April 19th, 2018

Multiple instance regression (MIR) operates on a collection of bags, where each bag contains multiple instances sharing an identical real-valued label. Only few instances, called primary instances, contribute to the bag labels. The remaining instances are noise and outliers observations. The goal in MIR is to identify the primary instances within each bag and learn a regression model that can predict the label of a previously unseen bag. In this thesis, we introduce an algorithm that uses robust fuzzy clustering with an appropriate distance to learn multiple linear models from a noisy feature space simultaneously. We show that fuzzy memberships are useful in allowing instances to belong to multiple models, while possibilistic memberships allow identification of the primary instances of each bag with respect to each model. We also use possibilistic memberships to identify and ignore noisy instances and determine the optimal number of regression models. We evaluate our approach on a series of synthetic data sets, remote sensing data to predict the yearly average yield of a crop and application to drug activity prediction. We show that our approach achieves higher accuracy than existing methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

LIST OF ALGORITHMS

ALGORITHM Page

CHAPTER 1

INTRODUCTION

Multiple Instance Learning (MIL) [1, 2] was proposed as an alternative to traditional single instance supervised learning for problems with incomplete knowledge about labels of training examples. In single instance (SI) learning, an object is represented by one instance and every training instance is assigned a discrete or real-valued label. In comparison, in MIL, an object is represented by a collection of feature vectors, or instances, called a bag. Each bag can contain a different number of instances. Labels are available at the bag level, however, labels of individual instances within a bag are unknown. This many-to-one relationship between object's instances and object's label produces an inherent ambiguity in determining which instances in a given bag are responsible for its associated label.

MIL was formalized in 1997 by Dietterich et al. providing a solution to drug activity prediction [1]. In this application, a molecule is represented by a bag that contains all possible conformations of this molecule, and every molecule's conformation is represented as an unlabeled instance. Thus, a bag, which is either positive or negative, is represented as a collection of unlabeled instances. Each instance of the bag is a fixed-length vector of numeric or nominal attribute values, like in the standard SI learning. Ever since, MIL has increasingly been applied to a wide variety of tasks including drug discovery [3], image analysis [4–6], content-based information retrieval [7], time series prediction [2], landmine detection [8], information fusion [9, 10], and remote sensing [11].

Most of the existing work in MIL has focused on multiple instance classification (MIC). Given a training set of labeled bags, the goal of MIC is to learn a concept that predicts the labels of training data at the instance level and generalizes to predict the labels of testing bags and their instances [1]. Many algorithms have been proposed to solve MIC problem in the past two decades. Examples include APR [1], MILES [5], MILIS [12], SVM [13], Diverse Density [3], MI kernels [14], EM-DD [15], Citation-kNN [16] and BP-MIP [17], [18].

In the traditional multiple instance formulation, a bag is labeled negative if all of its instances are negative, and positive if at least one of its instances is positive. As shown in figure 1.1, if one of the

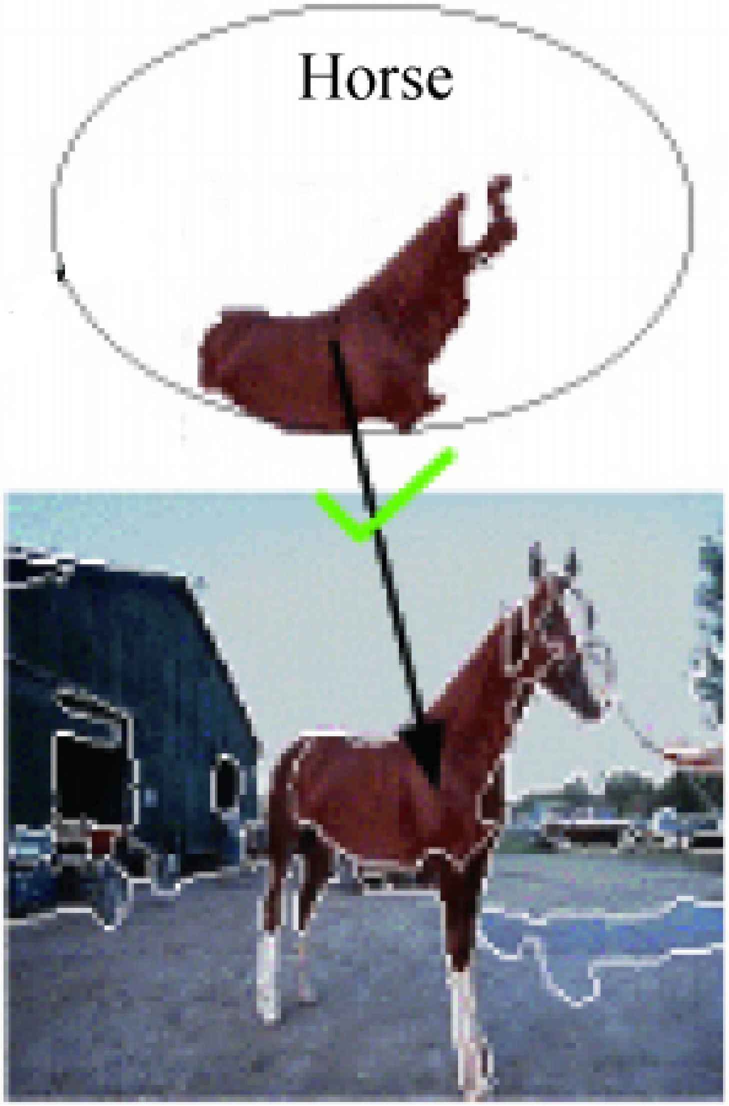
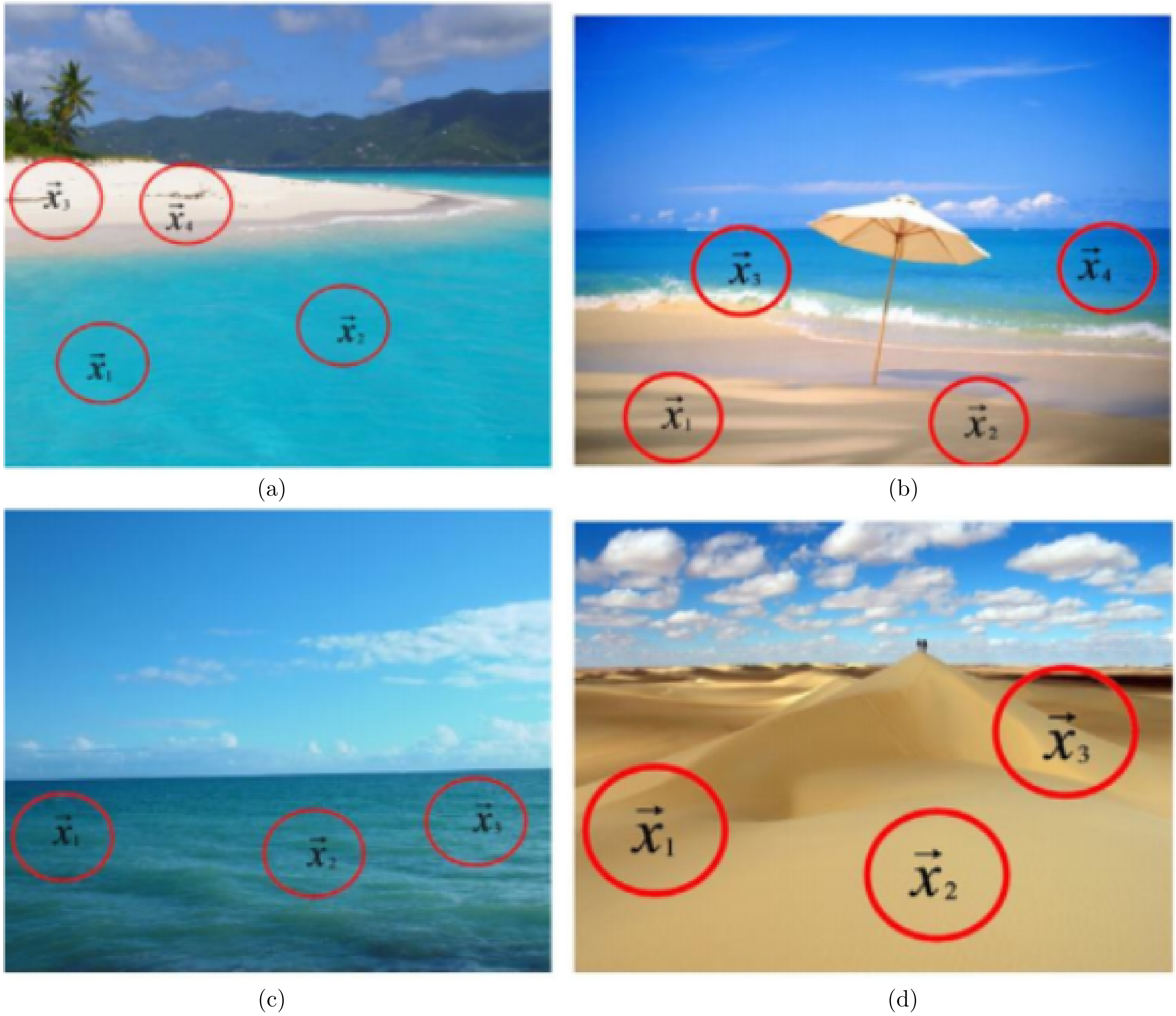


Figure 1.1: Illustration of Horse concept for MIL

regions in the image is a horse region, the image is positive for the concept horse. Recently, studies have shown that the traditional MIL formulation is not able to solve some complex problems [19]. Therefore, several other MIL formulations are proposed. For example, some paradigms try to better describe the content of an image by introducing concurrency of some concepts [19,20]. An intuitive example is shown in figure 1.2, where an image that contains regions of sea and sand is most likely to be positive for concept beach. That is, both water and sand segments are positive instances for beach images. If only one of these concepts is present, then the picture could belong to the desert or sea class as illustrated in figure 1.2.

Some MIL formulations explore the positive and negative concurrency so that a bag is classified



(a) (b)

(c) (d)

Figure 1.2: Illustration of Beach example for MIL. Both (a) and (b) illustrate positive bags for class beach where both sand and sea concepts are present. (c) and (d) illustrate negative bags for the Beach class where either only the Sea or Desert concepts are present.

positive when instances associated with positive concepts are present and instances associated with negative concepts are not [21]. In addition to the above approaches, multiple MIL paradigms have been proposed [22].

Compared to MIC, multiple instance regression (MIR) has received much less attention. In MIR, bags have real-valued labels and the goal is to learn a regression model that can predict the label of a new bag from the features of its instances. MIR is a challenging learning task since we have no prior knowledge of the primary instances, i.e., instances within each bag that are relevant to its label. In fact, for the general MIR setting, the unknown number of relevant instances can vary from one bag to another.

In this work, we propose a novel MIR framework, called Robust Fuzzy Clustering for MIR (RFC-MIR). In RFC-MIR, we show that regression models can be identified as clusters when appropriate features and distances are used. Our approach uses two types of memberships. The first one is a constrained fuzzy membership while the second one is unconstrained and based on possibility theory. We show that fuzzy memberships are useful in allowing all instances within each bag to contribute to all potential models. We also show that possibilistic memberships can be used to identify non-primary instances as noise and outliers and reduce their influence on the learned regression parameters.

The remainder of this thesis is organized as follows. Chapter 2 provides a review of existing methods used for Multiple Instance Classification and Multiple Instance Regression. Chapter 3 introduces our proposed learning approach for Multiple Instance Regression. Chapter 4 provides experimental results and analysis of the proposed RFC-MIR. Finally, chapter 5 provides conclusions and potential future work.

CHAPTER 2

LITERATURE REVIEW

Most existing work in MIL has focused on multiple instance classification (MIC). MIC algorithms can be categorized into three main paradigms: instance space, bag space, and embedded instance space.

2.1 MIC Algorithms based on instance space paradigm

Instance space-based algorithms rely on the standard multiple instance assumption, which states that a positive bag must contain at least one positive instance [1]; the labels of remaining instances are irrelevant. These algorithms seek points in the instance feature space with strong correlation to instances from positive bags and no or low correlation to instances from negative bags. These points, called target concepts (TC), serve as loci for instance-level class labeling. Examples of instance space-based algorithms include the Axis-Parallel Rectangles (APR) [1], the Diverse Density (DD) [3], EM-DD [23] and MI-SVM [13]. These algorithms are outlined in the following subsections.

2.1.1 The Axis-Parallel Rectangles (APR) algorithm

APR was introduced to solve the drug activity prediction problem [1]. APR constructs a set of boundaries in the problem feature space to capture the TC. There are three variations of the APR algorithm: Standard APR; Outside-in multiple instance APR and Inside-out multiple instance APR.

In standard APR, the first step is to cover all instances from positive bags by finding the smallest hyper-rectangle that encloses these instances. The next step is to remove negative instances by shrinking the bounds. APR suggests a greedy feature selection in order to remove the most negative instances. In Outside-in APR algorithm, the feature selection is modified in order to keep at least one instance per positive bag. The elimination of negative instances is based on a kernel density estimate (KDE) [24] of the positive instances. This method is computationally more expensive than the first one because it requires an estimation of KDE. The Inside-out algorithm is an alternative to the Outside-in approach. The APR starts with a single positive instance and grows to include more positive instances. APR introduces the Iterated Discrimination algorithm [1] that is based on three procedures: Grow, Discriminate and Expand. The APR grows with tight bounds for a set of features. Then it chooses a set of distinguishing features and expands the APR's bounds for generalization.

2.1.2 Diverse Density (DD) and EM-DD

Diverse Density (DD) [3] and EM-DD [23] algorithms use optimization techniques to learn the TC.

Let $D = \{B_j, j = 1 \ldots N_B\}$ be a collection of N_B bags and labels $L = \{y_1, \ldots, y_{N_B}\}$, where $B_j = \{(b_{ij}, y_j), i = 1 \ldots n_j\}$, $b_{ij} \in \mathrm{I\!R}^d$ is the attribute vector representing the ith instance from the jth bag, y_j is the categorical target value of the jth bag and n_j is the number of instances in the j^{th} bag. The Diverse Density of a target h is defined as

$$DD(h) = Pr(h \mid D) = \frac{Pr(D \mid h)Pr(h)}{Pr(D)}, \tag{2.1}$$

With a uniform prior on the hypothesis space and independence of D given h , the maximum likelihood hypothesis, h_{DD} , is defined as

$$h_{DD} = \arg\max_{h \in H} Pr(D \mid h) = \arg\min_{h \in H} \sum_{i=1}^{N_B} -(log(Pr(y_i \mid h, B_i))) \tag{2.2}$$

$Pr(y_i \mid h, B_i)$ is estimated as

$$Pr(y_i \mid h, B_i) = 1 - |y_i - \mathit{Label}(B_i \mid h)| \tag{2.3}$$

where $\mathit{Label}(B_i \mid h)$ is the label that would be given to B_i if h is the correct hypothesis.

The influence of every feature to the bag's label is unequal for most applications. This characteristic is modeled in the DD algorithm by introducing scale factors. Thus, the target concept consists of feature and scale values. To estimate $\mathit{Label}(B_i \mid h)$, the DD algorithm used a model introduced by Maron and Lozano-Perez in [3]

$$\mathit{Label}(B_i \mid h) = \max_j \left\{ exp[-\sum_{k=1}^{d}(s_k(b_{jik} - h_k))^2] \right\} \tag{2.4}$$

where s_k is a scale factor indicating the importance of feature k, h_k is the feature value for dimension k, and b_{jik} is the feature value of instance bji on dimension k.

The negative logarithm of DD is defined as

$$NLDD(h,D) = \sum_{i=1}^{N_B} -(log(Pr(y_i \mid h, B_i))) \tag{2.5}$$

or

$$NLDD(h,D) = \sum_{i=1}^{N_B} -(log(1 - |y_i - \max_j \Big\{exp[-\sum_{k=1}^{d}(s_k(b_{jik} - h_k))^2]\Big\}|)) \tag{2.6}$$

The DD algorithm uses a two-step gradient ascent search [25] in order to find h (feature value and scale) that minimizes $NLDD$.

The EM-DD algorithm combines the Expectation Maximization (EM) method [26] with the DD algorithm. The EM-DD starts with initial guess of the hypothesis using instances from positive bags. In the E-step, the target is used to select the most positive instance from every positive bag. The expectation step reduces equation (2.6) by eliminating the maximum that occurs in its equation. Then in the M-step, the EM-DD algorithm uses a two-step gradient ascent to determine the new target and scales. The EM-DD algorithm keeps iterating between these two steps until a convergence criteria is reached.

In [27], clustering methods were used to generalize the DD algorithm to learn multiple target concepts simultaneously.

2.1.3 MI-SVM

MI-SVM [13] expands Support Vector Machines (SVMs) to deal with MIL problems. It generalizes the margin's notion to bags. MI-SVM intends to maximizing the bag margin directly. The prediction of the label of bag B_i in the training phase, using a generalization of SVM, is given by

$$\hat{y_i} = y_i \max_{j \in [1 \ldots n_i]} (< w, b_{ji} > +a) \tag{2.7}$$

with w and a are the parameters of the SVM model.

In (2.7), only one instance per positive bag is relevant to the bag's label. Thus, the MIL version of soft-margin classifier is given by

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{k=1}^{N_B}\xi_k \\ &\text{subject to} \qquad y_k \max_{j \in [1 \ldots n_k]} (< w, b_{jk} > +a) \geq 1 - \xi_k \ , \xi_k \geq 0 \ , \forall k. \end{aligned} \tag{2.8}$$

All instances of negative bags are negative and the label of negative bags is -1. Thus, the constraint in (2.8), on a negative bag B_i, can be rewritten as

$$- < w, b_{ji} > +a \geq 1 - \xi_i \ , \forall j \in [1, \ldots n_i] \tag{2.9}$$

For positive bags, MI-SVM algorithm introduces a selector variable $s(\{j = 1 \ldots n_i\})$, for each bag B_i , that represents its selected instance. This instance is denoted as the most positive instance of a positive bag. By introducing the selector variable, the constraint in (2.8), on a positive bag B_i, can be rewritten as

$$< w, b_{s(\{j=1 \ldots n_i\})} > +a \geq 1 - \xi_i \ , \forall j \in [1, \ldots n_i] \tag{2.10}$$

The final formulation of MI-SVM optimization problem is then defined as:

$$\begin{aligned} & \min_{s} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{k=1}^{N_B} \xi_k \\ & \text{subject to} \qquad \forall k \ , y_k = -1 \ and - < w, b_{jk} > +a \geq 1 - \xi_k \ , \xi_k \geq 0 \ . \\ & \qquad \forall k \ , y_k = 1 \ and \ < w, b_{s(\{j=1 \ldots n_k\})} > +a \geq 1 - \xi_k \ , \xi_k \geq 0 \ . \end{aligned} \tag{2.11}$$

MI-SVM algorithm introduces an optimization heuristic to solve (2.11) . For a given selector variable, MI-SVM update the SVM parameters and vice versa. For initialization, the mean of instances of every positive bag can be used. Then a standard SVM model is trained using the selected instances from positive bags and all instances from negative bags. The next step is to use the trained model to select the most positive instances from positive bags. The algorithm converges in the case of consistent selection in two consecutive iterations.

2.2 MIC Algorithms based on bag space paradigm

In the bag space MIC, each bag is mapped to an N_B-dimensional feature vector based on a bag-to-bag comparator metric with respect to all N_B bags within the training data. A key advantage of the bag space paradigm is that the mapped bag representation removes the instance-level ambiguity from the problem. Examples of such methods include the Citation k-NN classifier [28] and Multiple Instance Dissimilarity (MInD) [29].

2.2.1 The Citation k-NN algorithm

The k Nearest Neighbor (k-NN) can be adapted to MIL data with an appropriate function that extends the distance between bags. One such distance is the minimum Hausdorff which can be adapted to bags using

$$dist(B_i, B_j) = \min_k \min_l \|b_{ki} - b_{lj}\| \tag{2.12}$$

The label of an unlabeled bag can be predicted based on the label of its k-NN. The Citation k-NN [28] is a variation of the k-NN that is used to overcome the problem of the confused prediction by the false positive instances in positive bags. In addition to the nearest neighbors, known as references, Citation k-NN introduces the citers. For a given bag B_i, a citer is a bag that counts B_i as one of its nearest neighbors based on the minimum Hausdorff distance. So, Citation k-NN predicts the bag's label using the labels of both the citers and references of the bag.

2.2.2 Dissimilarity measures for Multiple Instance Data

Multiple Instance Dissimilarity [29] introduces an alternative to the standard methods that are based on the use of dissimilarities between bags and prototype instances, or between bags and prototype bags.

The distance between a given bag B_i and a prototype bag B_j can be defined as:

$$d^{bag}(B_i, B_j) = \frac{1}{n_i} \sum_{k=1}^{n_i} \min_l d(b_{ki}, b_{lj}) \tag{2.13}$$

with $d(b_{ki}, b_{lj})$ is the distance between two feature vectors, i.e, instances, b_{ki} and b_{lj}.
This representation results in a relatively low-dimensional feature space that depends on the number of bags in the training data.

Using partial distances at the instance level, the distance between a given bag B_i with n_i instances and a prototype bag B_j can be represented by an n_i-dimensional feature vector using:

$$d^{inst}(B_i, B_j) = [\min_l d(b_{1i}, b_{lj}), \min_l d(b_{1i}, b_{lj}), \ldots, \min_l d(b_{n_i i}, b_{lj})] \tag{2.14}$$

The instance representation results in a high-dimensional feature space, that depends on the total number of instances in the training data.

The bag representation approach can suffer from averaging dissimilarities when only few instances per bag are relevant. On the other hand, an instance representation of a bag can lead to a very high dimensional feature space. To address this potential drawback, in [30], the authors proposed a random subspace method (RSM) to train and combine multiple classifiers on lower dimensional subspaces.

In general, Multiple Instance Dissimilarity based algorithms fall into two categories. The first one chooses each bag prototype as a subspace while the second approach chooses each subspace randomly. Next, for each subspace, 1-norm SVM [31], that simultaneously performs feature selection and classification, is used for classification. Finally, various simple fusion methods can be used to fuse the multiple classifiers. Examples include majority voting and the product, maximum, or average of the posterior probabilities.

2.3 MIC Algorithms based on embedded instance space

Embedded instance space methods also map each bag to a single feature vector. Moreover, unlike the dissimilarity based approach, in the instance space, target concepts are learned and used for this mapping. Examples of algorithms that fall into this category include the DD-SVM [32], MILES [5] and MI-AdaBoost [21].

2.3.1 DD-SVM

The DD [3] algorithm uses optimization techniques to learn a single positive target concept (TC). In most applications, a single positive TC is not enough to describe the training data. Thus, the EM-DD [23] algorithms learns multiple positive TC's by combining EM technique and DD algorithm. In many applications, negative TC's can be relevant feature for the classification. Thus, DD-SVM [32] locates candidate for positive and negative target concepts (TC's) across multiple runs of the DD algorithm with distinct starting points. Then, a nonlinear mapping is defined using the selected prototypes and maps every bag to the bag feature space. Finally, a standard SVM classifier is trained in the bag feature space.

The first step is to select positive and negative prototypes from the training data $D = \{B_j, j = 1 \ldots N_B\}$. This step is based on the Diverse Density of a hypothesis h, defined as

$$DD(h, D) = \prod_{i=1}^{N_B} \Big[\frac{y_i + 1}{2} - y_i \prod_{j=1}^{n_i} (1 - \exp^{-\|b_{ji} - h\|_w}) \Big] \tag{2.15}$$

with

$$\|h\|_w = (h^T Diag(w)^2 h)^{\frac{1}{2}} \tag{2.16}$$

$DD(h, D)$ is close to 1 for hypothesis h that is far away from instances in all negative bags and close to instances from different positive bags. Learning positive prototypes is an optimization problem. The objective is to find different local maximizers of $DD(h, D)$ using gradient based methods. DD-SVM proposes a multiple runs of searching the local maximizers to solve the problem of unknown number of the local maximizers of $DD(h, D)$. The difference between these runs is the starting point. DD-SVM proposes the use of every instance in positive bags as a starting point and record the distinct maximizers that are obtained after the gradient search. So, the instances selected as positive prototypes must be different from each other and have a high DD value. Given a set M that contains all local maximizers of $DD(h, D)$, in every iteration, DD-SVM selects the element from M that has the maximum $log(DD)$ value (equivalently the DD value).

$$(h^*, w^*) = \arg \max_{(p,q) \in M} log(DD(p, q)) \tag{2.17}$$

Then, DD-SVM removes instances that are close to the selected instance or have a low DD value. The equations defined by the DD-SVM algorithm to verify these two conditions are

$$\|p \otimes abs(q) - h^* \otimes w^*\| < \beta \|h^* \otimes w^*\| \ , (p, q) \in M \tag{2.18}$$

$$log(DD(p, q)) < T \ , (p, q) \in M \tag{2.19}$$

with $\otimes$ denotes component-wise product, T is a threshold value and β is a parameter.
DD-SVM proposes a threshold value T defined as

$$T = \frac{\max_{(p,q) \in M} log(DD(p, q)) + \min_{(p,q) \in M} log(DD(p, q))}{2} \tag{2.20}$$

DD-SVM uses the same steps, after negating the labels of positive and negative bags, to learn the negative prototypes from the training data D. These prototypes can be relevant features to discriminate between positive and negative bags.

After collecting the positive and negative prototypes, DD-SVM calculates the bag feature using a nonlinear mapping. Given the collection of prototypes $C = \{(h_k^*, w_k^*), k = 1 \ldots |C|\}$ and a bag B_i, the bag feature of B_i is defined as

$$\phi(B_i) = [s((h_1^*, w_1^*), B_i), s((h_2^*, w_2^*), B_i), \ldots, s((h_{|C|}^*, w_{|C|}^*), B_i)]^T \tag{2.21}$$

with

$$s((h_k^*, w_k^*), B_i) = \min_{j=1,\ldots,n_i} \|b_{ji} - h_k^*\|_{w_k^*} \ , k = 1, \ldots, |C| \tag{2.22}$$

The final step of DD-SVM is to train a standard SVM model in the bag space using the mapped features and the labels of the bags.

2.3.2 MILES

In MILES [5], each instance in the training bags is a candidate for target concepts. So, MILES considers each instance from both negative and positive bags as a potential TC and selects an optimal subset of instances using a sparse SVM. The set of target concepts is defined as $C = \{b_{ji}, i = 1 \dots N_B, j = 1 \dots n_i\}$. The difference between MILES and DD (or EM-DD) is that the target concepts of DD is defined only for positive bags , whereas, in MILES target concepts can be related to either negative or positive bags. The similarity between a target concept $x_k \in C$ and a bag B_i is defined as

$$s(x_k, B_i) = \max_j exp\Big(- \frac{\|x_k - b_{ji}\|^2}{\sigma^2} \Big) \tag{2.23}$$

With σ^2 is a normalization factor.

A bag B_i is then embedded with coordinates $m(B_i)$ defined as

$$m(B_i) = [s(x_1, B_i), s(x_2, B_i), \dots, s(x_{|C|}, B_i)]^T \tag{2.24}$$

If a feature x_k produces a low similarity to some negative bags and a high similarity to some positive bags, this feature can be relevant to distinguish between the positive and negative bags. The embedding step results in a high-dimensional feature space when the number of instances in the training data is large. Using this embedding, many features may be irrelevant because some of the instances could not be discriminating in the classification of the bags, or similar to each other resulting in redundancy problem. After mapping, MILES uses the 1-norm SVM [31] to perform joint feature selection and classification. The predicted label of a bag B_i is defined as

$$\hat{y} = w^T m(B_i) + b \tag{2.25}$$

with w and b are the parameters of SVM model. MILES selects a subset of mapped features that is most relevant to the classification problem. The set of selected features is given as $\{s(x_k, .),\ k \in I\}$ where $I = \{k,\ |w_k| > 0\}$. The label of bag B_i is computed as

$$\hat{y} = sign\Big(\sum_{k \in I} w_k s(x_k, B_i) + b \Big) \tag{2.26}$$

2.3.3 MI-AdaBoost

MI-AdaBoost [21] introduces the positive and negative concurrency so that a bag is classified positive in the case of the presence of instances associated with positive concepts and the absence of instances associated with negative concepts. MI-AdaBoost considers each instance from both negative and positive bags as a potential TC and uses AdaBoost [33] to select the most discriminating subset of instances. In other words, the set of potential target concepts is defined as $C = \{b_{ji}, i = 1 \dots N_B, j = 1 \dots n_i\}$. The similarity between a target concept $x_k \in C$ and a bag B_i is a defined in (2.23), and the mapping of a bag B_i is as in (2.24).

Using this mapping, many features can be irrelevant as they are not discriminative in the classification of the bags, or redundant because of their similarity to other features. MI-AdaBoost uses AdaBoost [33] for feature selection and builds a strong classifier by combining weak classifiers. At each iteration, one weak classifier, corresponding to the most discriminating feature, is selected and the training samples are re-weighted to give more importance to the misclassified samples. The final strong classifier is a weighted combination of the weak classifiers, and is defined as

$$H(x) = sign\Big(\sum_{t=1}^{T} w_t(x) h_t(x)\Big) \tag{2.27}$$

In (2.27), $h_t(x)$ is the weak classifier used for the selected feature at iteration t, $w_t(x)$ is the weight of weak classifier t, and T is the number of weak classifiers.

In the AdaBoost procedure, one dimension is selected at each iteration, so the number of iterations T determines the number of relevant features (or weak classifiers).

2.4 Multiple Instance Regression Algorithms

In contrast to MIC, in MIR there is no notion of positive/negative bags and target concepts. MIR aims to learn a regression model that maps each bag to a real-valued output. Some MIR algorithms assume that all instances are relevant and can be used to train a regression model. Other methods map each bag to one instance either by trying to select the instance that is responsible of the bag's label, or by aggregating the instances of every bag. Other algorithms select a subset of primary instances per bag or assign weights to the bag's instances using optimization techniques. In the following subsections, we provide an overview of existing MIR algorithms.

2.4.1 Aggregated-MIR

The aggregated-MIR [11] represents each bag by a single meta-instance, typically the mean of all the bag's instances. Figure 2.1 illustrates an example of meta-instances used to learn a regression model. In this example, instances from each bag are plotted by a different color. The data consist of five bags containing some number of one-dimensional items. The Aggregated-MIR represents each

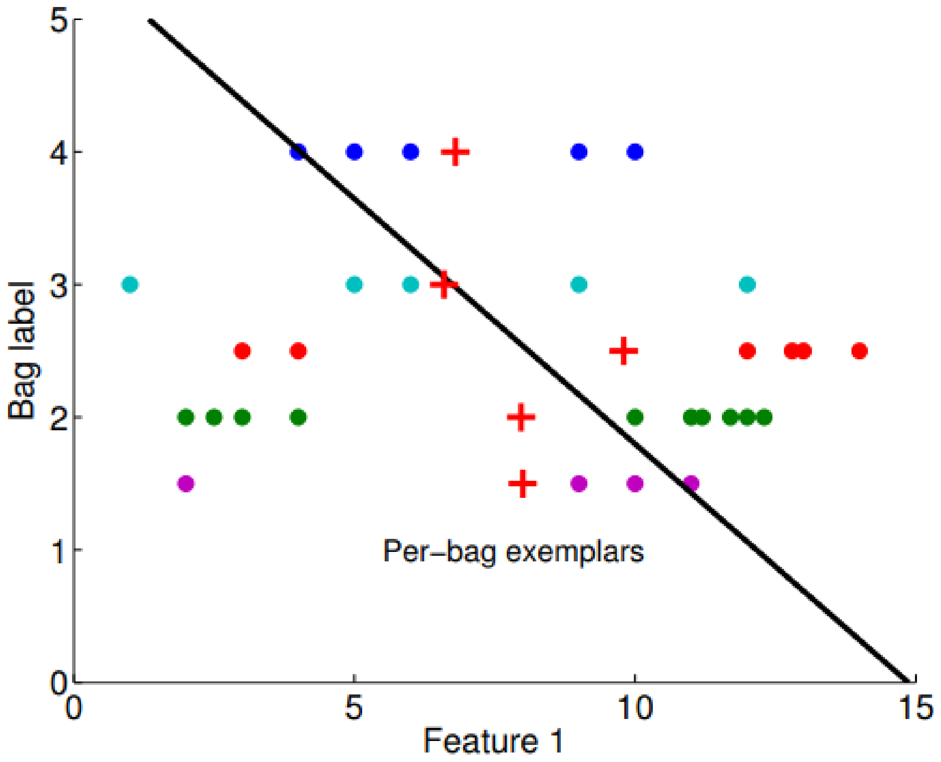


Figure 2.1: Illustrative example of Aggregated-MIR using a 1-dimensional data set with 5 bags. The X-axis represents the feature and the Y-axis represents the bag's label. Instances from one bag are displayed with the same color. The '+' represents the examplar of every bag (mean of all the bag's instances). Black line depicts the learned regression model.

bag by one examplar, denoted using '+', which is the mean of all the bag's instances. Then, a model is learned by applying traditional regression techniques on the meta-instances. In figure 2.1, the black line depicts the learned regression model. To predict the label of a new test bag, the bag's meta-instance is used as input to the learned model. This approach is reliable when all instances within a bag represent the "true instance" with small deviations.

2.4.2 Instance-MIR

Instance-MIR [11] propagates the bag label to all of its instances and then uses all instances and traditional regression techniques to learn the model. This approach assumes that every item in a bag has the same relationship to the bag's label, which can be modeled with a single global model. Figure 2.2 illustrates the Instance-MIR using an example of global model learned by fitting all the data. To predict the label of a new test bag, first, the label of each instance is predicted using the learned regression model. Then, the labels of all instances are aggregated using the mean or median to predict the label of the bag. This approach can fail when many instances within each bag are not

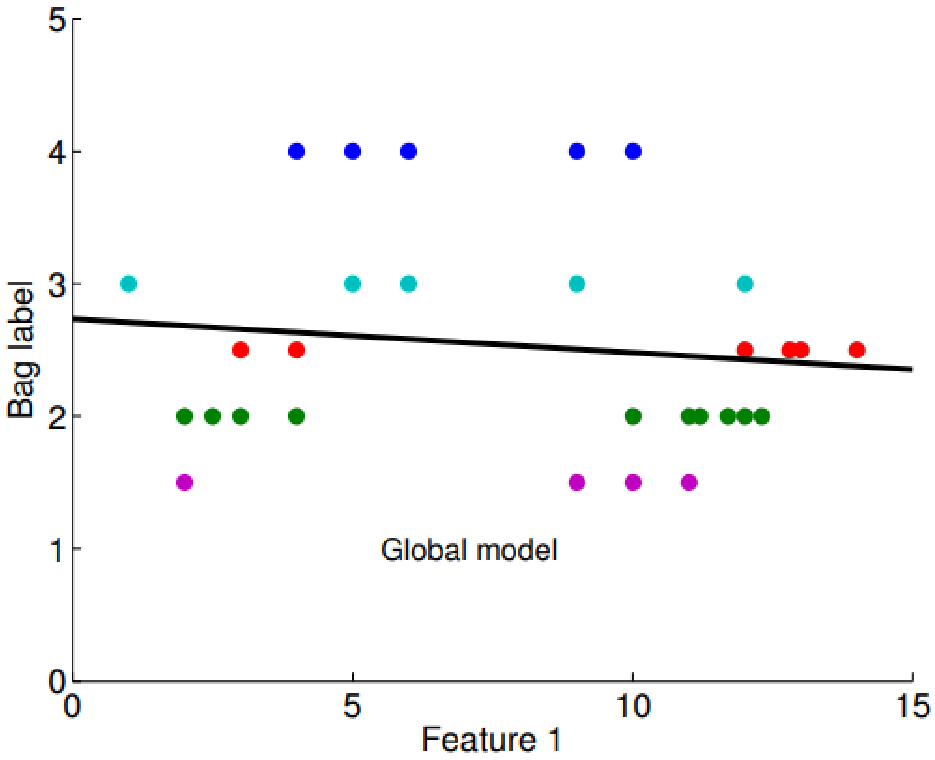


Figure 2.2: Illustrative example of Instance-MIR using a 1-dimensional data set with 5 bags. The X-axis represents the feature and the Y-axis represents the bag's label. Instances from one bag are displayed with the same color. Black line depicts the learned regression model.

representative of the ”true instance”.

2.4.3 Primary instance regression

Primary instance regression (PIR) [26] is one of the earliest MIR that maintains the bag structure. PIR assumes that the label of each bag is determined by a single instance, called primary instance (i.e ”true instance”), and that the rest of the instances in the bag are noisy observations. PIR operates on a given training data with N_B bags. The i^{th} bag consists of n_i instances, b_{ji}, $j = 1 \dots n_i$ where $b_{ji} \in \mathbb{R}^d$ and a real valued class label y_i. The objective of PIR is to find a hyper-plane, a^*, such that

$$a^* = \arg\min_a \sum_{i=1}^{N_B} L(y_i, b_{pi}, a) \tag{2.28}$$

where b_{pi} describes the primary instance of bag i, and L is some error function measuring the fitting error of the hyper-plane a with respect to each instance. The primary instances of the bags are also unknown, thus , finding the hyper-plane that minimizes (2.28) is not possible. Alternatively, in PIR, the optimal hyper-plane a is learned using

$$a^* = \arg\min_a \sum_{i=1}^{n} \min_j L(y_i, b_{ji}, a), 1 \leq j \leq n_i \tag{2.29}$$

with

$$L(y_i, b_{ji}, a) = (y_i - a^T b_{ji})^2 \tag{2.30}$$

PIR is an iterative algorithm that uses an Expectation-Maximization (EM) based approach to alternate between selecting the most probably primary instances and fitting a linear regression model to the selected instances. These steps are repeated until the convergence criteria is reached. PIR is not deterministic as different starting hyper-planes can lead to different regression models. To overcome this randomness, multiple runs of PIR algorithm, with random restarts, are recommended. The final regression model is selected as the one that has the lowest fitting error.

PIR does not provide a mechanism to predict the label of a new unlabeled test bag. Typically, as in the instance-MIR, the predicted value of a test bag is an aggregation (e.g., min, max, mean, or median) of its instances output.

2.4.4 EM-MIR

EM-MIR [11] is another multiple instance regression algorithm that assumes that each bag contains a primary instance that determines its label. EM-MIR treats the label of a bag as a random variable given by a mixture model:

$$p(y_i|B_i) = \sum_{j=1}^{n_i} \pi_{ji} p(y_i|b_{ji}) \tag{2.31}$$

In (2.31), n_i is the number of instance in the i^{th} bag B_i, b_{ji} is the j^{th} instance of B_i, y_i is the label of B_i, $p(y_i|b_{ji})$ is the label probability when the j^{th} instance is the primary instance and π_{ji} is the prior probability that the j^{th} instance is the primary instance of the i^{th} bag. Thus, the contribution of each instance to its bag's label is proportional to its probability of being the primary instance. The learning problem in EM-MIR is to determine π_{ji} and $p(y_i|b_{ji})$ from training data. EM-MIR assumes that both probabilities are parametric functions and denotes them as $p(y_i|b_{ji}, \theta_p)$ and $\pi_{ji}(\theta_g)$. The mixture model from (2.31) can be rewritten as:

$$p(y_i|B_i, \theta) = \sum_{j=1}^{n_i} \pi_{ji}(\theta_g) p(y_i|b_{ji}, \theta_p), \tag{2.32}$$

Where $\theta = (\theta_g, \theta_p)$ represents the model parameters. TheEM algorithm is then used to maximize (2.32) and learn both the prediction function parameters and the prior function.

EM-MIR starts with an initial guess of θ and then updates it by an alternation between an expectation (E) step and a maximization (M) step until the algorithm converges. In the Expectation step, the algorithm calculates the expected value of the log-likelihood given a fixed value of the parameter θ. In the Maximization step, the algorithm updates the parameters θ of the model in order to maximize the log-likelihood's expected value.

EM-MIR can include different choices for the prior probability. It can be a deterministic function based on domain knowledge so that there is no need to learn the parameter θ_g. It can be also a function of prediction deviation and assigns higher probability value to instances with closer prediction to the median prediction of a given bag. Alternatively, it can also be general parametric function. In [11], the authors propose a non-linear function, represented by a feed-forward neural network, to define the parametric function of the prior probability. In this case, the parametric function needs a training step.

2.4.5 MI-ClusterRegress

MI-ClusterRegress [34] is a different approach to MIR that uses clustering to reduce MIR to a standard regression problem. It is motivated by the fact that bags can contain instances drawn from a number of different data distributions.

MI-ClusterRegress uses a clustering step to group instances of all bags into a predefined number of clusters. Instances that are relevant to each cluster, called exemplars, are identified and used to build a local model for each cluster using standard SI regression techniques. The next step consists of selecting the best model. In general, model selection methods seek a tradeoff between minimum error and model complexity. MI-ClusterRegress proposes two heuristics to select the best model. The first one is based on minimizing the number of used Support Vectors in order to minimize the complexity of the learned model. The second one is based on minimizing the training error and looks for the best fit to a given training data. The best cluster, depending on which heuristic is used, is identified as the prime cluster and considered as the model responsible for the bags' labels. The training procedure of MI-ClusterRegress is summarized in Algorithm 2.1.

MI-ClusterRegress can transfer what is learned on the training data about relevance of instances, to the test bags, using the cluster models. This feature provides a mechanism to assign appropriate relevance values and predict the label of an unseen bag. The assigned label is an aggregation of the labels of the testing bag's instances that belong to the prime cluster. The predicting procedure of MI-ClusterRegress is summarized in Algorithm 2.2.

Algorithm 2.1 The MI-ClusterRegress training Algorithm

```
procedure MI-ClusterRegress($D$,$Y$,$k$)
    **Inputs:**
 Training data $D = \{B_j, j = 1 \ldots N_B\}$ and $Y = \{y_j, j = 1 \ldots N_B\}$, number of clusters $k$
    **Outputs:** learned regression parameters $f'$ and cluster parameters $\theta'$ for the best local model.

    $X = \bigcup_{i=1,\ldots,N_B} B_i$ : concatenation of all bags into single set.
    $\theta_{i=1,\ldots,N_B} = Cluster(X, k)$ : Cluster all items into $k$ clusters.
    for $i$ =1 to $N_B$ do
        for  $j = 1$ to $k$ do
            $R = relevance(B_i, \theta_j)$: Relevance vector of instances in $B_i$ with respect to cluster $j$
            $\hat{B_j^i} = B_i R$: exemplar of $B_i$ in cluster $j$.
        end
    end
    for  $j = 1$ to $k$ do
        $f_j = Regress(\{\hat{B_j^i}\}_{1,\ldots,N_B}, Y)$: Regression model for cluster $j$
    end
    $[f', \theta'] = Select(\{f_j, \theta_j\}, \{\hat{B_j^i}\}_{1,\ldots,N_B}, Y)$: Select the best local regression model.
```

Algorithm 2.2 The MI-ClusterRegress predicting Algorithm

```
procedure MI-ClusterRegress-Predict($B^T$,$f'$,$\theta'$)
    **Inputs:** New test bag $B^t$, regression model parameter $f'$, cluster parameters $\theta'$.
    **Outputs:** Prediction for $B^t$: $\hat{y}(B^t)$
    $R = relevance(B^t, \theta')$: Relevance vector of instances in $B^t$ with respect to cluster parameters $\theta'$
    $\hat{B^t} = B^t R$: exemplar of $B^t$.
    $\hat{y}(B^t) = RegressPredict(\hat{B^t}, f')$: Prediction of the label of $B^t$ using the examplar $\hat{B^t}$ and the regression parameters $f'$.
```

The potential drawback of MI-ClusterRegress is that clustering is performed in an unsupervised manner, without considering the bag labels. Moreover, it assumes that all primary instances will be grouped into one cluster, which is usually not the case specially in high dimensional feature spaces.

2.4.6 Pruning-MIR

In Instance-MIR, all instances are used to train the regression model. In contrast, in PIR, only one selected instance per bag is used to train a regression model that is used to predict the label of an unseen bag. Pruning-MIR [35] is a compromise between the two extremes. It uses an EM procedure to select a subset of primary instances from every bag and to train a regression model using the selected instances. In each E step, a small fraction of the instances, with the highest noise, is discarded. In the M step, the remaining instances are used to train a new predictor. Using this mechanism, the noisy instances are gradually removed while the remaining instances are used to build a new regression model. The algorithm keeps iterating as long as the prediction accuracy on the training data keeps improving. The accuracy is defined as the Mean Squared Error (MSE) of bag label predictions, i.e.,

$$MSE = \frac{1}{N_B} \sum_{j=1}^{N_B} (y_j - med(f(b_{ij}), i = 1, \ldots, n_j))^2 \tag{2.33}$$

In (2.33), N_B is the number of the bags, y_j is the label of the j^{th} bag, b_{ij} is the i^{th} instance of the j^{th} bag, n_j is the number of instances per bag, f is a regression function and med is the median (The mean can be used instead of the median).

Pruning-MIR describes the noisiest instances using two approaches. The first one is a global pruning, where, a regression model is trained and $r\%$ of the instances with the highest prediction error are discarded. The pruned data set is used to train a new regression model and the procedure is repeated until the algorithm converges. Global pruning can lead to imbalanced bags. In addition, the chance of obtaining a predictor that is different than the initial one is very low.

To overcome these difficulties, Pruning-MIR introduces a second approach, called a Balanced Pruning. In this approach, the algorithm discards $r\%$ of the noisiest instances from each bag. In addition, the instances, whose predictions are far away from the median prediction over the non-pruned instances of a bag, are discarded.

2.4.7 AP-Salience Algorithm

The AP-Salience algorithm [36] optimizes the contribution of each instance in each bag to the bag's label. It computes the best salience values assigned to instances under a fixed regression model. Then, given the fixed salience of items, the regression model is updated in order to minimize a given objective function.

The exemplar of a bag B_i is a convex combination of the bag's items and is defined as

$$H^i = \sum_j \alpha_j^i b_{ji} \tag{2.34}$$

In (2.34), H^i is the examplar of bag B_i, b_{ji} is the j^{th} instance of B_i, α_j^i is a constant value that indicates how salient item b_{ji} contributes to predicting y_i for bag B_i, where $\sum_j \alpha_j^i = 1$ and all $\alpha_j^i \geq 0$.

The algorithm minimizes an $L2$ loss function, with regularization terms $\epsilon 1$ and $\epsilon 2$ on α^k and a vector of regression weights W. That is, it minimizes

$$
\begin{aligned}
&\min_{W,\{\alpha^k\}_{k=1,\ldots,N_B}} f(W,\{\alpha^k\}_{k=1,\ldots,N_B}) = \sum_{k=1}^{N_B}[(y_k - W^T B_k \alpha^k)^2 + \epsilon 1\|\alpha^k\|^2] + \epsilon 2\|W\|^2 \\
&\text{subject to : } \alpha_i^k \geq 0 \ \ \forall i,k; \ \ \sum_{i=1}^{n_k} \alpha_i^k = 1, \ \ \forall k
\end{aligned}
\tag{2.35}
$$

In (2.35), N_B is the number of training bags and n_k is the number of instances in the k^{th} bag and the minimization of the objective function is with respect to W and $\{\alpha^k\}_{k=1,\ldots,N_B}$.

The exact solution of (2.35) is NP-hard. AP-Salience Algorithm proposes an alternating projection [36] method to find W and $\{\alpha^k\}_{k=1..m}$ that minimizes (2.35).

In contrast to Primary Instance Regression (PIR), that selects only a single instance from each bag as the primary instance, AP-Salience Algorithm gives the possibility to each instance in the bag to contribute to the bag's label by a weighted amount.

The AP-Salience algorithm does not provide a mechanism for testing where both the bag labels and items relevance are unknown.

CHAPTER 3

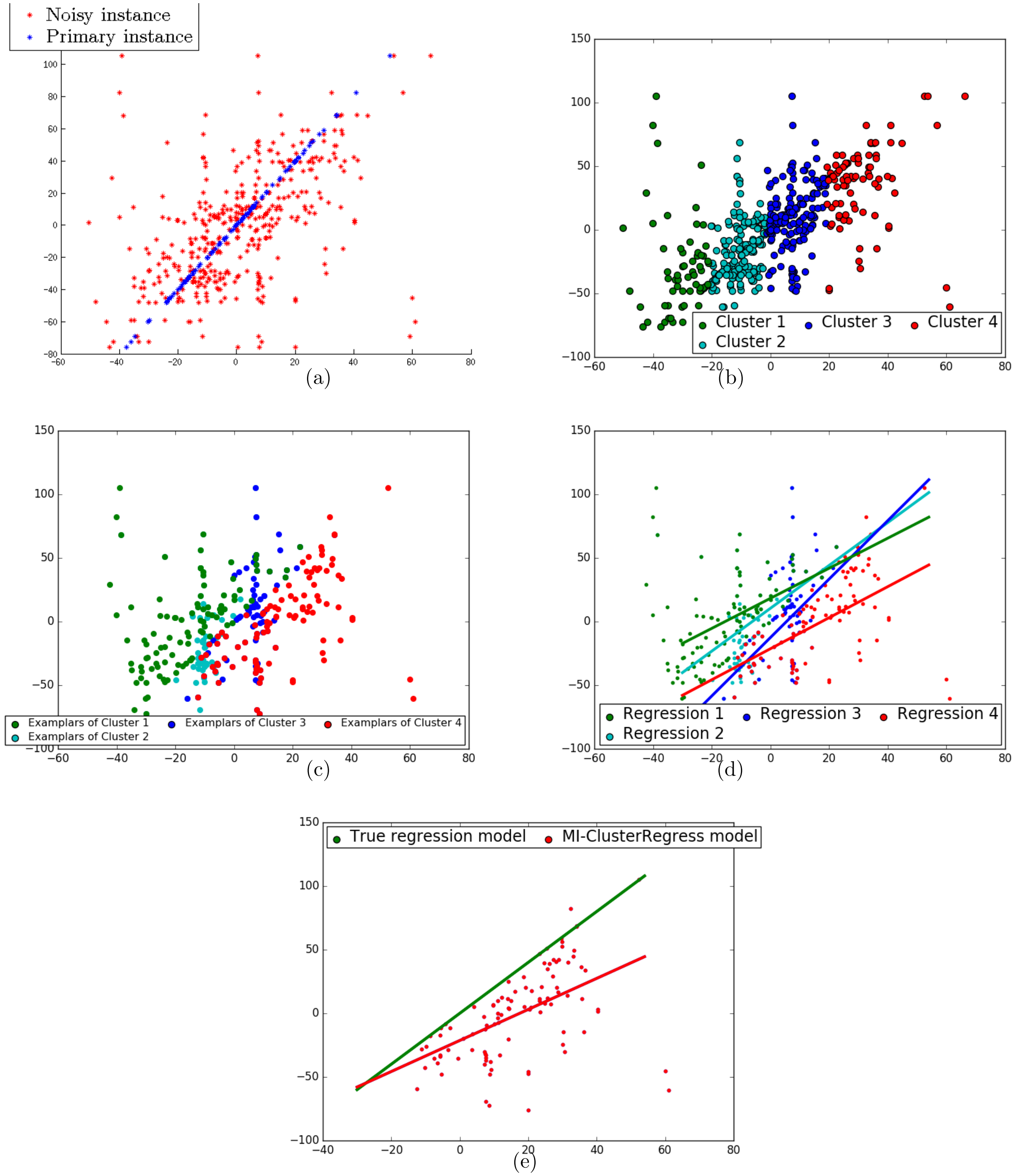
ROBUST CLUSTERING TO LEARN MULTIPLE REGRESSION MODELS

In this chapter, we describe our novel approach for Multiple Instance Regression. Let $D = \{B_j, j = 1 \dots N_B\}$ be a collection of N_B bags, where $B_j = \{(b_{ij}, y_j), i = 1 \dots n_j\}$, $b_{ij} \in \mathbb{R}^d$ is the attribute vector representing the ith instance from the jth bag, y_j is the real-valued target value of the jth bag and n_j is the number of instances in the j^{th} bag. The instances b_{ij} that determine the label y_j, called primary instances, are unknown. The objective of MIR is to identify primary instances in every bag, learn the regression model, and be able to predict the label of previously unseen bags. We start the description of our approach by a motivating example.

3.1 Motivating Example

First, we motivate our clustering-based approach by using a simple 1-D data that has 100 bags and each bag has 5 instances. We use this data to illustrate the MI-ClusterRegress algorithm and its shortcomings.

The data is displayed in figure 3.1(a) where the x-axis represents the 1-D feature of the instances and the y-axis represents the label of each bag (all instances in one bag have the same y value as they share the same label). All primary instances are displayed as blue dots and the remaining instances are displayed as red dots. Recall that in MIR this information is not available, and that we use it here for illustrative purposes only. The first step in MI-ClusterRegress is to partition all instances into k clusters. We use the K-Means algorithm [37] for this example and we let $k = 4$. The resulting partition is shown in figure 3.1(b). Since the K-Means was applied to the instance features only, it simply partitions the x-axis into 4 intervals. The next step in MI-ClusterRegress is to identify the closest instance from each bag to each cluster center. These instances are referred to as examplars. In figure 3.1(c), we show examplars of each cluster. The basic assumption in MI-ClusterRegress is that the examplars of one of the clusters will correspond to the primary instances of all bags. However, comparing the primary instances in figure 3.1(a) to the examplars in figure 3.1(c), we notice that this is not the case. The next step in MI-ClusterRegress



(a) (b) (c) (d) (e)

Figure 3.1: Illustration of the MI-ClusterRegress algorithm [34] to learn a regression model from multiple instance data. (a) Multiple instance data. Each bag has one primary instance (blue dots) and 4 noisy instances (red dots). (b) The 4 clusters obtained after partioning all instances. (c) Examplars of the 4 clusters. (d) Regression models learned using examplars of every cluster. (e) True regression model vs model learned using MI-ClusterRegress.

is to fit a linear regression model to the examplars of each cluster (as shown in figure 3.1(d)) and identify the cluster that has the smallest error fit. For this example, cluster 4 was selected. However, as illustrated in figure 3.1(e), the learned regression model is quite different from the true model used to generate the data.

In the above example, MI-ClusterRegress failed to learn the correct regression model because the assumption that most examplars of one of the clusters will correspond to the true primary instances did not hold. This assumption will be harder to maintain as the dimensionality of the feature space increases.

3.2 Robust clustering for MIR

To overcome the limitations of MI-ClusterRegress, we propose a new approach, called Robust Fuzzy Clustering for Multiple Instance Regression (RFC-MIR). RFC-MIR performs clustering and multiple model fitting simultaneously. Compared to MI-ClusterRegress, RFC-MIR has four additional properties. First, instead of using clustering to partition the instances in the feature space regardless of the labels of their bags, we combine the features and labels and use clustering, with an appropriate distance, to identify multiple local regression models. Second, we use a robust clustering approach so that non-primary instances (that incorrectly inherit the label of the bag they belong to) can be treated as noise and outliers to minimize their influence on the learned regression parameters. Third, we use fuzzy clustering so that each instance can contribute to each local regression with a fuzzy membership degree. Finally, we use properties of the possibilistic memberships to find the optimal number of regression models.

Let $x_{ji} = [b_{ji}, y_i] \in \mathbb{R}^{d+1}$ represent the concatenation of the j^{th} instance from the i^{th} bag and the label of its bag. Recall that labels are not available at the instance level and that y_i is valid only for the primary instances of bag i. Thus, many of the x_{ji}'s can have an irrelevant y_i. We combine x_{ji} from all training bags into $D = \{x_{ji}, i = 1 \ldots N_B, j = 1 \ldots n_i\}$. To simplify notation, we assume that all bags have the same number of instances $n_i = n$ for $i = 1 \ldots N_B$, and we rewrite $D = \{x_i, i = 1 \ldots N\}$, where $N = n \times N_B$. Next, we show how clustering could be used to identify the primary instances from all of the N instances and learn the MIR models simultaneously.

The fuzzy c-means (FCM) [38] algorithm minimizes

$$J_F = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij}^F)^m dist_{ij}^2 \tag{3.1}$$

In (3.1), C is the number of clusters, $dist_{ij}$ is the distance from x_j to cluster i, $m > 1$ is a weighting exponent called the fuzzifier and u^F_{ij} is the fuzzy membership of x_j in cluster i and satisfies the constraint:

$$u^F_{ij} \in [0,1] \text{ for all i,j; and } \sum_{i=1}^{C} u^F_{ij} = 1 \text{ for all j .} \tag{3.2}$$

The distance $dist_{ij}$ used in (3.1) controls the type and shape of clusters that will be identified. Various distances have been proposed to identify ellipsoidal, linear, and shell clusters such as lines, circles, ellipses, and general quadratics [39, 40]. In this work, we assume that the underlying regression model is linear and we use (3.1) to identify multiple linear models. In particular, we use a generalization of the distance in [41, 42] and let:

$$dist^2_{ij} = \sum_{k=1}^{d+1} v_{ik}((x_j - c_i) \cdot e_{ik})^2 \tag{3.3}$$

where c_i is the center of cluster i, e_{ik} is the k^{th} unit eigenvector of the covariance matrix Σ_i of cluster i. The eigenvectors are assumed to be arranged in ascending order of the corresponding eigenvalues λ_{ik}. In (3.3), we let

$$v_{ik} = \frac{\left[\prod_{j=1}^{d+1} \lambda_{ij}\right]^{\frac{1}{d+1}}}{\lambda_{ik}}, \tag{3.4}$$

that is, more importance will be given to distances projected on the eigenvectors associated with the smaller eigenvalues.

To optimize J_F with respect to the membership u^F_{ij} , we incorporate the constraints using Lagrange multipliers and obtain:

$$L^F = \sum_{i=1}^{C}\sum_{j=1}^{N}(u^F_{ij})^m dist^2_{ij} - \sum_{j=1}^{N}\xi_j(\sum_{i=1}^{C} u^F_{ij} - 1) \tag{3.5}$$

where $\Xi = [\xi_1, \ldots, \xi_N]$ is a vector of Lagrange multipliers corresponding to the N constraints in (3.2). The above optimization problem can be reduced to N simpler independent problems because the memberships of the different observations are independent of each other. Thus, for $j = 1, \ldots, N$, we minimize

$$L^F_j = \sum_{i=1}^{C}(u^F_{ij})^m dist^2_{ij} - \xi_j(\sum_{i=1}^{C} u^F_{ij} - 1) \tag{3.6}$$

To calculate u^F_{ij} , we set the derivative of L^F_j with respect to u^F_{ij} to zero and obtain

$$u^F_{ij} = \left(\frac{\xi_j}{m \times dist^2_{ij}}\right)^{\frac{1}{m-1}} \tag{3.7}$$

The Lagrange constant ξ_j could be solved using the constraint that $\sum\limits_{k=1}^{C} u_{kj}^F = 1$.
Doing so, we obtain

$$\left(\frac{\xi_j}{m}\right)^{\dfrac{1}{m-1}} = \sum_{k=1}^{C} \left(\frac{1}{dist_{kj}^2}\right)^{\dfrac{1}{m-1}} \tag{3.8}$$

After plugging (3.8) in (3.7), we obtain the fuzzy membership

$$u_{ij}^F = \left[\sum_{k=1}^{C} \left(\frac{dist_{ij}^2}{dist_{kj}^2}\right)^{\dfrac{1}{m-1}}\right]^{-1} \tag{3.9}$$

Optimization of (3.1) with $dist_{ij}$ in (3.3) subject to (3.2), using alternate optimization, results in an iterative algorithm that alternates between updating the fuzzy memberships using (3.9) and the center c_i and covariance Σ_i of cluster i using

$$c_i = \frac{\sum\limits_{j=1}^{N} (u_{ij}^F)^m x_j}{\sum\limits_{j=1}^{N} (u_{ij}^F)^m}, \tag{3.10}$$

and

$$\Sigma_i = \frac{\sum\limits_{j=1}^{N} (u_{ij}^F)^m (x_j - c_i)(x_j - c_i)^T}{\sum\limits_{j=1}^{N} (u_{ij}^F)^m}. \tag{3.11}$$

The objective function of the FCM in (3.1) is known to be sensitive to noise and outliers, and thus, is not suitable for the considered MIR application where we know a priori that the data is very noisy as non-primary instances and their labels should be treated as noise. Instead, we use the possibilistic c means (PCM) [38], which relaxes the constraint in (3.2) and minimizes

$$J_P = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij}^P)^m dist_{ij}^2 + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - u_{ij}^P)^m \tag{3.12}$$

In (3.12), $u_{ij}^P \in [0, 1]$ is a possibilistic membership degree that is not constrained to sum to 1 across all clusters. It is close to 0 for samples that are considered outliers, and close to 1 for inliers.
The above optimization problem can be reduced to an individual and independent optimization function because the memberships of the different observations in all clusters are independent of each other. Thus, for a given observation j in a given cluster i, we minimize

$$J_P^{ij} = (u_{ij}^P)^m dist_{ij}^2 + \eta_i (1 - u_{ij}^P)^m \tag{3.13}$$

To calculate u_{ij}^P , we set the derivative of J_P^{ij} with respect to u_{ij}^P to zero and obtain

$$u_{ij}^P = \frac{1}{1 + (\dfrac{dist_{ij}^2}{\eta_i})^{\dfrac{1}{m-1}}} \tag{3.14}$$

Optimization of (3.12) also results in an iterative algorithm that alternates between updating u_{ij}^P using (3.14) and the center c_i and covariance Σ_i as in (3.10) and (3.11) respectively. In (3.14), η_i is a cluster resolution parameter that could be fixed a priori or estimated using the distribution of the data within each cluster [38].

In [38], the authors define η_i using two rules. The first one makes η_i proportional to the average possibilistic intracluster distance of cluster i and is computed as:

$$\eta_i = K \frac{\sum\limits_{j=1}^{N} (u_{ij}^P)^m dist_{ij}^2}{\sum\limits_{j=1}^{N} (u_{ij}^P)^m} \tag{3.15}$$

In (3.15), K is a constant that is typically chosen to be 1. The second rule is based on an α-cut of cluster i, $(\prod_i)_\alpha$, that defines the set of vectors b_{ji} whose memberships is greater than α in cluster i. Thus, η_i is computed as:

$$\eta_i = \frac{\sum\limits_{b_{ji} \in (\prod_i)_\alpha} dist_{ij}^2}{|(\prod_i)_\alpha|} \tag{3.16}$$

Since the PCM does not constraint the memberships u_{ij}^P to sum to 1, it can result in several identical clusters. We use this feature to identify the optimal number of regression models [43]. We simply start with an over specified number of models, then identify and merge similar ones. Two models are considered similar and merged if

$$\frac{\sum\limits_{k=1}^{N} |u_{ik}^P - u_{jk}^P|}{\sum\limits_{k=1}^{N} |u_{ik}^P| + \sum\limits_{k=1}^{N} |u_{jk}^P|} < \theta_M \tag{3.17}$$

where θ_M is a threshold constant.

Currently, we assume that the underlying regression model is linear and thus, it can be captured by a single linear cluster. Consequently, if the algorithm identifies more than one cluster, say $c' > 1$, we need to select the "optimal" cluster, p. We propose two possible criteria to select this cluster. The first one is based on minimizing the fitting errors, i.e.,

$$p = arg \min_{i=1,\dots c'} \left\{ \varepsilon_i = \sum_{j=1}^{N} (u_{ij}^P)^m dist_{ij}^2 \right\} \tag{3.18}$$

An alternative approach is to select the cluster that covers the maximum number of bags. Let

$$\mathcal{P}^i = \{x_j, j = 1 \dots N \mid u_{ij}^P > threshold\} \tag{3.19}$$

be the set of inliers (i.e primary instances) assigned to cluster i, and

$$\mathcal{B}^i = \{B_k \mid x_j \in \mathcal{P}^i \text{ and } x_j \text{ is an instance of } B_k\}$$

be the set of bags that contribute to cluster i. The "optimal" cluster, p, can be identified as the one that has the largest number of unique bags in $\mathcal{B}^i$.

After identifying the optimal cluster, p, we let the primary instances of the data D be the primary instances of cluster p, i.e., $\mathcal{P} = \mathcal{P}^p$.

The linear regression model parameters can be identified from the cluster center c_p and covariance matrix $\varSigma_p$. Let $e_{min} = [e^1_{min}, \ldots, e^{d+1}_{min}]$ be the eigenvector associated with the smallest eigenvalue λ_{min} of $\varSigma_p$ and let $x = [x_1, \ldots, x_d, y] \in \mathcal{P}$ be a primary instance. The fact that x and c_p belong to the regression model leads to

$$e_{min}.(x - c_p) = 0,$$

or

$$e_{min}.x = e_{min}.c_p.$$

Decomposing x into the instance feature vector $[x_1, \ldots, x_d]$ and its label y, we obtain

$$e^{d+1}_{min} y + \sum_{k=1}^{d} e^k_{min} x_k = e_{min}.c_p$$

Solving for y, we obtain the regression model:

$$y = f(x) = \frac{e_{min}.c_{opt}}{e^{d+1}_{min}} - \sum_{k=1}^{d} \frac{e^k_{min}}{e^{d+1}_{min}} x_k \tag{3.20}$$

The resulting RFC-MIR algorithm is summarized in Algorithm 3.1.

3.3 Prediction Algorithm for RFC-MIR

Primary instances in the training data can be identified using (3.19) as the inliers, i.e points that have high possibilistic membership. For testing, this process is not as trivial since labels are needed to assign new memberships. Instead we use the following approach.

Let $B^t = \{x^t_1 \ldots x^t_N\}$ be a test bag with N instances. First, for each $x^t_i \in B^t$, we identify the closest primary instance (from training data) $x^P_i \in \mathcal{P}$. Then, we assume that y^P_i, the label of x^P_i, is a good initial estimate of the label of x^t_i and use $[x^t_i, y^P_i]$ to estimate the possibilistic membership u^P_i of x^t_i in the regression model f using (3.14). The primary instance of test bag B^t is identified as the instance that has the highest possibilistic membership, i.e.

$$x^t_{prim} = \{x^t_k \mid u^P_k = \max_{i=1 \ldots N} \{u^P_i\}\} \tag{3.21}$$

Finally, test bag B^t is labeled using

$$\hat{y}(B^t) = f\left(x^t_{prim}\right) \tag{3.22}$$

Algorithm 3.1 The RFC-MIR Algorithm

```
procedure RFC-MIR($D$,$C$,$m$)
    Inputs:
  Training data $D$, an overestimated number of clusters $C$,  fuzzifier $m$
    Outputs:  learned regression model $f$, set of primary instances $\mathcal{P}$

    Run FCM [41] for few iterations to get initial partition
    % get initial C distinct regression models
    for few (10) iteraions do
        update centers using (3.10)
        update covariance matrices using (3.11)
        update fuzzy memberships using (3.9)
    end
    % Refine $C$ models by ignoring noise and outliers
    repeat
        update centers using (3.10)
        update covariance matrces using (3.11)
        update possibilistic memberships using (3.14)
    until  centers and covariances do not change
    Merge similar clusters using (3.17)
    if number of remaining cluster $c' > 1$  then
        select "optimal" cluster using (3.18)
    Identify $\mathcal{P}$, the set of primary instances using (3.19)
    Identify regression model using (3.20)
```

Algorithm 3.2 The RFC-MIR-Predict Algorithm

```
procedure RFC-MIR-PREDICT($B^T$,$f$,$\mathcal{P}$)
    Inputs:  New test bag $B^t$, primary instances $\mathcal{P}$ from training data, learned regression model
  $f$.
    Outputs: Primary instance of $B^t$: $x^t_{prim}$, Prediction for $B^t$: $\hat{y}(B^t)$
    for each $x_i^t \in B^t$ do
        Find closest primary instance in $\mathcal{P}$, $x_i^P$, to $x_i^t$
         % $x_i^P$ has label $y_i^P$
        Approximate the label of $x_i^t$ with $y_i^P$
        Estimate $u^P(x_i^t)$ using $[x_i^t$,$y_i^P]$ in (3.14)
    end for
    Identify primary instance of $B^t$,$x^t_{prim}$, using (3.21)
    Label $B^t$ using (3.22)
```

We should note here that it is possible to select multiple primary instances for each test bag (e.g all instances with possibilistic membership above a threshold). In this case, the label of B^t can be taken as the average of the labels of all primary instances. The labeling algorithm is summarized in Algorithm 3.2.

CHAPTER 4

EXPERIMENTAL RESULTS

To validate the proposed MIR and evaluate its performance, we generate a series of synthetic multiple instance data sets with linear models. We vary the dimensionality of the feature space, the number of instances per bag, and the noise level. We compare the results of RFC-MILR with 4 existing MIR algorithms. These are the MI-Cluster Regress [34], the Instance-MIR and Aggregated-MIR [34,35], and the Primary-MIR [26]. The experiments were ran on a computer equipped with a 3.6 GHz Intel Xeon processor and a 24 GB RAM.

4.1 Synthetic datasets

4.1.1 Approach

First, we generate the instances features, $b_{ij} \in \mathbb{R}^d$, using

$$\mathbf{b}_{ij} = \mathbf{t}_i + \epsilon_{ij}^F, i = 1, \cdots, N_B, \quad and \quad j = 1, \cdots, n_i, \tag{4.1}$$

where $\mathbf{t}_i$ is the primary instance of bag, B_i, generated from a d-dimensional Gaussian distribution with zero mean and covariance $=10I^{d\times d}$. In (4.1), ϵ_{ij}^F is a noise term added to the features. It is generated using a normal distribution $\mathcal{N}^F(\mu^F{=}0, \sum^F{=}\sigma^F I^{d\times d})$. As the noise level increases, $\mathbf{b}_{ij}$ will divert from being a primary examplar to an irrelevant instance.

The label of each bag , B_i, is generated using

$$y_i = h(\mathbf{t}_i) + \epsilon_i^L, \tag{4.2}$$

where $h()$ is a linear d-dimensional function. We use

$$h(\mathbf{x}) = \sum_{k=1}^{d} a_k x_k \tag{4.3}$$

where a_k are constant coefficients. In (4.2), ϵ_i^L is a noise term, added to the true label. It is generated from a normal distribution $\mathcal{N}^L(\mu^L{=}0, \sigma^L)$.

Using the above strategy, we generate multiple data sets by varying:

1. The dimensionality of the feature space, d from 1 to 10.

2. The noise level added to the features in (4.1). We let

$$\sigma^F = k_1 \times \sigma_0^F, \tag{4.4}$$

   with σ_0^F=0.1 and k_1 varies from 1 to 100.

3. The noise level added to the bags' labels in (4.2). We let

$$\sigma^L = k_2 \times \sigma_0^L, \tag{4.5}$$

   with σ_0^L=0.05 and k_2 varies from 1 to 25.

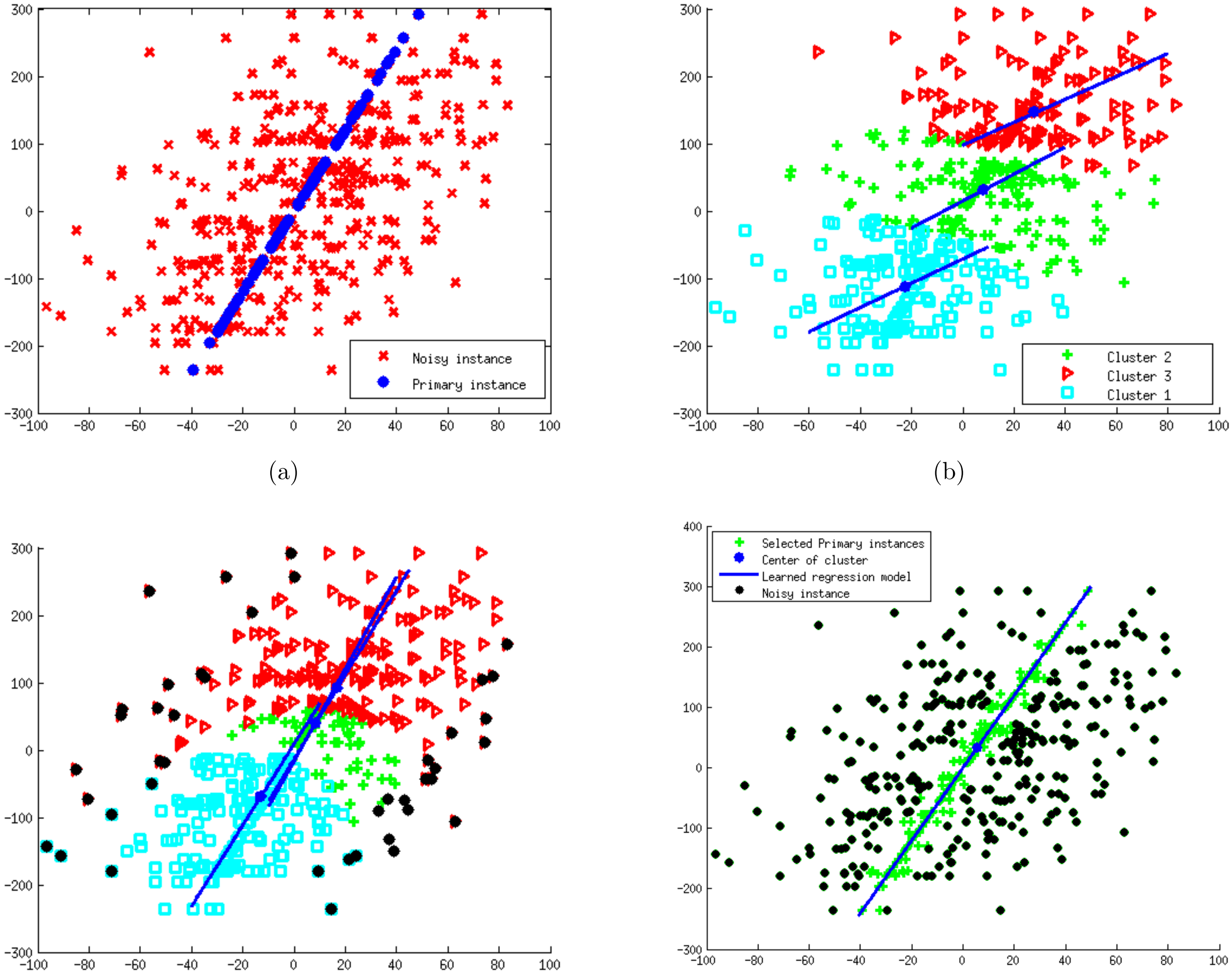
4. The number of instances per bag, n_i, from 5 to 100.

For each set of parameters, we create 10 linear models by generating random coefficients a_k (used in (4.3)). For each model, we generate one data collection that includes 100 bags, i.e. N_B=100.

4.1.2 Illustrative Example

First, we use a simple 1-Dim data to illustrate the different steps of the proposed MIR approach. The true model is $h(x) = 6x$, and each bag has 5 instances. For the noise levels, we use k_1=1000 and k_2=2.

The data is displayed in figure 4.1(a) where the x-axis represents the 1-D feature of the instances and the y-axis represents the label of each bag (all instances in one bag have the same y value as they share the same label). All primary instances are displayed as filled blue circles and the remaining ones are displayed as red 'x'. Recall that in MIR this information is not available, and that we use it here for illustrative purposes only. Using C=3, figure 4.1(b) displays the 3 initial clusters obtained after running the RFC-MILR for few iterations with fuzzy memberships. Points that belong to different clusters are displayed with different symbols and colors. Figure 4.1(c) displays the results after switching from fuzzy to possibilistic memberships and running the algorithm for 3 iterations. As it can be seen, RFC-MILR started identifying noisy instances (displayed as black circles) and the 3 linear clusters started converging to the same true model. Figure 4.1(d) displays the final results after the clusters became identical and got merged into one using (3.17). Points with high possibilistic memberships (> 0.9) are located along the linear model. These points will be considered the primary instances. All others, will be treated as irrelevant ones.



(a)

(b)

(c)

(d)

Figure 4.1: Illustrations of the steps of RFC-MILR: (a) Example of MIR data, (b) Result of fuzzy clustering, (c) Result of possibilistic clustering after 3 iterations, (d) Result of RFC-MILR after merging similar clusters

4.1.3 Results of Synthetic data sets

To compare the performance of the different MIR algorithms, for each data set, we compute the mean square error (MSE) using:

$$MSE = \frac{1}{N_B} \sum_{i=1}^{N_B} (y_i - \hat{y}(B_i))^2, \tag{4.6}$$

where y_i is the true label of bag i and $\hat{y}(B_i)$ is the label estimated using the different MIR algorithms.

For all data sets, we set the initial number of models C to 10, θ_M in (3.17) to 0.1, and fuzzifier m to 2. The value of η_i in (3.14) is estimated using the average fuzzy intra-cluster distance

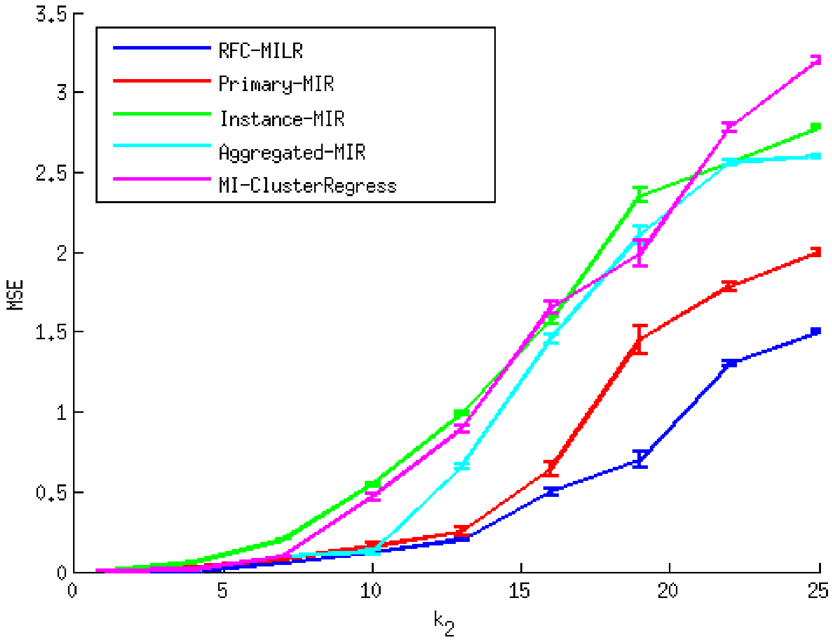


Figure 4.2: Comparison of RFC-MILR with previous MIR algorithms when varying the noise level added to the bags' labels in (4.2)

of cluster i as shown in (3.15).

In the following experiments, unless stated otherwise, we fix the k_1 value, used to control the level of noise added to the instances (4.4) to 10. We also fix k_2, used to control the noise added to bags' labels (4.5) to 10. The number of instances per bag, n_i, and the dimensionality of the instance space, d, to 5 and 1 respectively.

In the first experiment, we vary the noise level added to the bags' labels by increasing k_2 in (4.5) from 1 to 25. For each value of k_2, we generate 10 data sets using 10 linear models that use random coefficients a_k's (refer to (4.3)). The results of this experiment are displayed in Figure 4.2 where for each value of k_2, we display the mean MSE averaged over the 10 random models. We also display the variance of the MSE as a vertical error bar. As it can be seen, RFC-MILR has the lowest error. Moreover, the results of the 10 random models are consistent as indicated by the low MSE variations across the random models. In a second experiment, we vary k_1 from 1 to 100. The results are displayed in figure 4.3 where the proposed RFC-MILR has the lowest MSE average and variation.

In a third experiment, we vary the number of instances per bag, n_i from 5 to 100. In general, adding more instances increases the number of irrelevant instances and makes the MIR problem more challenging. The results of this experiment are displayed in Figure 4.4. As it can

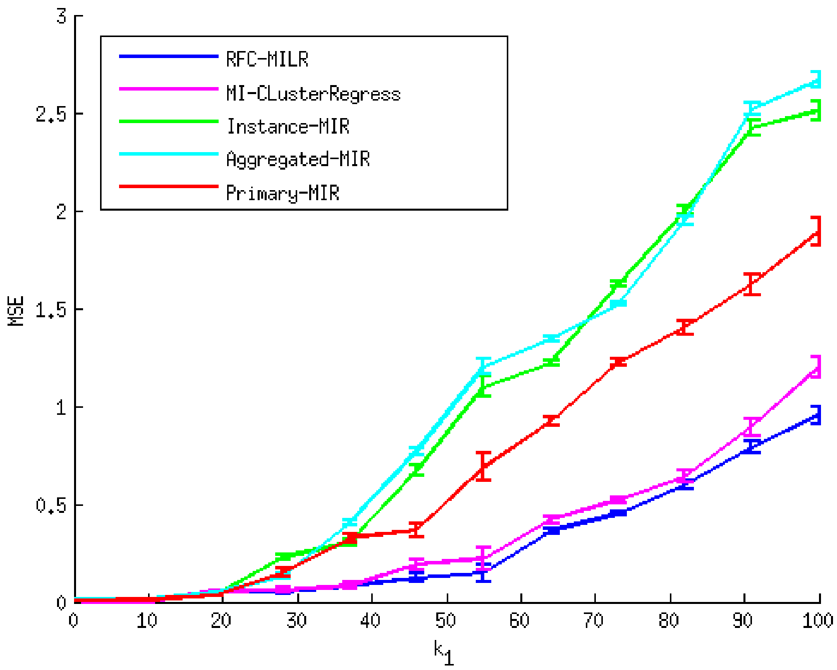


Figure 4.3: Comparison of RFC-MILR with previous MIR algorithms when varying the noise level added to the features in (4.1)

be seen, the proposed RFC-MILR algorithm is very robust even in the presence of a large number of irrelevant instances. On the other hand, for all other 4 algorithms the average MSE increases significantly as more irrelevant instances are included in each bag.

In a fourth experiment, we vary the dimensionality of the instances, d, from 1 to 10. The results are displayed in Figure 4.5. As for the previous experiments, the proposed RFC-MILR has the lowest MSE values.

4.2 Applications in remote sensing

A common application that has been used to validate MIR algorithms is the prediction of crop yield based on remote sensing observations [11, 34, 44, 45]. Predicting the yearly average yield of a crop per acre for a given region, especially when done early in the growing seasons, can be very beneficial. We use data collected by the MODIS instruments onboard satellites that provide cover of the entire US every 1-2 days [45]. In particular, we use the 8-day aggregate product which provides observations, in the red and near infrared (NIR), of each pixel location (250m×250m on the surface of the earth) every 8 days. The RED and NIR values are combined to generate the Normalized

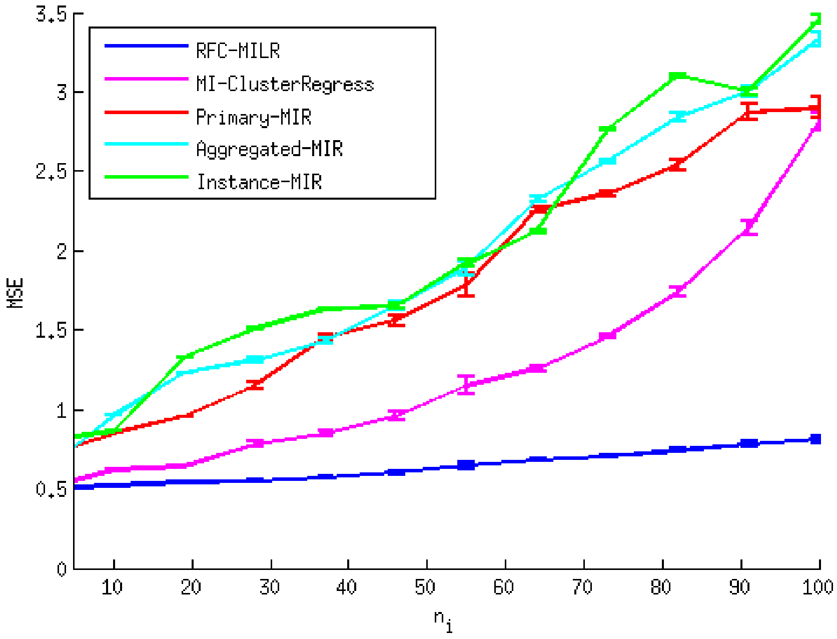


Figure 4.4: Comparison of RFC-MILR with previous MIR algorithms when varying the number of instances per bag

Difference Vegetation Index (NDVI) using:

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{4.7}$$

NDVI provides good indication of vegetation abundance and is good for identifying pixels that contain crops. Consequently, each pixel is represented by a time series where the i^{th} observation corresponds to the pixel's NDVI after $8 \times i$ days.

We use data from the California region over a period of 5 years (2001-2005) to predict the yield of corn and wheat in each county. The total number of counties that reported the yield for corn and wheat is summarized in table 4.1.

These are the same data sets used in [34] to validate MI-ClusterRegress. This application is challenging because each county contains thousands of pixels and we do not know which pixels (or even how many) contain the crop of interest. We use a randomly sub-sampled data such that 100

TABLE 4.1

Counties reporting yield for corn and wheat between 2001 and 2005

year	2001	2002	2003	2004	2005
number of counties	17	16	18	15	13

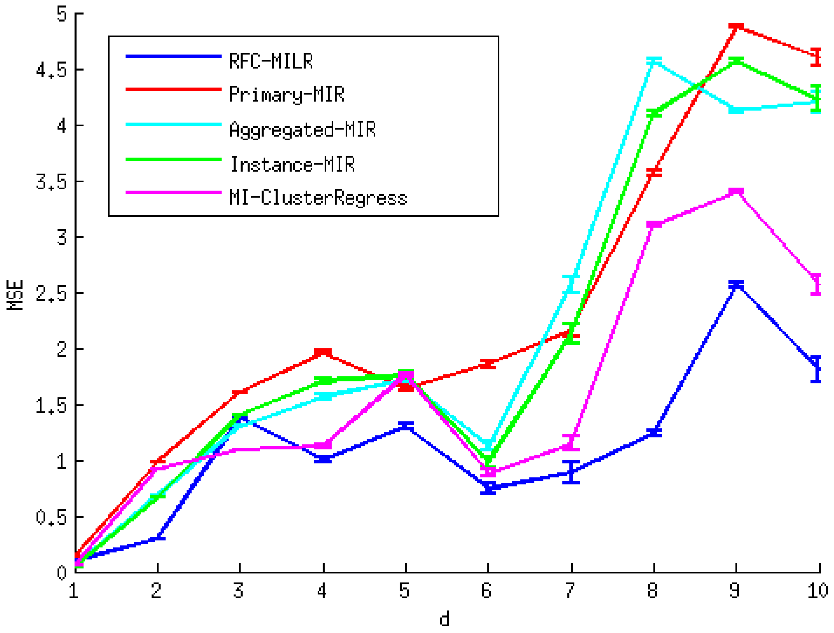


Figure 4.5: Comparison of RFC-MILR with previous MIR algorithms when varying the dimensionality of the feature space

pixels are selected for each county. Thus, each county is represented by one bag of 100 instances. Observations from the first 4 years (2001 2004) are used for training. The learned regression models are then used to predict the yield for 2005. Let f_D, for $D = 8, 16, \cdots, 360$ be the regression model to predict the yield at day D. f_D is trained with the sequence of NDVI observations taken every 8 days from the beginning of the year until day D. Thus, f_D will involve D/8-dimensional instance vectors. For each data, we run MI-ClusterRegress, Aggregated-MIR, Instance-MIR, Primary-MIR (PIR) and RFC-MIR 10 times and report the mean MSE and standard deviation of all runs.

Figure 4.6 compares the MSE of the five algorithms to predict corn yield and figure 4.7 compares the results to predict wheat yield. We only consider the days of the growing season (days 140-280 for corn and days 0-180 for wheat). As it can be seen for both crops, RFC-MIR provides more accurate and consistent prediction. The Instance-MIR algorithm was expected to perform poorly because every bag contains many irrelevant instances. It's the same case for Aggregated-MIR because the meta-instance obtained by the mean of bag's instances is influenced by the irrelevant instances. Primary-MIR algorithm presents a large variation as shown in both figures 4.6 and 4.7. This behavior can be explained by its sensitivity to initialization. Adding the label as a feature in RFC-MIR leads to better results comparing to MI-ClusterRegress as shown in both figures 4.6 and 4.7.

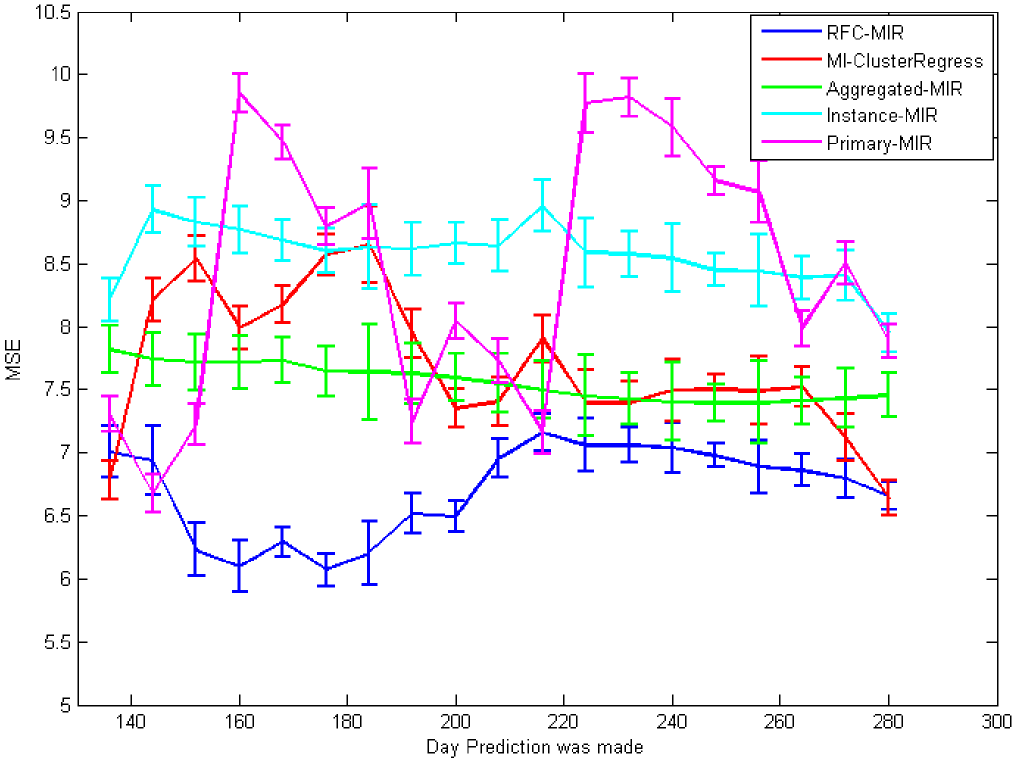


Figure 4.6: Comparison of the MSE of MI-ClusterRegress, Aggregated-MIR, Instance-MIR, Primary-MIR (PIR) and RFC-MIR for corn yield prediction

4.3 Applications to Drug Activity Prediction

A common application, in pharmaceutical industry, that has been used to validate MIR algorithms is the drug activity prediction [46]. This application, known as Quantitative Structure-Activity Relationships (QSAR) [47], is based on the concept that a biological effect of a given drug is a function of its chemical structure. Given a set of drugs with their possible structures and known affinities to a target protein, the objective is to predict the affinity to the target of a new drug. In other words, in the MIR settings, a drug is represented by a bag that contains all possible structures of this molecule. The bag's label is the affinity of the drug to a given target protein. However, the specific structure(s) that affects the label of each bag is unknown.

We use a dataset that consists of Thrombin inhibitors [48], that can be used as anti-coagulant. This dataset consists of 40 thrombin inhibitors. Each drug or inhibitor contains between 3 and 334 structures and is assigned a real valued affinity to a target protein. Each instance or structure is a 6 dimensional feature vector. It corresponds to a 4-point pharmacophore representation [49]. In this representation, the Euclidean distances between 4 different chemical groups are calculated, to give a $\binom{4}{2}$=6 feature vector.

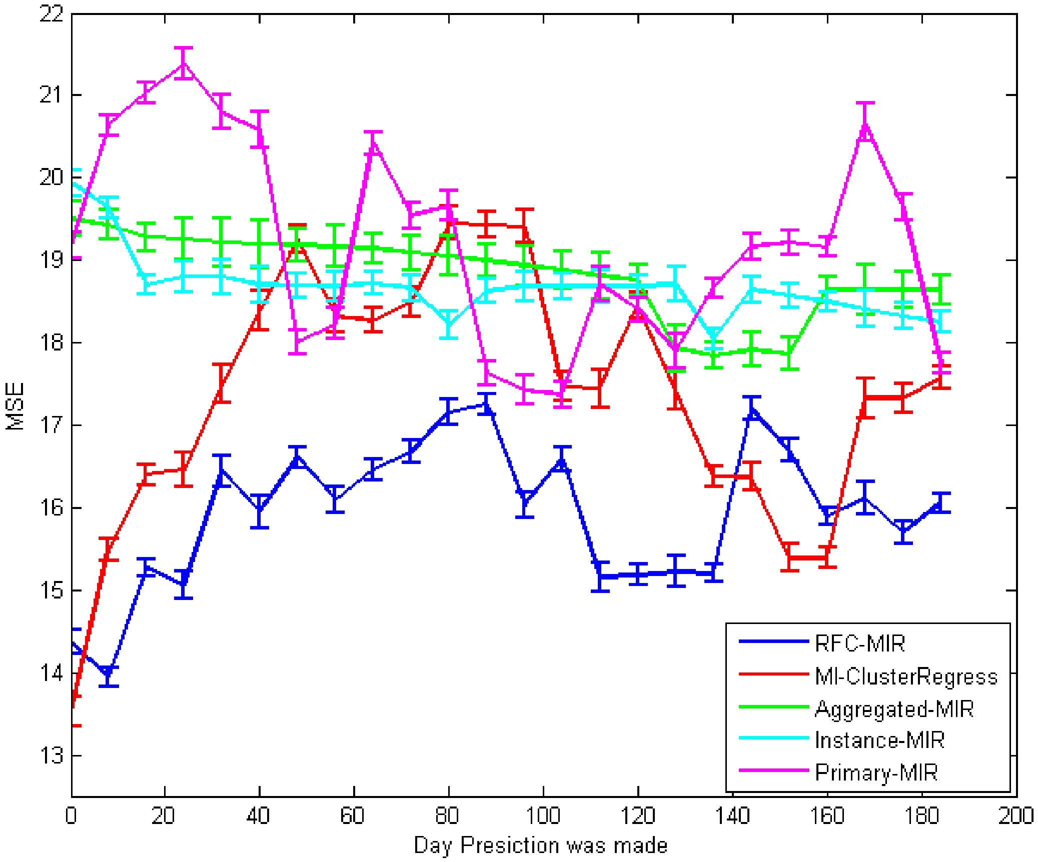


Figure 4.7: Comparison of the MSE of MI-ClusterRegress, Aggregated-MIR, Instance-MIR, Primary-MIR (PIR) and RFC-MIR for wheat yield prediction

TABLE 4.2

Comparison of the accuracy of the proposed RFC-MIR with other state of the art MIR methods on the Thrombin Inhibitors dataset .

Dataset	MI-Cluster Regress	Instance-MIR	Aggregated-MIR	Primary-MIR	RFC-MIR
Thrombin Inhibitors	3.89 ± 0.87	4.83±0.97	4.25 ± 0.77	3.92 ± 0.89	**3.74 ± 0.76**

We divide the data into training and testing using 5 fold cross validations. We run MI-ClusterRegress, Instance-MIR, Aggregated-MIR, Primary-MIR and RFC-MIR 10 times using random partitioning into train and test, and report the mean MSE and standard deviation of all runs.

The results are summarized in Table 4.2. As it can be seen, RFC-MIR outperforms other methods in term of MSE and standard deviation. Our method doesn't assume that a single structure is responsible for the drug's affinity to a target protein. It assumes that every instance can contribute to the molecule's activity with a possibilistic membership, unlike the Primary-MIR that assumes only one instance per bag is responsible for the bag's label. Instance-MIR and Aggregated-MIR have the worst performance because the high number of irrelevant conformations in some drugs leads to inaccurate learned regression model.

CHAPTER 5

CONCLUSIONS AND POTENTIAL FUTURE WORK

5.1 Conclusions

We proposed a new approach to multiple instance regression based on robust multi-model fitting. By combining the bags' instances and labels, and using an appropriate distance that measures the deviation of a point from a linear model, we showed that a possibilistic clustering algorithm can be used to estimate the regression model in a MIR setting. More importantly, we showed that the possibilistic memberships can be used to identify the primary instances and the irrelevant instances within each bag. Using several synthetic data sets with known structure and different levels of noise and difficulty, we showed that our approach achieves higher accuracy than state of the art methods. We have also validated our approach using a real application in remote sensing. Using a multiple instance data representation, we showed that RFC-MIR can be used to predict the yearly average yield of a crop for different region without the need to label the training data at the pixel level. We also showed that RFC-MIR can provide more accurate and consistent predictions than MI-ClusterRegress, Aggregated-MIR, Instance-MIR and Primary-MIR.

5.2 Potential Future Work

Currently, we assume that the regression model is linear and after clustering, we identify a single model that has instances from the maximum number of distinct bags and/or minimizes the fitting error. We are currently investigating two strategies to generalize our approach to non-linear regression. The first one is based on the assumption that a non-linear model can be approximated by multiple piecewise linear models. In this case, after convergence, instead of selecting the best model, we need to identify the multiple valid models and their domains. The second approach modifies the distance measure used within the clustering objective function to represent the fitting error with respect to a non-linear model.

REFERENCES

[1] T.G. Dietterich, R.H. Lathrop, and Tomás Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[2] Oded Maron, *Learning From Ambiguity*, Ph.D. thesis, Massassachusetts Institute of Technology, 1998.

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

[4] Rouhollah Rahmani and Sally A. Goldman, "MISSL: Multiple-instance semi-supervised learning," in *Proceedings of the 23rd international conference on Machine learning.* 2006, pp. 705–712, ACM.

[5] Yixin Chen, Jinbo Bi, and James Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006.

[6] Changbo Yang, Ming Dong, and Farshad Fotouhi, "Region based image annotation through multiple-instance learning," in *Proceedings of the 13th annual ACM international conference on Multimedia.* 2005, pp. 435–438, ACM.

[7] C. Zhang, X. Chen, and W. B. Chen, "An online multiple instance learning system for semantic image retrieval," in *Ninth IEEE International Symposium on Multimedia Workshops (ISMW 2007)*, 2007, pp. 83–84.

[8] A. Karem and H. Frigui, "A multiple instance learning approach for landmine detection using ground penetrating radar," in *2011 IEEE International Geoscience and Remote Sensing Symposium*, July 2011, pp. 878–881.

[9] Amine Khalifa and Hichem Frigui, "Fusion of multiple algorithms for detecting buried objects using fuzzy inference," in *Proc. SPIE*, 2014, vol. 9072, pp. 90720V–90720V–10.

[10] A. B. Khalifa and H. Frigui, "A multiple instance neuro-fuzzy inference system for fusion of multiple landmine detection algorithms," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2015, pp. 4312–4315.

[11] Z. Wang, L. Lan, and S. Vucetic, "Mixture model for multiple instance regression and applications in remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2226–2237, June 2012.

[12] Z. Fu, A. Robles-Kelly, and J. Zhou, "Milis: Multiple instance learning with instance selection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 958-977*, 5 2011.

[13] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *In S. Becker, S. Thrun, and K. Obermayer, Eds. Advances of Neural Information Processing Systems 15, Cambridge, MA: MIT Press, pp.561-568*, 2003.

[14] T. Grtner, P.A. Flach, A. Kowalczyk, and A.J. Smola, "Multi-instance kernels," in *In Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, pp.179-186*, 2002.

[15] Q. Zhang and S.A. Goldman, "Em-dd: an improved multi-instance learning technique," in *T.G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Advances in Neural Information Processing Systems 14, Cambridge, MA: MIT Press, pp.1073-1080*, 2002.

[16] J. Wang and J.-D. Zucker, "Solving the multiple-instance problem: a lazy learning approach," in *In Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA, pp.1119-1125*, 2000.

[17] Z.-H. Zhou and M.-L. Zhang, *Neural networks for multi-instance learning. Technical Report, AI Lab, Computer Science and Technology Department, Nanjing University, Nanjing, China*, 8 2002.

[18] M.-L. Zhang and Z.-H. Zhou, "Improve multi-instance neural networks through feature selection," in *Neural Processing Letters, vol.19, no.1, pp.1-10*, 2004.

[19] G.J. Qi, X.S. Hua, Y. Rui, T. Mei, J.H. Tang, and H.J. Zhang, "Concurrent multiple-instance learning for image categorization," *in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, June 2007.

[20] N. Weidmann, E. Frank, and B. Pfahringer, "A two-level learning method for generalized multi-instance problem," *in Machine Learning. Berlin, Germany: Springer*, vol. 2837, pp. 468–479, 2003.

[21] Richang Hong, Meng Wang, Yue Gao, Dacheng Tao, Xuelong Li, and Xindong Wu, "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE TRANSACTIONS ON CYBERNETICS*, vol. 44, no. 5, May 2014.

[22] Ethem Alpaydin, Veronika Cheplygina, Marco Loog, and David M.J. Tax", "Single- vs. multiple-instance classification," *Pattern Recognition*, vol. 48, no. 9, pp. 2831–2838, 2015.

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

[24] JooSeuk Kim and Clayton D. Scott, "Robust kernel density estimation," *Journal of Machine Learning Research*, vol. 13, pp. 2529–2565, 2012.

[25] W.H., Teukolsky, S.A., Vetterling, W .T., and B.P. Flannery, "Numerical recipes in c: the art of scientific computing," *Cambridge University Press, New York, second edition.*, 1992.

[26] Soumya Ray and David Page, "Multiple instance regression," in *Proceedings of the Eighteenth International Conference on Machine Learning.* 2001, ICML '01, pp. 425–432, Morgan Kaufmann Publishers Inc.

[27] A. Karem and H. Frigui, "Fuzzy clustering of multiple instance data," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Aug 2015, pp. 1–7.

[28] Veronika Cheplygina, David M.J. Tax, and Marco Loog, "Multiple instance learning with bag dissimilarities," *Pattern Recognition*, vol. 48, no. 1, pp. 264–275, 2015.

[29] Veronika Cheplygina, David M.J. Tax, and Marco Loog, "Dissimilarity-based ensembles for multiple instance learning," *arXiv:1402.1349v1 [stat.ML]*, February 2014.

[30] T. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[31] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," *in Advances in Neural Information Processing Systems*, vol. 16, no. 1, pp. 49–56, 2004.

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

[33] X. Yuan, X.S. Hua, G.J. Qi M. Wang, and X. Wu, “A novel multiple instance learning approach for image retrieval based on adaboost feature selection,” *in Proc. IEEE Int. Conf. Multimedia Expo*, pp. 1491–1494, July 2007.

[34] Kiri L. Wagstaff, Terran Lane, and Alex Roper, “Multiple-instance regression with structured data,” in *2008 IEEE International Conference on Data Mining Workshops*, Dec 2008, pp. 291–300.

[35] Zhuang Wang, Vladan Radosavljevic, Bo Han, Zoran Obradovic, and Slobodan Vucetic, “Aerosol optical depth prediction from satellite observations by multiple instance regression,” in *Proceedings of the 2008 SIAM International Conference on Data Mining*, 2008, pp. 165–176.

[36] Kiri L. Wagstaf and Terran Lane, “Salience assignment for multiple-instance regression,” *ICML 2007 Workshop on Constrained Optimization and Structured Output Spaces, Corvallis, OR*, 2007.

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

[38] R. Krishnapuram and J.M. Keller, “A possibilistic approach to clustering,” *Fuzzy Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 98–110, May 1993.

[39] H. Frigui and R. Krishnapuram, “A comparison of fuzzy shell-clustering methods for the detection of ellipses,” *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 193–199, May 1996.

[40] Frank Hppner, Frank Klawonn, Rudolf Kruse, and Thomas Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, John Wiley and Sons, England, 1999.

[41] James C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[42] Rajesh N. Dave, “Use of the adaptive fuzzy clustering algorithm to detect lines in digital images,” in *Proc. SPIE*, 1990, vol. 1192, pp. 600–611.

[43] Krishnapuram R. Frigui H., “A robust algorithm for automatic extraction of an unknown number of clusters from noisy data,” *Pattern Recognition Letters*, vol. 17, no. 12, pp. 1223–1232, 1996.

[44] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon, “Deep gaussian process for crop yield prediction based on remote sensing data,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 4559–4566.

[45] Kiri L. Wagstaff and Terran Lane, “Salience assignment for multiple-instance regression,” in *ICML 2007 Workshop on Constrained Optimization and Structured Output Spaces, Corvallis, OR, 2007*, 07 2007.

[46] Jesse Davis and Soumya Ray, “Tightly integrating relational learning and multiple-instance regression for real-valued drug activity prediction,” *International Conference on Machine Learning*, 2007.

[47] Xiao-Fei Zhou, Qingxiang Shao, Robert A. Coburn, and Marilyn E. Morris, “Quantitative structure activity relationship and quantitative structure pharmacokinetics relationship of 1,4-dihydropyridines and pyridines as multidrug resistance modulators,” *Pharmaceutical Research*, vol. 22, no. 12, December 2005.

[48] J. Cheng, C. Hatzis, H. Hayashi, Krogel M.-A., Morishita S., Page D., and Sese J, "Kdd cup 2001 report," *SIGKDD Exploration*, vol. 3, pp. 47–64, 2005.

[49] PAUL FINN, STEPHEN MUGGLETON, DAVID PAGE, and ASHWIN SRINIVASAN, "Pharmacophore discovery using the inductive logic programming system progol," *Machine Learning*, vol. 30, pp. 241–270, 1998.

CURRICULUM VITAE

NAME: Mohamed Trabelsi

ADDRESS:
Computer Engineering & Computer Science Department
Speed School of Engineering
University of Louisville
Louisville, KY 40292

EDUCATION:

M.S., Computer Science & Engineering
May 2018
University of Louisville, *Louisville, Kentucky*

B.Eng., Signals and Systems
June 2016
Tunisia Polytechnic School, *Tunis, Tunisia*

Research Interests:

Machine Learning, Computer Vision and Artificial Intelligence

Academic Experience:

Research Assistant in Multimedia Research Lab, CECS, University of Louisville

Publications:

M. Trabelsi and H. Frigui, Fuzzy and Possibilistic Clustering for Multiple Instance Linear Regression, FUZZ IEEE 2018 conference.