MULTIVARIATE DATA MODELING AND ITS APPLICATIONS TO CONDITIONAL OUTLIER DETECTION

by

Charmgil Hong

B.S.E., Handong Global University, 2010

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

THE DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Charmgil Hong

It was defended on

August 9, 2017

and approved by

Milos Hauskrecht, **PhD**, Professor, Department of Computer Science

Rebecca Hwa, **PhD**, Associate Professor, Department of Computer Science

Adriana Kovashka, **PhD**, Assistant Professor, Department of Computer Science

Gregory Cooper, **MD**, **PhD**, Professor, Department of Biomedical Informatics

Dissertation Director: **Milos Hauskrecht**, **PhD**, Professor, Department of Computer Science

MULTIVARIATE DATA MODELING
AND ITS APPLICATIONS TO
CONDITIONAL OUTLIER DETECTION

Charmgil Hong, PhD

University of Pittsburgh, 2017

With recent advances in data technology, large amounts of data of various kinds and from various sources are being generated and collected every second. The increase in the amounts of collected data is often accompanied by increase in the complexity of data types and objects we are able to store. The next challenge is the development of machine learning methods for their analyses. This thesis contributes to the effort by focusing on the analysis of one such data type, *complex input-output data objects with high-dimensional multivariate binary output spaces*, and two data-analytic problems: *Multi-Label Classification* and *Conditional Outlier Detection*.

First, we study the *Multi-label Classification* (MLC) problem that concerns classification of data instances into multiple binary output (class or response) variables that reflect different views, functions, or components describing the data. We present three MLC frameworks that effectively learn and predict the best output configuration for complex input-output data objects. Our experimental evaluation on a range of datasets shows that our solutions outperform several state-of-the-art MLC methods and produce more reliable posterior probability estimates.

Second, we investigate the *Conditional Outlier Detection* (COD) problem, where our goal is to identify unusual patterns observed in the multi-dimensional binary output space given their input context. We made two important contributions to the definition and solutions of COD. First, by observing a gap in between the development of unconditional and con-

ditional outlier detection approaches, we propose a ratio of outlier scores (ROS) that uses a pair of unconditional scores to calculate the conditional scores. Second, we show that by applying the chain decomposition of the probabilistic model, the probabilistic multivariate COD score decomposes to a set of probabilistic univariate COD scores. This decomposition can be subsequently generalized and extended to a broad spectrum of multivariate COD scores, including the new ROS score and its variants, leading to a new multivariate conditional outlier scoring framework. Through experiments on synthetic and real-world datasets with simulated outliers, we provide empirical results that support the validity of our COD methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

PREFACE

The work presented in this thesis would not have been possible without the help and support of many people. I take this opportunity to extend my sincere gratitude and appreciation to all those who made this thesis possible.

I would like to express my heartfelt gratitude to my advisor, Milos Hauskrecht, who has guided me throughout my Ph.D. journey. Milos introduced me to this exciting field of machine learning and data mining and trained me to his own high standards. Through the lectures, seminars, projects, and discussions, he has molded me into the independent researcher I am today. All along, Milos has been someone who always trusted and supported me in literally every situation.

I am very grateful to our former post-doc, Iyad Batal, who was an inspiring mentor, as well as a good friend of mine. Iyad got me interested in multi-label classification. Many parts of the thesis have been done in collaboration with him. I also would like to thank Sangyuen Cho, who first exposed me to the academic research experience when I was a visiting undergraduate student here at Pitt and offered me a lot of help when I joined back as a graduate student.

I would like to thank my thesis committee members, Rebecca Hwa, Adriana Kovashka, and Greg Cooper, for their valuable feedback and insightful discussions during my thesis defense. Greg was also a member of our research team investigating various clinical/medical projects. It was a privilege to work with him and also with Gilles Clermont and Shyam Visweswaran on these important projects. I am grateful for their positive influence on me that promoted and developed my scientific thinking and reasoning skills.

Besides Iyad, I was also fortunate to work with extraordinary lab mates, Quang Nguyen, Salim Malakouti, Zhipeng Patrick Luo, and Siqi Liu, on our awesome projects on clinical

outlier monitoring and alerting. I would like to thank all of them for the collaborations and our endless discussions. I also want to thank other lab alumni: Hamed Valizadegan, Lei Wu, Michal Valko, Dave Krebs, Saeed Amizadeh, Eric Heim, and Zitao Liu.

During my Ph.D. career, I spent two amazing summers at Bosch RTC and Siemens CT. I would like thank to my mentors and managers, Rumi Ghosh, Soundar Srinivasan, Dmitriy Fradkin, and Amit Chakraborty, who motivated and helped me to stretch my boundaries and grow in my abilities. I also want to extend my thanks to Zubin Abraham, Heng Wang, Mahmudur Rahman, Congrui Yi, Goktug Cinar, Ruobing Chen, Mohammad Shokoohi-Yekta, Ioannis Akrotirianakis, Ramamani Ramaraj, Sindhu Suresh, Chao Yuan, Bernardo Hermont, Xiaoyan Xiang, Tugba Kulahcioglu, Chabin Guillaume, Xi He, Jie Liu, Inbeom Song, Wonki Yoon and his family, Hyun-a Song and Youngjin Kim, Pastor Jungwook Kang and Korean Church of Love, Pastor Dongwook Kim and Princeton Korean Presbyterian Church, who made my stay just like a home away from home.

I am grateful to our tech staff, Bob Hoffman, Terry Wood, the late Russ Howard, Adam Hobaugh, and Walter Gibson, who were always on standby for trouble calls and made sure the Elements cluster is running 24/7. I would also like to thank our administrative staff, including Keena Walker, Karen Dicks, Michele Thomas, Deb Lauro, the late Kathleen Allport, Wendy Bergstein, and Nancy Kreuzer, whose support is vital to our department.

My life in Pittsburgh would have been merely monotone and even gloomy if not for these people: Jinyoung Jung and Seunghyun Yoon, Jieun Kim, Chilman Bae and Eunjoo Kim, Nohyun Park and his family, Eun-kyung Hong and Chanil Jung, Seungjae Baek and his family, Heekwon Park and his family, Kiyeon Lee, Ju-young Jung, Hyungbo Shim and Ji Young Song, Okrae Kim and his family, Daesup Lee and his family, Hyunjin Abraham Lee, Rakan Maddah, Phillip Walker, Shih-Yi James Chien, Donghun Don Lee, Judong Lee, Donghyun Ku and Gahgene Gweon, Pastor Hongkil Lee and his family, Pastor Youngsun Cho and Pittsburgh Korean Assembly of God, Pastor Jonathan Kim and State College Korean Church, Pastor Paul Becker and First Presbyterian Church of Bakerstown. I would like to mention and express my appreciation to the faculty members of Handong Global University and many friends I met there, especially the members of *SLE* – I know many of you still keep me in your prayers. I also want to mention and thank Jinhun Isaac Park, who generously

helped me proofread this thesis.

Lastly, my biggest love and appreciation go to my wonderful family. I am deeply thankful to my closest friend and wife, Hyun Joo, and our proud daughter, Cielle, who both have been absolute blessings to me. I thank my parents and parents-in-law, who have made innumerable sacrifices throughout their life to bring me and Hyun Joo up. I would like to send my love to my extended families in New Jersey and everywhere in Korea for their sincere love and prayers.

I praise God for His love, righteousness, and faithfulness (Psalm 1).

1.0 INTRODUCTION

With recent advances in data acquisition and storage technologies, vast amounts of data of various kinds and from various sources are being generated and collected every second. The increase in the amounts of collected data is often accompanied by the increase in the complexity of data types and objects we are able to store: univariate time series data are being replaced with multivariate time series, low-dimensional data objects are becoming high-dimensional, input-output data pairs for classification tasks include multiple (not just one) class labels, *etc.* All these prompt the development of new data analytic and machine learning solutions that are scalable to these new types of data and capable of overcoming the new complexity challenges.

This thesis focuses on the development of analytic methods for one such data type: *complex input-output data objects with high-dimensional multivariate binary output spaces.* The input-output data objects are typically used for classification and annotation purposes and a large number of data analytic and modeling algorithms have been developed over many years to solve them. However, the majority of them assume that data instances are linked to simple (univariate) class variable. Much less research and solutions are available when data objects are associated with multiple class variables. Examples of real-world problems when data objects come with multiple class variables are:

- **Document topic classification**: In text classification, a document can cover multiple predefined topics [Kazawa et al., 2005, Zhang and Zhou, 2006]. For example, a news article may belong to *politics* and *economics*. As in the image/video classification example, these topics can be represented by a set (vector) of mutually non-exclusive indicators.
- **Semantic image/video analysis**: In image (or video) analysis and classification, an

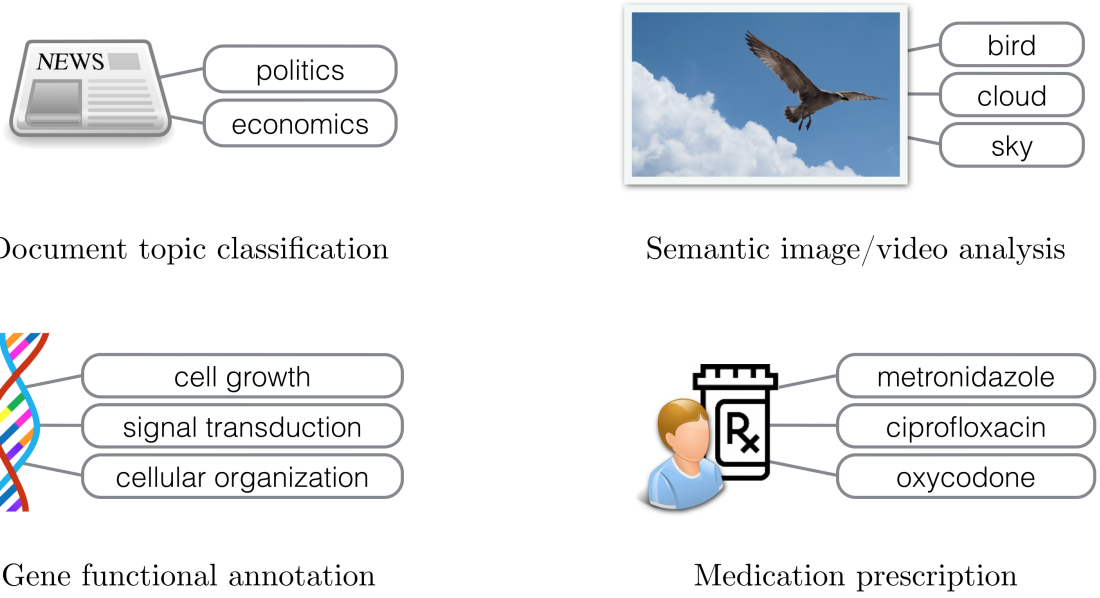


Figure 1.1: Data objects with multiple output variables.

image (or video) can be annotated with multiple tags [Boutell et al., 2004, Qi et al., 2007a]. For example, an image can be tagged with *bird*, *cloud*, and *sky*. Typically, such annotations are defined by an indicator vector, where each element represents a keyword.

- **Music emotion recognition**: In detecting emotions from music, each time-varying music feature sequence is labeled with combinations of different emotions such as *happy*, *sad*, *angry*, *calm*, and so on [Trohidis et al., 2011, Kim et al., 2010]. Given a predefined set of emotions, the labels can be coded as a binary vector where each element represents an emotion class.
- **Gene functional annotation**: In gene functional analysis, a single gene may be associated with several functionalities, which can be represented as a vector of functional class variables [Clare and King, 2001, Zhang and Zhou, 2006].
- **Medication prescriptions in electronic health records (EHR)**: In hospitals, a patient may receive multiple medications in a prescription. Such records of medication orders can be expressed in a vector where the elements denote whether individual medications are ordered or not [Hauskrecht et al., 2007, Hauskrecht et al., 2010, Hauskrecht

et al., 2013, Hauskrecht et al., 2016].

The main goal of the thesis is to develop computational methods that find data objects with abnormal (unusual) input and output associations in the above described data collections. There are two fundamental questions that arise in addressing this goal:

Question 1 - Representation or definition of normality: For given input-output data objects, how should one obtain the representation or definition of normal (usual) data?
Question 2 - Measures of abnormality: Given a representation/definition of normal data, how should one measure and identify the abnormality of individual data object?

To answer the questions, we hypothesize that we can adopt machine learning approaches to build statistical models representing the input-output patterns of the population and, in turn, that we can utilize the resulting representation to analyze data objects for abnormalities by assessing normalcy or deviation from expected patterns. Accordingly, our data analytic and algorithm development work in this thesis will focus on the following two problems that are defined on data with multivariate binary output:

1. **Multi-Label Classification** which is pertinent to the question of how to *learn and predict the best output (response)* from complex input-output data. We study existing solutions to the multi-label classification problem and investigate ways to acquire more accurate and efficient data representations.
2. **Conditional Outlier Detection** which is concerned with how to *identify unusual output patterns* in multivariate conditional data. To our knowledge, no precedent work has focused on this specific research problem. We conduct an exploratory study to formalize a definition of conditional outlier detection and propose effective solutions to the problem.

Below we briefly introduce the two problems and our solutions.

1.1 MULTI-LABEL CLASSIFICATION

In the traditional supervised learning scenarios, each data instance (represented by an input vector) is assumed to be associated with a single class label (output). Accordingly, the

process of learning from data to predict class labels can be described as seeking unidirectional dependence relations from input to output. When it comes to data with high-dimensional multivariate output, however, the same approach may not properly address the task because of the following properties of the data: (1) each output variable is not only dependent on input, but also dependent on other output variables – *e.g.*, in the semantic image analysis example above, knowing that an image is tagged with *bird* may increase the possibility of the image being tagged with *sky* – and (2) the number of all possible output combinations grows exponentially to the output dimensionality – *e.g.*, in the document classification example, if the number of all topics is d, the number of all possible topic combinations is 2^d. These properties apparently make the classification task larger and harder. More specifically, to learn a classifier, one has to capture the dependences in both input-output and output-output relations. To predict the best output, one has to evaluate exponentially many label configurations. Consequently, to effectively perform classification on data with multivariate output, more sophisticated yet efficient supervised learning methods are required.

The problem of classification on data with multiple binary class variables, which reflect different views, functions, or components describing the data, is often referred to in the literature as *Multi-label Classfication* (MLC) [Tsoumakas et al., 2010, Zhang and Zhou, 2013]. The goal of MLC is to learn a function from data that assigns to each data instance, represented by a continuous feature vector (input), a binary vector of class labels (output).

Early MLC methods [Clare and King, 2001, Boutell et al., 2004] assumed that all class variables are conditionally independent of each other, and learned individual classification functions to predict each output dimension separately. Obviously, this does not suffice to address the MLC problem, because the methods ignore all dependences among different class variables. Realizing the deficiency of the early solutions, a number of approaches have been developed and proposed to better model the relations among class variables. These include the methods using two levels of classifiers [Godbole and Sarawagi, 2004, Cheng and Hüllermeier, 2009], multi-label extension of the k-nearest neighbor algorithm [Zhang and Zhou, 2007], error-correcting output coding approach [Hsu et al., 2009, Zhang and Schneider, 2012], classifier chains methods [Read et al., 2009, Dembczynski et al., 2010], multi-dimensional Bayesian networks [van der Gaag and de Waal, 2006, Bielza et al., 2011],

etc. However, each of the solutions has its own limits in terms of either optimality or complexity and scalability.

In this thesis, we present three MLC frameworks that effectively learn and predict the best output from complex input-output data objects. In the first solution, we assume that dependences among class variables are restricted and follow a *directed tree* structure. The tree structure can represent limited dependence relations in the output space, where each output variable can be conditioned (dependent) on at most one other output variable (its parent in the tree). The benefit is that this restriction lets us define efficient learning and prediction methods. We develop a learning algorithm that efficiently discovers the optimal tree structure from a pairwise conditional dependence analysis, and a linear-time prediction algorithm that finds the best class labels for a given input. We implement the ideas using a special Bayesian network, whose conditional distributions are defined using a set of probabilistic classifier functions. We refer to the model as *Conditional Tree-structured Bayesian Networks* (CTBN; Section 3.2).

Though the tree-structure assumption facilitates efficient learning and prediction in the CTBN framework, yet it may restrict a full recovery of the underlying dependence relations especially when the true dependences do not follow a tree. To alleviate this, we propose and build a mixture ensemble framework for MLC that leverages the computational advantages of CTBN and the abilities of mixtures that compensate for the tree-structure restrictions. In particular, we extend the Mixtures-of-Trees [Meilă and Jordan, 2000] framework, which is originally a generative framework that models multi-dimensional discrete data, to incorporate multiple CTBNs as its base models. Consequently, our mixture can learn various dependence relations, which a single tree-structured model cannot capture, and combine them to make ensemble predictions that achieve a higher predictive accuracy. Our second MLC solution is referred to as *Mixtures-of-Conditional Tree-structured Bayesian Networks* (Section 3.3).

Lastly, we further refine the above mixture solution by allowing the base MLC models to have different structural assumptions other than a tree (*e.g.*, chain [Read et al., 2009]). Using the Mixtures-of-Experts [Jacobs et al., 1991] framework, our approach captures different input–output and output–output relations that tend to change across data. As a result, we can recover a rich set of dependence relations that a single MLC model cannot capture due to

its modeling simplifications. We refer to this last solution as *Multi-Label Mixtures-of-Experts* (Section 3.4).

Note that our MLC solutions presented in this thesis are based on the structured probabilistic graphical models [Koller and Friedman, 2009, Bakir et al., 2007, Nowozin et al., 2014], which provide the principles and techniques that support probabilistic model construction, learning, and inference for complex data. Accordingly, at the end of the learning process, our methods produce a well-defined model of posterior class probabilities that is extremely useful not only for prediction, but also for decision making [Raiffa, 1997, Berger, 1985] and for performing any inference over subsets of output class variables. In contrast to this, the majority of existing MLC methods aim to only identify the best output configuration for the given input.

The conditional outlier detection problem, which we will introduce next, is one of such problems that can be effectively solved by utilizing our MLC models and methods. While there are other types of problems that would benefit from our solutions, as we will describe shortly, conditional outlier detection in multi-dimensional output spaces is particularly less studied in the literature despite its potential importance. In the second half of the thesis, we conduct an investigation on how to effectively approach and solve this specific type of outlier detection problem.

1.2 CONDITIONAL OUTLIER DETECTION

Outlier detection [Markou and Singh, 2003, Kriegel et al., 2010, Aggarwal, 2017, Pimentel et al., 2014] is a data analysis method that aims to find atypical behaviors, unusual outcomes, or erroneous readings and annotations in data.[1] It has been an active research topic in data mining and machine learning communities and frequently used in various applications to identify rare and interesting data patterns that may be associated with either beneficial or malicious events such as fraud identification [Fawcett and Provost, 1997, Wang, 2010], network intrusion surveillance [Tan et al., 2002, Garcia-Teodoro et al., 2009], disease

[1] *Outliers* are also referred to as *anomalies*, *abnormalities*, *novelties*, *discordances*, or *deviants*.

outbreak detection [Wong et al., 2003], patient monitoring for preventable adverse events (PAE) [Hauskrecht et al., 2007, Hauskrecht et al., 2013], *etc.* It is also utilized as a primary data preprocessing step that helps to remove noisy or irrelevant signals in data [Hodge and Austin, 2004, Liu et al., 2004].

Despite huge effort and progress in outlier detection research, however, the majority of existing methods are designed only to detect *unconditional* outliers, that are unusual data instances manifested in the joint space of all data attributes. Such methods may not work when one wants to identify *conditional* (contextual) outliers that reflect unusual responses for a given set of contextual information. That is, since the outliers are conditioned on the context or properties of data instances, applying unconditional outlier detection methods to the conditional outlier detection problem may lead to incorrect results.

Compared with the unconditional outlier detection problem, the multivariate conditional setting is more useful in certain types of applications where the data are naturally associated with multiple descriptors, views, or decisions. Below are some examples of multivariate conditional outliers:

- **Errors in user-annotated image/video tags**: Most online media sharing services allow the users to tag their images (or videos) with simple, relevant keywords. The tags given by the users then serve as mnemonic indices, which make the images easily accessible. However, those user-annotated tags may mistakenly include keywords that are irrelevant to the associated image, which can be considered as multivariate conditional outliers [Boutell et al., 2004, Qi et al., 2007a].
- **Misassigned document classes**: Classification-based search engines index the web documents according to their topics, which are assigned by some labeling schemes or automated algorithms. Considering a document can have multiple topics (*e.g.*, a news article that simultaneously covers political and economic issues), unusual assignments of topics can be treated as multivariate conditional outliers [Kazawa et al., 2005, Zhang and Zhou, 2006].
- **Unusual gene or protein annotations**: In bioinformatics, various methods are used to identify biologically meaningful genomic sequences and annotate their function. Since each sequence can be associated with multiple functional classes or protein products,

incorrect annotation could be well considered as multivariate conditional outliers [Clare and King, 2001, Zhang and Zhou, 2006].

- **Monitoring for preventable adverse events (PAE)**: In hospitals, physicians and medical professionals make decisions on what tests or treatments to give to a patient. For those clinical decisions that are made based on the symptoms and conditions of patients, the multivariate conditional framework could be useful for discovering unusual decision patterns that potentially correspond to medical mistakes. [Hauskrecht et al., 2007, Hauskrecht et al., 2010, Hauskrecht et al., 2013, Hauskrecht et al., 2016]

In spite of the importance and impact that the problem has, only recent years have seen increased interest in conditional outliers and proposed methods that deal with them [Hauskrecht et al., 2007, Song et al., 2007, Valko et al., 2011a, Hauskrecht et al., 2016]. Consequently, the existing methods still exhibit limitations in availability and diversity.

In this thesis, we study the problem of *conditional outlier detection* (COD), where outliers are manifested in multivariate binary response (output) space and are conditioned on their context (input). Our goal is to identify irregular response patterns given a set of input-output data pairs. This special type of outlier detection problem is challenging because both the context, represented as input, and interdependences of responses should be taken into account when identifying outliers.

Our investigations of the COD problem focus on two main directions: (1) We develop a new COD framework in which multivariate conditional outlier scores decompose into a set of univariate conditional outlier scores representing full dependences among input and outputs. (2) We develop and investigate a new conditional outlier scoring approach that is built from unconditional outlier scores.

First, we start our investigation of the multivariate COD problem by considering a probabilistic view of outlier detection: *conditional outliers are data instances with a low conditional probability* [Hauskrecht et al., 2007]. Using this definition we formulate the multivariate COD problem with the help of multi-label classification (MLC) models that let us decompose the multivariate conditional probability into the product of univariate conditional probabilities. We show that this decomposition also transfers to outlier scores; that is, the multivariate conditional outlier score is decomposed into a set of univariate conditional scores (one score

per input dimension). After that, we extend the idea to support other types of outlier scores including non-probabilistic ones, yielding a decomposable multivariate conditional outlier score framework. Throughout the thesis, we study different instances of this framework that: (1) rely on relaxations of the exact probabilistic model, (2) permit different weighting of univariate conditional scores, and (3) accept a new class of conditional scores based on unconditional methodologies (see next).

Second, motivated by a gap in between two kinds of outlier detection problems, *conditional* and *unconditional*, we focus on the development of a new class of conditional outlier detection methods that rely on the solutions of unconditional methods. We propose to compute the conditional outlier score for a data instance by comparing (via ratio) two unconditional outlier scores: one in which the score is calculated against the instances with the same observed output value; and another in which the score is calculated for the instances with the opposite output value. We explore how this new outlier score applies to univariate conditional outlier detection, where data comes with high-dimensional input. After that, we investigate how to utilize the score to support multivariate COD.

1.3 OUR CONTRIBUTIONS

The main contributions of this thesis are:

- Modeling and Prediction of Multivariate Responses
  - We present a novel tree-structured probabilistic model that represents the posterior distribution of multivariate output.
  - We show how to build an ensemble model that incorporates multiple tree-structured Bayesian networks into a data model that represents the joint conditional probability of multivariate output.
  - We present a generalized representation of the posterior distribution that includes a number of previous relevant data models [Boutell et al., 2004, Clare and King, 2001, Batal et al., 2013, Read et al., 2009].

  - We extend the Mixtures-of-Experts [Jacobs et al., 1991] framework such that the framework represents the joint conditional distribution of multivariate output using our generalized posterior models as base classifiers.

- Conditional Outlier Detection
  - We extend the definition of conditional outliers [Hauskrecht et al., 2007] to the multivariate conditional outlier problem where outliers are manifested in the multivariate binary response (output) space, conditioned on their context (input).
  - We develop a new multivariate conditional outlier detection framework that relies on the decomposable models and extends the current state-of-the-art conditional outlier detection approaches [Hauskrecht et al., 2007, Hauskrecht et al., 2010, Hauskrecht et al., 2013, Hauskrecht et al., 2016] to multivariate settings.
  - We propose and develop a new ratio-based conditional outlier scoring approach that is derived by combining the results of any unconditional outlier scoring approach.
  - We enhance the new ratio-based conditional outlier scoring approach with discriminative dimensionality reduction methods to improve its performance for high-dimensional settings.

We would like to note that parts of this thesis work have been published as [Batal et al., 2013, Hong et al., 2014, Hong et al., 2015, Pakdaman et al., 2014, Hong and Hauskrecht, 2015, Hong and Hauskrecht, 2016].

1.4 ORGANIZATION OF THE THESIS

The rest of this thesis is organized as follows. Chapter 2 formally defines the problem that we are addressing in this thesis and reviews the existing solutions to the problem. Chapter 3 studies the multivariate data modeling problem and presents our solutions that are based on the structured probabilistic modeling approach and ensemble techniques. Chapter 4 investigates the conditional outlier detection problem and presents our solutions, including the probabilistic model-based approach and the ratio-based conditional outlier scoring approach. Finally, Chapter 5 concludes the thesis and outlines the future research directions.

2.0 BACKGROUND

This chapter provides the background of the problems investigated in the thesis. We start with the problem of *probabilistic modeling and prediction of multivariate responses* (Section 2.1). We describe the problem and discuss how it can be effectively addressed by learning the joint conditional probability from data. We briefly review the existing solutions in the literature. We then move on to the problem of *conditional outlier detection in multi-dimensional response space* (Section 2.2). We review the existing multivariate outlier detection approaches and motivate the multivariate conditional approach by pointing out the limitations of the previous solutions in identifying certain types of outliers. At the end of each section, we stress the differences of our solutions from the existing methods.

2.1 MODELING AND PREDICTION OF MULTIVARIATE RESPONSES

This section considers the problem of modeling and prediction in the multi-dimensional binary response space, which is referred in the literature as *Multi-Label Classification* (MLC) [Tsoumakas and Katakis, 2007, Tsoumakas et al., 2010, Zhang and Zhou, 2013]. In particular, we formulate our target problem as follows: We are given labeled training data $D = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$, where $\mathbf{x}^{(n)} = (x_1^{(n)}, ..., x_m^{(n)})$ is a m-dimensional *context* vector representing the n-th instance (*input*) and $\mathbf{y}^{(n)} = (y_1^{(n)}, ..., y_d^{(n)})$ is its corresponding d-dimensional binary *response* vector (*output*). As discussed in Chapter 1, this problem formulation applies to various real-world applications, such as document topic classification, semantic image/video analysis, and gene functional annotation (see Chapter 1 for detailed description). Our objective is to learn a function h from D such that h assigns to each instance, represented

by its context vector, a response vector (*i.e.*, $h : \mathbb{R}^m \rightarrow \{0,1\}^d$).

One approach to this task is to model and learn the *joint conditional distribution* $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{Y} = (Y_1, ..., Y_d)$ is a random variable for the response vector and $\mathbf{X}$ is a random variable for the context vector. Assuming the 0-1 loss function, the optimal classifier h^* assigns to each instance $\mathbf{x}$ the maximum a posteriori (MAP) assignment of the response variables:

$$\begin{aligned} h^*(\mathbf{x}) &= \arg\max_{\mathbf{y}} P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) \\ &= \arg\max_{y_1,...,y_d} P(Y_1 = y_1, ..., Y_d = y_d|\mathbf{X} = \mathbf{x}) \end{aligned} \tag{2.1}$$

A key challenge in modeling and learning $P(\mathbf{Y}|\mathbf{X})$ from data, as well as for defining the corresponding MAP classifier, is that the number of all possible response combinations to be considered is 2^d. Accordingly, our goal is to develop efficient models and methods for learning and inference that overcome this difficulty.

Remarks on Notation and Terminology:

- For notational convenience, we will omit the index superscript $^{(n)}$ when it is not necessary.
- We may also abbreviate the expressions by omitting variable names; *e.g.*, $P(Y_1 = y_1, ..., Y_d = y_d|\mathbf{X} = \mathbf{x}) = P(y_1, ..., y_d|\mathbf{x})$.
- We will interchangeably use terms *context*, *input*, and *feature*, which are denoted by variable $\mathbf{X}$. Similarly, we will use terms *response*, *output*, and *class* interchangeably, which are denoted by variable $\mathbf{Y}$.

2.1.1 Binary Relevance – Why Learning Independent Classification Models is Not Enough

A simple solution to the MLC problem is to learn a collection of independent classifiers – one for each class variable [Boutell et al., 2004, Clare and King, 2001, Schapire and Singer, 2000] – which is known as the *Binary Relevance* (BR) approach. That is, BR learns a separate classifier h_i for each class variable $Y_i : i \in \{1, ..., d\}$ and determines the output of a new instance $\mathbf{x}$ by simply aggregating the predictions of all classifiers:

$$h_{\text{BR}}(\mathbf{x}) = (h_1(\mathbf{x}), ..., h_d(\mathbf{x})) \tag{2.2}$$

$$= \left(\arg\max_{y_1} P(y_1|\mathbf{x}), ..., \arg\max_{y_d} P(y_d|\mathbf{x}) \right) \tag{2.3}$$

From a probabilistic point of view, this approach can be justified by conditional independence that all class variables are conditionally independent of each other given $\mathbf{x}$:

$$P_{\mathrm{BR}}(y_1, ..., y_d|\mathbf{x}) = \prod_{i=1}^{d} P(y_i|\mathbf{x}) \tag{2.4}$$

However, this simple approach does not always produce correct results, as shown in the following example [Batal et al., 2013].

Example 1. *Assume the joint conditional distribution of class variables Y_1 and Y_2 for a specific instance $\mathbf{x}$ is as shown in Table 2.1. The optimal classification for $\mathbf{x}$ (according to Equation (2.1)) is $h^*(\mathbf{x}) = (Y_1 = 1, Y_2 = 0)$. However, the result of BR (according to Equation (2.3)) is $h_{BR}(\mathbf{x}) = (Y_1 = 0, Y_2 = 0)$.*

$P(Y_1, Y_2|\mathbf{X} = \mathbf{x})$	$Y_1 = 0$	$Y_1 = 1$	$P(Y_2|\mathbf{X} = \mathbf{x})$
$Y_2 = 0$	0.25	**0.40**	**0.65**
$Y_2 = 1$	0.30	0.05	0.35
$P(Y_1|\mathbf{X} = \mathbf{x})$	**0.55**	0.45	

Table 2.1: The joint distribution of class variables Y_1 and Y_2 conditioned on instance $\mathbf{x}$. The optimal (MAP) prediction is $h^*(\mathbf{x}) = (Y_1 = 1, Y_2 = 0)$.

2.1.2 Early Multi-label Classification Approaches

Realizing the deficiency of BR [Boutell et al., 2004, Clare and King, 2001, Schapire and Singer, 2000] in addressing the MLC problem, several research directions have been proposed to model the relations between the class variables. [Godbole and Sarawagi, 2004] proposed a method that builds two levels of classifiers: The first level classifiers learns to predict values of each class variable using the original features (*i.e.*, the first level is equivalent to BR). The second level learns to predict values of each class variable using the original features and the output of the first level. [Zhang and Zhou, 2007] presented the Multi-Label k-Nearest Neighbor (ML-KNN) method, which learns a classifier for each class variable by

binding k-nearest neighbor with Bayesian inference. A combination of ML-KNN and logistic regression was presented in [Cheng and Hüllermeier, 2009], where the class proportions of nearest neighbors are used as additional features for the logistic regression classifiers. The limitation of these early approaches is that class dependences are either not modeled at all or modeled in a very limited way.

2.1.3 Output Coding Approaches

An alternative approach to MLC is based on the error-correcting output coding (or simply output coding) approach [Dietterich and Bakiri, 1995]. The idea is to encode the output values into a codeword, learn how to predict the codeword, and then recover the correct output from noisy predictions. A variety of output coding methods have been proposed by utilizing different encoding strategies, such as compressed sensing [Hsu et al., 2009], principal component analysis [Tai and Lin, 2010], and canonical correlation analysis [Zhang and Schneider, 2011]. The state-of-the-art in this approach utilizes a maximum margin formulation that promotes both discriminative and predictable codes [Zhang and Schneider, 2012]. The limitation of output coding methods is that they can only predict the single "best" output for a given input, and they cannot compute probabilities for different input-output pairs.

2.1.4 Classifier Chains and Its Extensions

[Read et al., 2009] introduced the Classifier Chains (CC) method for MLC. The idea is to link different binary classifiers in a chain, such that each classifier incorporates the (0/1) predictions of all preceeding classifiers in the chain as additional features. For example, assume that the order of the class variables in the chain is $Y_1 < Y_2, ... < Y_d$. To classify a new instance $\mathbf{x}$, classifier h_1 first predicts $Y_1 = \widehat{y}_1 \in \{0, 1\}$ from $\mathbf{x}$. After that, h_2 predicts $Y_2 = \widehat{y}_2$ using $\mathbf{x}$ and the predicted value $\widehat{y}_1$. By repeating this to Y_d along the chain, h_i predicts $Y_i = \widehat{y}_i$ using $\widehat{y}_1, ..., \widehat{y}_{i-1}$ as additional input features.

The CC method has been extended in several ways. [Zhang and Zhang, 2010] realized that the performance of CC is influenced by the order of class variables in the chain (the

original proposal [Read et al., 2009] orders arbitrarily) and proposed a method that learns such ordering from data. [Zaragoza et al., 2011] explored the unconditioned dependence relations in the output space and constructed chains using the mutual information between the class variables.

The main disadvantage of CC and its extensions [Read et al., 2009, Zhang and Zhang, 2010, Zaragoza et al., 2011] is that they do not perform proper probabilistic inference for classification (*i.e.*, they do not correctly solve Equation (2.1)). Instead, they simply propagate the predictions through the class variables according to the order defined by the chain, which is a greedy mode-seeking heuristic [Dembczynski et al., 2010]. However, such a heuristic may produce incorrect results as we show in the following examples.

Example 2. *Consider the conditional distribution in Table 2.1 and assume the order of the class variables in the chain is* $Y_1 < Y_2$*. CC starts incorrectly by predicting* $Y_1 = 0$ *and eventually produces the suboptimal prediction* $(Y_1 = 0, Y_2 = 1)$.

[Dembczynski et al., 2010] discussed the suboptimality of CC, which is depicted in the above example, and presented Probabilistic Classifier Chains (PCC) that estimates the entire posterior distribution of the class labels. However, this method has to evaluate exponentially many label configurations, which greatly limits its applicability.

2.1.5 Multi-Label Conditional Random Fields

Another approach for modeling $P(\mathbf{Y}|\mathbf{X})$ relies on conditional random fields (CRFs) [Lafferty et al., 2001]. [Ghamrawi and McCallum, 2005] presented a method called Collective Multi-Label with Features classifier (CMLF) that captures label co-occurrences conditioned on features. However, CMLF assumes a fully connected CRF structure which requires a high computational cost. Later, [Shahaf and Guestrin, 2009] and [Bradley and Guestrin, 2010] proposed to learn tractable (low-treewidth) structures of class variables for CRFs using conditional mutual information. More recently, [Pakdaman et al., 2014] used pairwise CRFs to model the class dependences and presented L_2-optimization-based structure and parameter learning algorithms. Although the later methods share similarities with our ap-

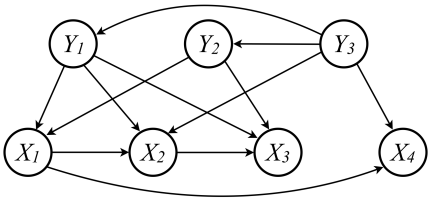


Figure 2.1: An example MBC [van der Gaag and de Waal, 2006, Bielza et al., 2011] which defines the joint probability distribution over three class variables $\{Y_1, Y_2, Y_3\}$ and four feature variables $\{X_1, X_2, X_3, X_4\}$.

proach by modeling the conditional dependences in $\mathbf{Y}$ space using restricted structures, their optimization of the likelihood of data is computationally more demanding. To alleviate this, CRF-based methods often resort to optimization of a surrogate objective function (*e.g.*, the pseudo-likelihood of data [Pakdaman et al., 2014]) or include specific assumptions (*e.g.*, features are assumed to be discrete [Ghamrawi and McCallum, 2005]; relevant features for each class are assumed to be known [Shahaf and Guestrin, 2009, Bradley and Guestrin, 2010]), which complicate the application of the methods.

2.1.6 Multi-Dimensional Bayesian Network Classifiers

Multi-dimensional Bayesian network Classifiers (MBC) [van der Gaag and de Waal, 2006, Bielza et al., 2011, Antonucci et al., 2013] build a generative model $P(\mathbf{X}, \mathbf{Y})$ using special Bayesian network structures that assume all class variables are ancestors of all feature variables (see Figure 2.1). To facilitate the model, MBC parameterize three sets of arcs between the input and output variables: namely, $\mathcal{A}_\mathcal{Y}$, $\mathcal{A}_\mathcal{X}$, and $\mathcal{A}_{\mathcal{X}\mathcal{Y}}$ such that $\mathcal{A}_\mathcal{Y} \subseteq \mathcal{V}_\mathcal{Y} \times \mathcal{V}_\mathcal{Y}$ are the arcs between the output variables, $\mathcal{A}_\mathcal{X} \subseteq \mathcal{V}_\mathcal{X} \times \mathcal{V}_\mathcal{X}$ are the arcs between the input variables and $\mathcal{A}_{\mathcal{X}\mathcal{Y}} \subseteq \mathcal{V}_\mathcal{Y} \times \mathcal{V}_\mathcal{X}$ are the arcs from the output variables to the input variables, where $\mathcal{V}_\mathcal{X} = \{X_1, ..., X_m\}$ and $\mathcal{V}_\mathcal{Y} = \{Y_1, ..., Y_d\}$ respectively denote the sets of input and output variables. Figure 2.1 shows an example MBC that is defined over three class variables and four feature variables.

Although our approach shares a few similarities with MBC, there are significant differ-

ences:

- MBC only handles discrete features and, thus, all features should be a priori discretized; while our approach handles both continuous and discrete features.
- MBC defines a joint distribution over both feature and class variables and the search space of the model increases with the input dimensionality m; while our search space does not depend on m.
- Feature selection in MBC is done explicitly by learning the individual relationships between features and class variables; while we perform feature selection by regularizing the base classifiers.
- MBC requires expensive marginalization to obtain class conditional distribution $P(\mathbf{Y}|\mathbf{X})$; while we directly model and estimate $P(\mathbf{Y}|\mathbf{X})$.

2.1.7 Ensemble Approaches

Several researchers proposed to use the ensemble approach for MLC in order to overcome the limitations and disadvantages that individual models have and achieve more precise and robust performance. [Read et al., 2009] presented Ensemble of Classifier Chains (ECC) that simply averages the predictions of multiple randomly structured CC models that are trained on bootstrapped subsets of data. [Zaragoza et al., 2011] followed the same ensemble approach and proposed Ensemble of Bayesian Classifier Chains (EBCC) that combines several chain-structured MBCs, obtained by changing the root node in the chain. [Antonucci et al., 2013] proposed an ensemble of multi-dimensional Bayesian networks combined via simple averaging. Each MBC in the ensemble represents different $\mathbf{Y}$ relation (the structures are set a priori and not learned) and all of the networks adopt the naïve Bayes assumption (*i.e.*, features are independent given class labels).

Although these methods significantly improve the predictive accuracy of the base models, they are limited in that the way they diversify the base classifiers heavily relies on randomization. Also their ensemble predictions are based on simple (uniform) averaging. Unlike these methods, our ensemble approaches learn the base models (both the structures and

parameters of base classifiers) and the mixing coefficients of the ensemble from data in a principled way.

2.1.8 Our Work

In Chapter 3, we develop and study novel probabilistic approaches that model and predict multi-label data with multivariate responses.

- First, we present a new model that represents the posterior distribution of multivariate responses (class labels) $P(\mathbf{Y}|\mathbf{X})$ using tree-structured Bayesian networks [Batal et al., 2013]. By restricting the conditional dependence relations between class variables to follow a *directed tree*, we devise efficient structure and parameter learning algorithms and a linear time ($O(d)$) exact MAP inference algorithm.

- Second, we build an ensemble method that incorporates multiple tree-structured Bayesian networks [Batal et al., 2013] into a data model that represents the joint conditional probability $P(\mathbf{Y}|\mathbf{X})$ [Hong et al., 2014]. Our approach is based on the Mixtures-of-Trees [Meilă and Jordan, 2000] framework that originally defines a generative model of $P(\mathbf{Y})$ for discrete multi-dimensional domains. We extend the Mixtures-of-Trees framework and present efficient supporting algorithms that learn the structures and parameters of the mixture model and perform a fast MAP inference for MLC.

- Last, we improve our ensemble method [Hong et al., 2014] by developing a generalized mixture framework for MLC [Hong et al., 2015]. We first propose a generalized representation of the class posterior distribution $P(\mathbf{Y}|\mathbf{X})$ that includes a number of previous MLC models [Boutell et al., 2004, Clare and King, 2001, Batal et al., 2013, Read et al., 2009] as special cases. We then extend the Mixtures-of-Experts [Jacobs et al., 1991] framework, which was originally built for the conditional distribution $P(Y|\mathbf{X})$ such that, by using our generalized class posterior models as base classifiers, the framework represents the joint conditional distribution. Our mixture representation recovers a rich set of dependence relations among inputs and outputs that a single MLC model cannot capture due to its modeling simplifications.

2.2 CONDITIONAL OUTLIER DETECTION

This section considers the conditional outlier detection problem in (possibly high-dimensional) binary response space – which we refer to as the *conditional outlier detection* (COD) problem. *Conditional outlier detection* is a special type of the outlier detection problem where data consists of m-dimensional continuous input vectors (context attributes) and corresponding d-dimensional binary output vectors (response attributes). Our goal is to precisely identify the instances with unusual input-output associations. Following the definition of an outlier given by Hawkins [Hawkins, 1980],[1] we define multivariate conditional outlier in plain language as follows:

Definition 1. *A multivariate conditional outlier is an observation, which consists of context and associated responses, whose responses are deviating so much from the others in similar contexts as to arouse suspicions that it was generated by a different response mechanism.*

As we illustrated in Chapter 1, this definition of conditional outlier fits well with various practical outlier detection problems that require contextual understanding of data.

However, the majority of existing methods are designed only to detect *unconditional* outliers that correspond to unusual data patterns expressed in the joint space of all data attributes. Apparently, these methods do not consider the dependences among the attributes and are not able to properly detect conditional outliers. Although there are several *conditional* outlier detection approaches that attempt to recover and reflect the input-output relations for outlier detection, existing solutions are rather limited and not capable of handling the particular problem that we are interested in. Below we briefly review existing outlier detection research, discuss their limitations in solving the multivariate conditional outlier detection problem in detail, and differentiate our multivariate conditional approach to them.

[1] While the concept of outlier is rather ill-defined and, indeed, there is no clear consensus on what an outlier is, probably the most referenced definition has been given by [Hawkins, 1980]: *"An outlier is an observation deviating so much from the others as to arouse suspicions that it was generated by a different mechanism."*

2.2.1 Unconditional Outlier Detection Approaches

One of the most important components in outlier detection research is the assumption regarding how outliers occur in a dataset. Below we categorize existing unconditional outlier detection approaches into *six* general groups (according to the assumption that the approaches have: *distance-based*, *density-based*, *depth-based*, *deviation-based*, *classification-based*, and *high-dimensional* approaches), and summarize their main ideas.

2.2.1.1 Distance-based Approaches Distance-based approaches are one of the commonly used unconditional outlier detection approaches. The methods that fall in this category assume that normal data instances are located in or near the main body of data distribution, while outliers are found far away from most data instances. Several parametric and nonparametric methods are proposed based on this assumption. Typical parametric examples are [Rousseeuw and Hubert, 2011, Rousseeuw and Zomeren, 1990, Rousseeuw and Leroy, 1987] that assign each data instance an outlier score using a robust distance metric ([Hubert and Debruyne, 2010, Rousseeuw and Driessen, 1999, Rousseeuw, 1984]) between each instance to the distribution center (*e.g.*, mean or median).

The nonparametric methods in this category have been proposed to evaluate outlier scores by analyzing the distance to the local neighbors of each data instance. [Knorr and Ng, 1997] computes the outlier score by counting the number of neighboring instances within a hypersphere of radius d. An instance is considered as an outlier if more than a fraction α of its k nearest neighbors are further than d from it. Similarly, [Byers and Raftery, 1998] and [Guttormsson et al., 1999] evaluate the outlier score of an instance using the distance to its k-th nearest neighbor in the dataset. [Eskin et al., 2002, Angiulli and Pizzuti, 2002] extend the preceding methods to replace the outlier score with the sum of the distances to the k nearest neighbors.

The distance-based approaches have been very popular in many applications as they are easy and flexible. In particular, the nonparametric methods in this category are flexible with respect to different data types in that they do not make assumptions about the underlying data distribution and can adapt by replacing the distance metric. However, the approaches

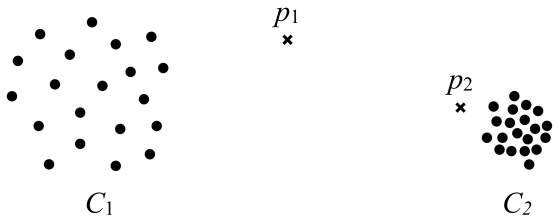


Figure 2.2: Example where the use of local density is desired.

are often computationally very demanding as they require to compute the distance between every instance pairs. Moreover, coming up with a proper distance metric is difficult when the data type is mixed or complex, such as graphs and sequences. Lastly, the approaches suffer from the "curse of dimensionality" issue [Weber et al., 1998, Hinneburg et al., 2000, Aggarwal et al., 2001]; *i.e.*, as the dimensionality of data increases, the distance metrics and density estimators become analytically ineffective and computationally intractable. These make the methods less suitable for high-dimensional data.

2.2.1.2 Density-based Approaches Another category of widely used outlier detection approaches is the density-based approaches. This category of methods assumes that the density around a normal data instance is similar to the density around its neighbors while that of an outlier is relatively lower than its neighbors. This assumption is particularly useful in many real-world application where the data has regions of varying densities. For example, in the dataset shown in Figure 2.2, the clusters C_1 and C_2 have different densities whereas the instances p_1 and p_2 are outliers that we want to identify. With the distance-based approaches, only p_1 can be identified as an outlier because, for any instance in C_1, the distance between the instance and its nearest neighbor is greater than the distance between p_2 and C_2. In other words, p_2 would be considered as an outlier, only after all instances in C_2 are considered as outliers.

A number of nonparametric methods have been proposed to tackle the above illustrated issue by estimating local density. Local Outlier Factor (LOF) [Breunig et al., 2000] is one of the most popular methods in this regard. LOF evaluates the outlier score of a data instance

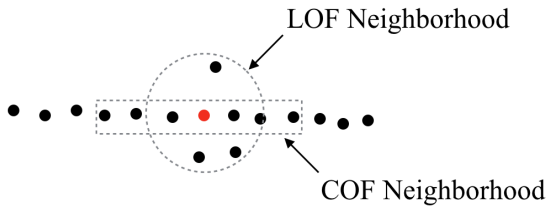


Figure 2.3: Difference between the neighborhoods used by LOF and COF (when $k = 6$).

by computing the ratio between the local density of the instance and the average local density of k neighboring instances:

$$LOF\,(\mathbf{x}, k) = \frac{\sum_{\mathbf{x}' \in N_k(\mathbf{x})} \frac{lrd_k(\mathbf{x}')}{lrd_k(\mathbf{x})}}{|N_k(\mathbf{x})|}$$

where $N_k(\mathbf{x})$ denotes the k-nearest neighborhood of instance $\mathbf{x}$ and

$$lrd_k(\mathbf{x}') = \frac{|N_k(\mathbf{x}')|}{\sum_{\mathbf{x}'' \in N_k(\mathbf{x}')} \max(k\text{-}dist(\mathbf{x}''), dist(\mathbf{x}', \mathbf{x}''))}$$

is the local reachability density, which in essence measures the geometric dispersion of the k-nearest neighborhood. The score given by LOF can be understood as the inverse of the relative density within the local neighborhood. Instances that have LOF score greater than 1 are generally considered as outliers.

LOF has influenced several subsequent works in the literature. For instance, Connectivity-based Outlier Factor (COF) extends LOF [Tang et al., 2002] to detect outliers from data on a manifold. The key difference of COF in contrast to LOF is how the method defines the local neighborhood. Specifically, to find the k nearest neighbors of an instance $\mathbf{x}$, COF incrementally grows a neighbor set denoted as $N'(\mathbf{x})$: First off, COF adds the nearest neighbor of $\mathbf{x}$ to $N'(\mathbf{x})$. Then, COF repeatedly finds and adds other neighboring instances to $N'(\mathbf{x})$ until $|N'(\mathbf{x})| = k$, such that the newly added instance has the smallest distance to any of the previously added instances in $N'(\mathbf{x})$. Unlike LOF, COF can capture localities over line components as compared in Figure 2.3.

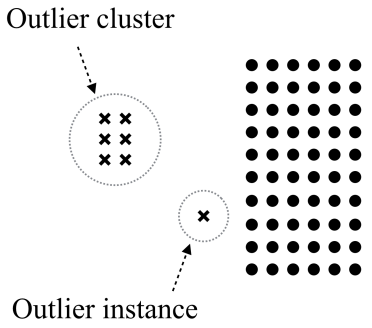
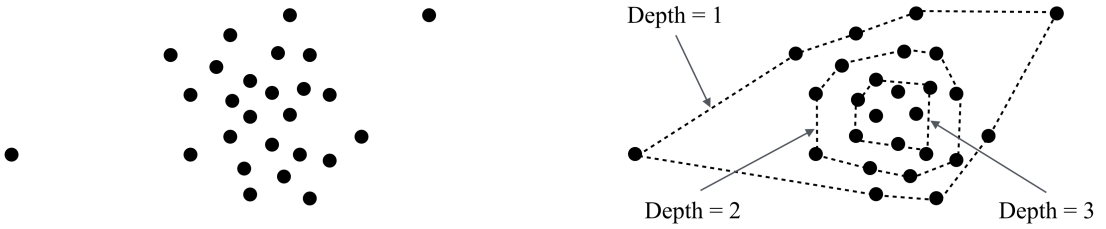


Figure 2.4: An example multi-granularity problem.

[Papadimitriou et al., 2003] propose another variants of LOF, called Local Correlation Integral (LOCI), to address the multi-granularity problem in outlier detection. The multi-granularity problem refers to a case where data is polluted not only from outlier instances but also from outlying-clusters (groups of outliers; see Figure 2.4). LOF is not able to handle such a problem unless a proper value of the neighborhood size k is provided. LOCI addresses it by introducing Multi-Granularity Deviation Factor (MDEF) that, for each test instance, computes the standard deviation of the local densities of the nearest neighbors. The outlier score of the instance is assigned by taking the inverse of this standard deviation.

The unique advantage of the density-based approaches is in that the solutions can be locally sensitive, which in turn let the approaches achieve a better detection accuracy in many real-world applications. The approaches are also very flexible as they do not make assumptions about the underlying data distribution. However, similar to the distance-based approaches, the density-based approaches are not easily scalable to larger datasets, because the solutions require a pairwise distance matrix to find neighborhoods. In addition, sometimes a proper distance metric cannot be easily determined, which may limit the applicability of the approaches. Lastly, as with the distance-based approaches, the approaches also suffer from the "curse of dimensionality" issue [Weber et al., 1998, Hinneburg et al., 2000, Aggarwal et al., 2001].



(a) Dataset

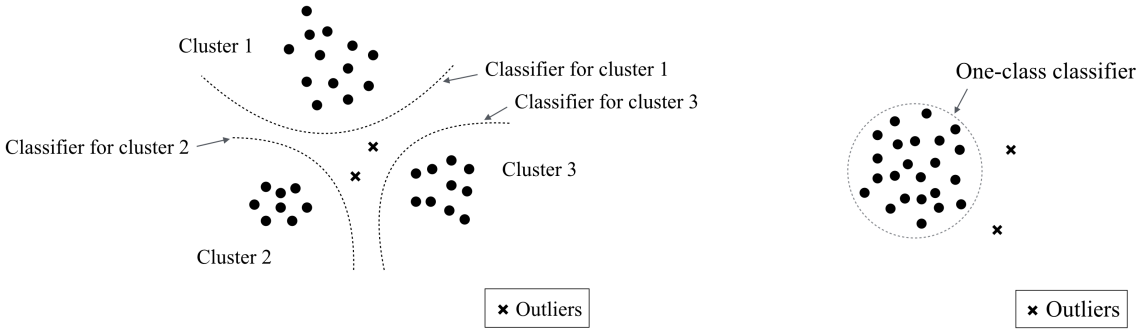
(b) Depth assigned by convex hull analysis

Figure 2.5: Depth-based outlier detection.

2.2.1.3 Depth-based Approaches Depth-based approaches assume that normal data instances are close to or in the center of data clusters, whereas outliers are at the fringes. Several nonparametric methods fall in this category that assign each data instance a depth k by gradually removing data using iterative convex hull analysis (Figure 2.5). At each iteration, all points that lie on the convex hull of data instances are removed; a depth of k is assigned to the removed instances. The instances with a low depth are considered as "fringe" instances and are possible candidates for outliers [Ruts and Rousseeuw, 1996, Johnson et al., 1998].

The approaches are flexible in that no assumption regarding the underlying data distribution is required. However, an application of the approaches could be very limited due to the computational cost of the convex hull analysis (usually only efficient with low dimensional datasets). Also, a convex hull in d-dimensional space contains at least 2^d points, which induces a large portion of data to be considered as outliers. This makes the approaches in high-dimensional spaces extremely ineffective.

2.2.1.4 Deviation-based Approaches Deviation-based approaches assume outliers are the outermost instances in a dataset, such that a removal of an outlier lowers the variance of the set to a large extent. One of the well-known nonparametric methods in this category is Linear Method for Deviation Detection (LMDD) [Arning et al., 1996]. Given a dataset, the method computes how much the variance is reduced (called smoothing factor) by removing



(a) Multi-class strategy

(b) One-class strategy

Figure 2.6: Classification-based outlier detection.

an instance. The instances whose exclusion minimizes the variance are treated as outliers.

As with other nonparametric methods, the deviation-based approaches do not require an assumption with respect to the underlying data distribution. However, the approaches require $O(2^n)$ times of variance estimation, which limits the application of the approaches. That is, for a dataset of size n, there are 2^n options of which instances to remove, which makes the approaches less scalable to large datasets.

2.2.1.5 Classification-based Approaches Classification-based approaches are based on a parametric assumption that a function of feature classifying normal and outlier instances can be learned from data. There are two major strategies to achieve classification-based outlier detection: *multi-class* and *one-class* classification strategies.

Multi-class classification strategy further assumes that normal data instances form a set of clusters (that the cluster information is either provided as class labels, or discovered by an additional clustering step). It then learns a classifier for each cluster in the one-vs-all manner.[2] In the testing time (when detecting outliers using the learned classifier), instances that do not belong to any of the clusters are considered to be outliers. Many early methods are developed based on this strategy. [De Stefano et al., 2000, Odin and Addison,

[2]For each data cluster, a classifier is trained by treating the cluster as the positive class and the other clusters as the negative class.

2000] apply multi-class neural network classifiers to learn normal data patterns for outlier detection. [Barbara et al., 2001, Valdes and Skinner, 2000] learn a simple Bayesian classifier from data that assumes the input variables are independent from each other (*i.e.*, a naïve Bayes classifier). Bayesian networks that model the hierarchical (conditional) structures of data have been applied in [Das and Schneider, 2007, Janakiram et al., 2006]. [Lee et al., 1997, Fan et al., 2001] derive a set of classification rules from data using rule mining algorithms (Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [Cohen, 1995] or decision trees [Quinlan, 1986]). The confidence associated with the rules are used to produce outlier scores.

On the other hand, one-class classification strategy assumes that all training instances are normal, and attempts to learn a discriminative boundary around the training (normal) instances. In the testing time, instances that fall out of the obtained decision boundary are considered to be outliers. support vector machines (SVMs) have been applied to outlier detection using this strategy. [Schölkopf et al., 1999] trains an SVM classifier by learning a decision boundary between training data and the origin (zero). Any test instances that lie across the boundary are classified as outliers. [Tax and Duin, 2004] learns the smallest hypersphere containing all training instances in the kernel space. This hypersphere is then used for classification such that instances that fall outside of it are considered as outliers. Similarly, [Roth, 2005, Roth, 2006] apply the one-class strategy to the Fisher discriminant analysis, and propose one-class kernel Fisher discriminants models.

The classification-based approaches turn many of the powerful classification algorithms to outlier detection methods. Accordingly, depending on the underlying data type and properties, one can further utilize various kernel techniques and optimization methods that are originally developed for the base classifiers. However, many classification-based approaches require an "outlier-free" training dataset to obtain a base classifier. This restricts their application to many real-world problems where acquiring such a dataset is difficult [Amer et al., 2013]. Also, some approaches (*e.g.*, methods based on SVMs) are only able to output binary labels indicating outliers, which are less informative than outlier scores.

2.2.1.6 Approaches for High-dimensional Data In high-dimensional spaces, the above described approaches often fail because the distance metrics and density estimators become analytically ineffective and computationally intractable [Weber et al., 1998, Hinneburg et al., 2000, Aggarwal et al., 2001]. In addition, as the volume of the spaces grows rapidly, all data objects appear to be sparse, and thus defining a meaningful neighborhood becomes difficult. Researchers have proposed outlier detection methods for high-dimensional data to handle such extreme cases. Typical methods in this category either adopt an invariant distance measurement, such as the angle-based outlier factor [Kriegel et al., 2008, Pham and Pagh, 2012], or apply dimensionality reduction techniques to project data to lower dimensional subspaces, such as grid-based subspace outlier detection [Aggarwal and Yu, 2001], reconstruction-based outlier detection [Hawkins et al., 2002, Williams et al., 2002], and outlier detection with subspaces [Lazarevic and Kumar, 2005, Keller et al., 2012].

Specifically, just as with depth-based approaches, the angle-based methods assume that outliers are located at the boundaries of data regions. However, the methods further exploit an observation that, at the boundaries, it is likely to enclose the entire data within a smaller angle [Kriegel et al., 2008]. Accordingly, the methods assume that the data instances with a smaller angle spectrum are outliers, while those with a larger spectrum are normal. On the other hand, the grid-based method [Aggarwal and Yu, 2001] partitions the original data space into a multi-dimensional equi-depth grid by segmenting each dimension with the same number of cells. Assuming statistical independence, the authors then sought k-dimensional grid cells (projections) that contain a significantly low number of instances than expected. [Hawkins et al., 2002, Williams et al., 2002] first find a lower dimensional representation of the original data using dimensionality reduction techniques (*e.g.*, replicator neural networks (auto encoders)). Then the methods attempt to reconstruct the original data from the lower dimensional representation. The outlier scores are assigned by measuring the reconstruction error on each instance.

[Lazarevic and Kumar, 2005, Keller et al., 2012] propose to adopt subspace analysis methods and utilize lower-dimensional representations of data for outlier detection. The methods detect outliers in two-fold. First, multiple sets of outlier scores are computed on different subspaces, which are either selected randomly [Lazarevic and Kumar, 2005] or found

by minimizing (the absolute value of) the correlation coefficient between attributes [Keller et al., 2012]. Second, the final outlier scores are obtained by combining the multiple scores in the first fold via simple heuristics (*e.g.*, average or maximum over multiple sets of scores). These methods can be considered as meta-approaches in that they allow a choice of outlier score. In Chapter 4, we present a similar meta-analysis technique for outlier detection.

So far we have reviewed existing unconditional outlier detection research by putting them into six categories. Next section continues our discussion on existing outlier detection solutions focusing on conditional outliers.

2.2.2 Conditional Outlier Detection

While the vast majority of existing methods attempt to solve the unconditional outlier detection problem, recent years have seen increased interest in conditional outlier detection (COD) that aims to identify outliers in responses given the observation of context variables. [Hauskrecht et al., 2007] formally introduced the concept of conditional outlier detection where the data instances consist of a set of input (context) and associated output (responses). The authors proposed a probabilistic model-based framework for COD in which a parametric model is used to describe the stochastic relations between the input and output variables. The framework can be integrated with any data models that are capable of producing class conditional probability $P(y|\mathbf{x})$. Outliers are assumed to have a low conditional probability given a trained model. In [Hauskrecht et al., 2007], the authors used a localized Bayesian belief network (BBN) or localized naïve Bayes model to represent data with discrete input and output variables. [Valko and Hauskrecht, 2008, Valko et al., 2008] investigated the instance-specific methods to acquire more accurate predictive models for COD. The authors presented a new metric learning algorithm to select instances that are similar to the target instance. In [Valko et al., 2011b], the authors employed local methods based on the graph laplacian [Chung, 1997] and value propagation methods [Zhu et al., 2003, Zhou et al., 2004] to estimate $P(y|\mathbf{x})$ for the target data instance. [Hauskrecht et al., 2010, Hauskrecht et al., 2013, Hauskrecht et al., 2016] further developed the framework to address COD with mixture of continuous and discrete input variables. To represent the dependence relations in data,

the authors adopted support vector machines (SVM) with a post-hoc calibration method to produce probability [Platt, 1999, DeGroot and Fienberg, 1983].

2.2.2.1 Multivariate Conditional Outlier Detection

While the above COD approaches only tackled the problem with 1-dimensional output (*i.e.*, univariate COD), in this thesis, we are also interested in the multivariate COD problem that aims to identify unusual sets of binary responses (output) given the observation of context (input).

Although it does not exactly match the problem that we are tackling, [Song et al., 2007] presented a relevant model-based approach for continuous input and output variables. The authors proposed to use two mixtures-of-Gaussian models to respectively represent input and output, and define a mapping function between them. The mapping function indicates the probability of a component in the input mixture being associated with that of the output mixture. The framework defines a generative process as:

$$f(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{|\mathbf{U}|} \frac{f_G(\mathbf{x}|\mathbf{U}_i)P(\mathbf{U}_i)}{\sum_{k=1}^{|\mathbf{U}|} f_G(\mathbf{x}|\mathbf{U}_k)P(\mathbf{U}_k)} \sum_{j=1}^{|\mathbf{V}|} f_G(\mathbf{y}|\mathbf{V}_j)P(\mathbf{V}_j|\mathbf{U}_i) \tag{2.5}$$

where

- $\mathbf{x}$ and $\mathbf{y}$ are input and output. $|\mathbf{U}|$ and $|\mathbf{V}|$ denote the number of mixture components for input and output, respectively.
- $f_G(\mathbf{x}|\mathbf{U}_i)$ and $f_G(\mathbf{y}|\mathbf{V}_i)$ are Gaussian density functions. For example, $f_G(\mathbf{x}|\mathbf{U}_i)$ is the likelihood that the i-th Gaussian component in $\mathbf{U}$ would produce $\mathbf{x}$.
- $P(\mathbf{V}_j|\mathbf{U}_i)$ is the probability that the i-th component from $\mathbf{U}$ maps to the j-th component in $\mathbf{V}$.

The authors presented an EM algorithm to learn this mixture from data. As in the above model-based COD approaches, outliers are identified by seeking a low estimate of $f(\mathbf{y}|\mathbf{x})$.

2.2.3 Our Work

In Chapter 4, we focus on the univariate and multivariate conditional outlier detection (COD) problem.

- First, we review and explore the probabilistic approach to the univariate COD problem by finding data instances that fall in the regions of low conditional probability $P(y|\mathbf{x})$. We illustrate the basics of the probabilistic COD approach that consists of two phases: *data modeling* and *outlier scoring*. We revisit the previous work that fits this approach and set the baseline for our following discussion.
- Second, we extend the probabilistic approach to address the multivariate COD problem. By applying the same definition regarding conditional outliers, we present a new framework that aims to find data instances that fall in the regions of low conditional joint probability $P(\mathbf{y}|\mathbf{x})$. To build a data model, we employ the decomposable multi-label classification (MLC) models that represent the conditional joint probability using a collection of discriminative probabilistic models. We present how to compute reliable multivariate conditional outlier score by exploiting the decomposable structure of the data model.
- Third, by recognizing a disconnect in the development of unconditional and conditional outlier methods, we develop and present a new COD method that builds upon unconditional outlier methods. We propose a new framework, Ratio of Outlier Scores (ROS), that computes a conditional outlier score using any outlier score developed for unconditional outlier methods. To cope with high-dimensional data, we present a variant of the ratio-based score that relies on discriminative dimensionality reduction methods.
- Finally, we apply the new ROS approach to the multivariate COD problem. By adopting the structured model building approach of MLC to the ROS framework, we present a decomposition of a multivariate COD problem into a set of univariate COD problems. We investigate how to combine the results from the decomposed problems to compute effective conditional outlier scores. The resulting methods complement the probabilistic approaches to multivariate conditional outliers.

3.0 MODELING AND PREDICTION OF MULTIVARIATE RESPONSES

This chapter focuses on the *Multi-Label Classification* (MLC) [Tsoumakas and Katakis, 2007, Tsoumakas et al., 2010, Zhang and Zhou, 2013] problem, where our goal is to predict the best multivariate output ($\mathbf{y}$) for a given input ($\mathbf{x}$). As pointed out earlier in the thesis, different models may be used to support the MLC prediction problem. In our work, we focus on probabilistic models $P(\mathbf{Y}|\mathbf{X})$ that attempt to represent the relations between inputs ($\mathbf{x}$) and outputs ($\mathbf{y}$) in data to make the predictions.

We present three different probabilistic models of $P(\mathbf{Y}|\mathbf{X})$ to represent the relations between inputs and outputs. For each model, we develop algorithms for learning the model from data and for predicting the best response for a given input. The key difference among the methods is the assumption that each method makes about the input-output relations, as well as relations among individual output variables.

First, we develop the *Conditional Tree-structured Bayesian Networks* (CTBN) model that restricts the dependence relations among the output variables to a *directed tree*. Section 3.2 describes the tree representation and parameterization of the model, presents a learning algorithm that efficiently discovers the optimal (tree-structured) dependence relations, and develops a linear-time prediction algorithm to find the maximum a posteriori (MAP) class assignments for a given input. Second, we develop the *Mixtures-of-Conditional Tree-structured Bayesian Networks* (MC) framework that builds a mixture ensemble of multiple tree-structured models (CTBNs) to better represent the dependence relations among the response variables. Section 3.3 briefly reviews the basics of the Mixtures-of-Trees (MT) [Meilă and Jordan, 2000] framework, on which our mixture model is based, and describes the representation of the new mixture model. The section also presents algorithms for learning the parameters of the mixture, finding multiple tree structures, and inferring the MAP output

configurations. Third, we develop the *Multi-Label Mixtures-of-Experts* (ML-ME) framework that combines MLC models in the *classifier chains family* — our generalization of structured MLC models that decompose the class posterior distribution $P(Y_1, ..., Y_d|\mathbf{X})$ using a product of posterior distributions over components of the output space. Section 3.4 describes the details of the classifier chains family, and briefly reviews the Mixtures-of-Experts [Jacobs et al., 1991] framework. The section then presents algorithms for learning the ML-ME models from data and making multi-label predictions from input data instances.

The rest of this chapter is structured as follows. In Section 3.1, we formally define the prediction problem and introduce the notation used in this chapter. Sections 3.2-3.4 describe in depth the three solutions we propose to model $P(\mathbf{Y}|\mathbf{X})$ and algorithms for their learning and prediction. At the end of Sections 3.3 and 3.4, we report the experimental results on multiple real-world datasets and demonstrate the effectiveness of our solutions compared with the existing state-of-the-art methods. Section 3.5 summarizes our contributions and concludes the chapter.

3.1 PROBLEM DEFINITION AND NOTATION

Multi-Label Classification (MLC) is a classification problem in which each data instance is associated with d binary class variables $Y_1, ..., Y_d$. We are given labeled training data $D = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$, where $\mathbf{x}^{(n)} = (x_1^{(n)}, ..., x_m^{(n)})$ is the m-dimensional feature variable of the n-th instance (input) and $\mathbf{y}^{(n)} = (y_1^{(n)}, ..., y_d^{(n)})$ is its d-dimensional class variable (output). We want to learn a function h that fits D and assigns to each instance a class vector ($h : \mathbb{R}^m \rightarrow \{0, 1\}^d$).

One approach to this task is to model and learn the *joint conditional distribution* $P(\mathbf{Y}|\mathbf{X})$ from D. Assuming the 0-1 loss function, the optimal classifier h^* assigns to each instance $\mathbf{x}$ the *maximum a posteriori* (MAP) assignment of class variables:

$$h^*(\mathbf{x}) = \arg\max_{y_1,...,y_d} P(Y_1 = y_1, ..., Y_d = y_d|\mathbf{X} = \mathbf{x}) \tag{3.1}$$

The key challenge in modeling, learning and performing MAP inference is that the number of configurations defining $P(\mathbf{Y}|\mathbf{X})$ is exponential in d. Overcoming this bottleneck is critical for obtaining efficient MLC solutions.

In this chapter, we use the following notations:

- $\mathbf{X}$, $\mathbf{x}$ — Input (feature) variable and value
- $\mathbf{Y}$, $\mathbf{y}$ — Output (class) variable and value
- m — Input dimensionality
- d — Output dimensionality
- N — Number of data instances
- n — Index of data instance
- T — A tree-structured MLC model (Section 3.2)
- K — Number of base MLC models in a mixture (Sections 3.3 and 3.4)
- k — Index of base MLC model in a mixture (Sections 3.3 and 3.4)
- T_k, M_k — A base MLC model (with an index k) in a mixture (Sections 3.3 and 3.4)
- $\Theta_T = \{\theta_{T_1}, ..., \theta_{T_K}\}$ — Parameters for base MLC models (Section 3.3)
- λ_k — Mixture coefficient with an index k (Section 3.3)
- $\Theta_M = \{\theta_{M_1}, ..., \theta_{M_K}\}$ — Parameters for base MLC models (Section 3.4)
- $\Theta_G = \{\theta_{G_1}, ..., \theta_{G_K}\}$ — Parameters for a gate (Section 3.4)

3.2 CONDITIONAL TREE-STRUCTURED BAYESIAN NETWORKS

In this section, we present our probabilistic approach, which we refer to as *Conditional Tree-sturctured Bayesian Networks* (CTBN) [Batal et al., 2013], to the MLC problem. In the CTBN model, the feature vector $\mathbf{X}$ is defined to be a common parent for all class variables (similar to BR [Boutell et al., 2004, Clare and King, 2001]). In addition to $\mathbf{X}$, each class variable can have at most another class variable as a parent (without creating a cycle). That is, the conditional dependence relations between the class variables follow a *directed tree*. We chose to restrict the dependence structure to a tree because (1) the optimal structure can be learned using a simple and efficient learning algorithm, and (2) the prediction can be

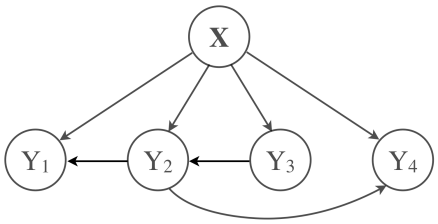


Figure 3.1: An example CTBN.

done efficiently using exact inference. Below we describe the details of our CTBN method.

3.2.1 Representation

Let T be a CTBN model and let $Y_{\pi(i,T)}$ denote the parent class of class variable Y_i in T.[1] The joint distribution of class vector $(y_1, ..., y_d)$ conditioned on feature vector $\mathbf{x}$ is expressed as follows:

$$P(y_1, ..., y_d|\mathbf{x}, T) = \prod_{i=1}^{d} P(y_i|\mathbf{x}, y_{\pi(i,T)}), \tag{3.2}$$

In Figure 3.1, we show an example CTBN with four class variables (Y_1, Y_2, Y_3, Y_4). The joint conditional distribution of class assignment (y_1, y_2, y_3, y_4) given $\mathbf{x}$ according to this network is defined as follows:

$$P(y_1, y_2, y_3, y_4|\mathbf{x}) = P(y_3|\mathbf{x}) \cdot P(y_2|\mathbf{x}, y_3) \cdot P(y_1|\mathbf{x}, y_2) \cdot P(y_4|\mathbf{x}, y_2)$$

The parameterization of the CTBN model corresponds to specifying the conditional probability distribution (CPD) of each class variable Y_i conditioned on its parents: $P(Y_i|\mathbf{X}, Y_{\pi(i,T)})$. The standard parameterization of Bayesian networks uses conditional probability tables (CPT) to define the distribution of each variable conditioned on every possible configuration of its parents. However, the CPT style parameterization is not feasible for the CTBN model. The reason is that the feature vector $\mathbf{X}$, which is a common parent for all variables,

[1] By convention, $Y_{\pi(i,T)} = \{\}$ if Y_i in T does not have a parent class.

can be a high-dimensional vector of continuous values, discrete values or a mixture of both (we cannot enumerate all possible configurations of $\mathbf{X}$).

To overcome this difficulty, we represent the CPDs using probabilistic prediction functions. More specifically, for each class variable $Y_i : i \in \{1, ..., d\}$, we approximate its CPD by learning a different probabilistic classifier $f_{iv}(\mathbf{X})$ for each possible value $v \in \{0, 1\}$ of the parent class variable:

$$\widetilde{P}(Y_i|\mathbf{X} = \mathbf{x}, Y_{\pi(i,T)} = v) = f_{iv}(\mathbf{x}), \qquad v \in \{0, 1\} \tag{3.3}$$

Note that we can use several standard probabilistic classifiers in the CTBN model, such as logistic regression, naïve Bayes, relevance vector machine or the maximum entropy model. In our experiments, we use logistic regression with L_2 regularization. The parameters of logistic regression are trained to maximize the regularized conditional likelihood of the training data and can be efficiently obtained using convex optimization techniques.

3.2.2 Learning the Structure

In this section, we describe how to automatically learn the structure from data. Our objective is to find the tree structure that best approximates the conditional joint distribution $P(\mathbf{Y}|\mathbf{X})$. This can be equivalently stated as finding the tree that maximizes the conditional log-likelihood (CLL) of validation data. To do this, we partition the data into two parts: training data D_t and hold-out data D_h. Given a CTBN T, we use D_t to train its parameters, which corresponds to learning classifiers $\widetilde{P}(Y_i|\mathbf{X}, Y_{\pi(i,T)})$ as described above. On the other hand, we use D_h to compute the score of T, which we define to be the CLL of D_h using T (adopting the standard i.i.d. assumption):

$$\begin{aligned} Score(T) &= CLL(D_h|T) \\ &= \sum_{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in D_h} \sum_{i=1}^{d} \log\left(\widetilde{P}(y_i^{(n)}|\mathbf{x}^{(n)}, y_{\pi(i,T)}^{(n)})\right) \end{aligned} \tag{3.4}$$

In the following, we provide an algorithm to efficiently obtain the optimal CTBN T^* (the model that has the maximum score) without having to explicitly evaluate all of the exponentially many possible tree structures. Let us start by defining a weighted directed graph

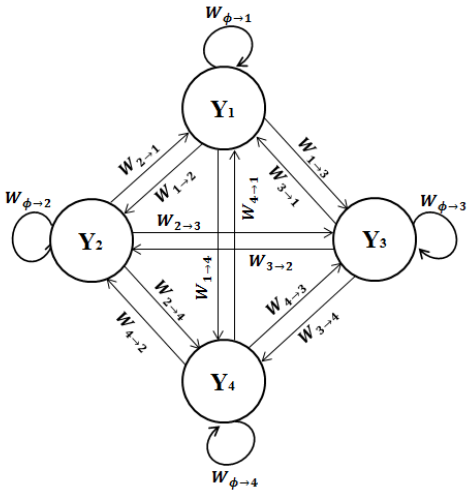


Figure 3.2: The complete directed graph G for four class variables. The weights of the edges are defined using Equations (3.5) and (3.6). The optimal CTBN is obtained by running a maximum branching algorithm on G.

$G = (V, E)$ as follows:

- There is one vertex V_i for each class variable Y_i.
- There is a directed edge $E_{j\to i}$ from each vertex V_j to each vertex V_i (G is complete). Furthermore, each vertex V_i has a self loop $E_{i\to i}$.
- The weights of the edges are defined as follows:
  - The weight of edge $E_{j\to i}$, denoted as $W_{j\to i}$, is the CLL of class Y_i conditioned on $\mathbf{X}$ and Y_j:

$$W_{j\to i} = \sum_{(\mathbf{x}^{(n)},\mathbf{y}^{(n)})\in D_h} \log\left(\widetilde{P}(y_i^{(n)}|\mathbf{x}^{(n)}, y_j^{(n)})\right) \text{ if } i \neq j \tag{3.5}$$

  - The weight of self-loop $E_{i\to i}$, denoted as $W_{\phi\to i}$, is the CLL of class Y_i conditioned only on $\mathbf{X}$:

$$W_{\phi\to i} = \sum_{(\mathbf{x}^{(n)},\mathbf{y}^{(n)})\in D_h} \log\left(\widetilde{P}(y_i^{(n)}|\mathbf{x}^{(n)})\right) \tag{3.6}$$

By using this definition of edge weights (Equations (3.5) and (3.6)) and switching the order of summation in Equation (3.4), we can rewrite the score of T simply as the sum of

Algorithm 1 Find-an-optimal-CTBN-structure

Input: Training data D_t; validation data D_h
Output: Optimal CTBN T^*

1: **for** $i = 1$ **to** d **do**
2: Learn $\tilde{P}(Y_i|\mathbf{X})$ from D_t
3: Compute $W_{\phi \to i}$ on D_h:
$W_{\phi \to i} = \sum_{(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \in D_h} \log \left(\tilde{P}(y_i^{(k)} | \mathbf{x}^{(k)}) \right)$
4: **for** $j = 1$ **to** d **do**
5: **if** $j \neq i$ **then**
6: Learn $\tilde{P}(Y_i|\mathbf{X}, Y_j)$ from D_t
7: Compute $W_{j \to i}$ on D_h:
$W_{j \to i} = \sum_{(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \in D_h} \log \left(\tilde{P}(y_i^{(k)} | \mathbf{x}^{(k)}, y_j^{(k)}) \right)$
8: **end if**
9: **end for**
10: **end for**
11: Construct the weighted digraph $G(V, E)$ using weights $W_{\phi \to i}$ and $W_{j \to i}$
12: **return** T^* = find_maximum_weight_branching(G)

its edge weights (by convention, a node without a parent has a self loop):

$$Score(T) = \sum_{i=1}^{d} W_{\pi(i,T) \to i}$$

Now we have transformed the problem of finding the optimal CTBN into the problem of finding the tree in G that has the maximum sum of edge weights. The solution can be obtained by solving the maximum branching (arborescence) problem [Tarjan, 1977], which finds the maximum weight directed tree in weighted directed graphs. Note that although this problem is similar in spirit to the problem of finding the maximum spanning tree in undirected graphs, the algorithms are quite different because applying a maximum spanning tree algorithm on a directed graph do not guarantee an optimal solution.

3.2.2.1 Complexity Algorithm 1 outlines how to learn the optimal CTBN. Lines 2-10 compute the edge weights (according to Equations (3.5) and (3.6)) for the complete directed graph G (see Figure 3.2). Doing so requires estimating $\widetilde{P}(Y_i|\mathbf{X}, Y_j)$ for all d^2 pairs of class variables, which in turn requires learning different probabilistic classifiers as described in Section 3.2.1. Line 12 finds the maximum branching in G, which can be obtained in $O(d^2)$

Algorithm 2 Predict-CTBN

Input: Instance $\mathbf{x}$; CTBN T
Output: Prediction of $\mathbf{x}$ according to T: $\mathbf{y}^*$

1: **for** Each node i (class Y_i) in the post-order traversal of T **do**
2: Send message $\lambda_{i\to j}$ to its parent $j = \pi(i, T)$:
$$\lambda_{i\to j}(y_j) = \max_{y_i} \left[\log \tilde{P}(y_i|\mathbf{x}, y_j) + \sum_{h\in \text{child}(i,T)} \lambda_{h\to i}(y_i)\right]$$
3: **end for**
4: **for** Each node i (class Y_i) in the pre-order traversal of T **do**
5: Compute its optimal prediction y_i^*:
$$y_i^* = \arg\max_{y_i} \left[\log \tilde{P}(y_i|\mathbf{x}, y_j^*) + \sum_{h\in \text{child}(i,T)} \lambda_{h\to i}(y_i)\right]$$
6: **end for**
7: **return** $\mathbf{y}^*$

using Tarjan's implementation [Tarjan, 1977] (this algorithm is as fast as Prim's algorithm for finding undirected maximum spanning tree). Therefore, the overall complexity is $O(d^2)$ times the complexity of learning the probabilistic classifiers (e.g., logistic regression).

3.2.3 Prediction

In order to make a prediction for a new instance $\mathbf{x}$, we should find the MAP assignment of class variables (solve Equation (3.1)). This problem is NP-hard for general Bayesian networks. However, since we have restricted our structure to a tree, we can solve the problem efficiently using exact inference.

In particular, we perform inference using a variant of the max-sum algorithm [Koller and Friedman, 2009][2] that we design for the CTBN model. This algorithm first computes the local CPTs for each node Y_i by applying the corresponding classifier for each possible value of the parent class (see Equation (3.3)). After that, it performs two phases to obtain the optimal prediction. In the first phase, the algorithm sends messages upward (from the leaves to the root) where each node Y_i applies the following steps: (i) compute the sum of the logarithm of its local CPT and all messages sent from its children, (ii) maximize the result over its value, and (iii) send it to the parent node (line 1-3, Algorithm 2). In the second phase, the algorithm propagates the optimal assignments downward (line 4-6, Algorithm 2).

[2]The max-sum algorithm is analogous to the sum-product algorithm for computing conditional probability queries.

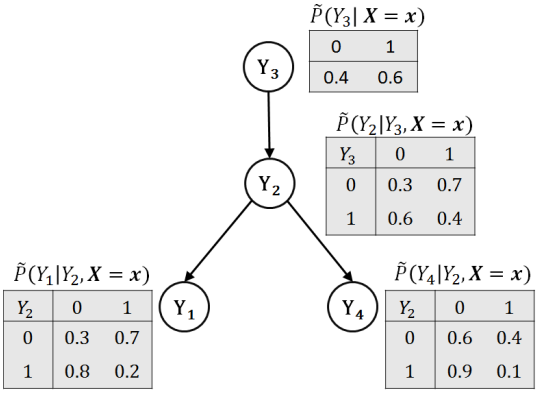


Figure 3.3: An example showing the CPTs of a CTBN model for a specific instance $\mathbf{x}$.

3.2.3.1 Complexity The inference algorithm described above runs in $O(d)$, where d is the number of class variables.

Example 3. *Consider the example in Figure 3.3, where we show the conditional probability tables of a CTBN model for a specific instance* $\mathbf{x}$ *(obtained by applying the classifiers on* $\mathbf{x}$*). The optimal prediction for* $\mathbf{x}$ *is* $(Y_3 = 0, Y_2 = 1, Y_1 = 0, Y_4 = 0)$*, which can be obtain by running our exact inference algorithm.*

3.2.4 Experiments

Experimental results of CTBN are reported in Section 3.3.6.

3.2.5 Discussion

In this chapter, we proposed a novel probabilistic approach to the MLC problem. Our approach encodes the conditional dependence relations between the class variables using a special tree-structured Bayesian network, whose conditional distributions are defined using probabilistic classifiers. We presented an efficient algorithm to learn the tree structure that maximizes the conditional log-likelihood. Furthermore, we presented an efficient exact inference algorithm that has a linear complexity in the number of class variables.

Although our CTBN approach effectively discovers the dependence relations in data and builds an accurate predictive multi-label data model, the approach may not fully recover the underlying dependence relations due to its structural restriction. In the following, we develop two variants of mixture frameworks that work with the structured probabilistic MLC models. We show how to reveal and learn various dependence relations among inputs and outputs, which a single MLC model cannot capture, and how to combine them into an ensemble to achieve a higher predictive accuracy.

3.3 MIXTURES-OF-CONDITIONAL TREE-STRUCTURED BAYESIAN NETWORKS

In this section, we describe the *Mixture of Conditional Tree-structured Bayesian Networks* approach that uses the Mixtures-of-Trees [Meilă and Jordan, 2000] framework in combination with the CTBN classifiers (Section 3.2) to improve the classification accuracy of MLC tasks. Our mixture ensemble aims to learn a more accurate representation of the class posterior distribution $P(\mathbf{Y}|\mathbf{X})$ by leveraging the computational advantages of conditional tree-structured models and the abilities of mixtures to compensate for tree-structured restrictions. Below we show the representation of this new mixture model, and develop algorithms for learning the structures and parameters from data and for performing multi-label predictions using a learned mixture.

3.3.1 Preliminary: Mixtures-of-Trees Framework

The MLC solution we propose in this section combines multiple base MLC classifiers using the *Mixtures-of-Trees* (MT) [Meilă and Jordan, 2000] framework, which uses a mixture of multiple trees to define a generative model of $P(\mathbf{Y})$ for discrete multi-dimensional domains. The base classifiers we use are based on the *Conditional Tree-structured Bayesian Networks* (CTBN) (Section 3.2). To begin with, we briefly review the basics of MT and CTBN.

MT consists of a set of *trees* that are combined using *mixture coefficients* λ_k to represent

the joint distribution $P(\mathbf{y})$. The model is defined by the following decomposition:

$$P(\mathbf{y}) = \sum_{k=1}^{K} \lambda_k P(\mathbf{y}|T_k), \tag{3.7}$$

where $P(\mathbf{y}|T_k)$ are called *mixture components* that represent the distribution of outputs defined by the k-th tree T_k. Note that a mixture can be understood as a soft-multiplexer, where we have a hidden selector variable which takes a value $k \in \{1, ..., K\}$ with probability λ_k. That is, by having a convex combination of mutually complementary tree-structured models, MT aims at achieving a more expressive and accurate model.

While MT is not as computationally efficient as individual trees, it has been considered as a useful approximation at a fraction of the computational cost for learning general graphical models [Kirshner and Smyth, 2007]. MT has been successfully adopted in a range of applications, including modeling of handwriting patterns, medical diagnostic network, automated application screening, gene classification and identification [Meilă and Jordan, 2000], face detection [Ioffe and Forsyth, 2001b], video tracking [Ioffe and Forsyth, 2001a], road traffic modeling [Šingliar and Hauskrecht, 2007] and climate modeling [Kirshner and Smyth, 2007].

In this section, we apply the MT framework in context of MLC. In particular, we combine MT with CTBN to model individual trees. CTBN is a recently proposed probabilistic MLC method that has been shown to be competitive and efficient on a range of domains. CTBN defines $P(\mathbf{Y}|\mathbf{X})$ using a collection of classifiers that model relations in between features and individual labels, which are tied together using a special Bayesian network structure that approximates the dependence relations among the class variables. In modeling of the dependences, it allows each class variable to have at most one other class variable as a parent (without creating a cycle) besides the feature vector $\mathbf{X}$.

Although our proposed method is motivated by MT, there are significant extensions and differences. We summarize the key distinctions below.

1. *Model*: Our model represents $P(\mathbf{Y}|\mathbf{X})$, the class posterior distribution for MLC, using CTBNs that each consists of a collection of logistic regression models, linked together by a directed tree; on the other hand, the MT model [Meilă and Jordan, 2000] represents the joint distribution $P(\mathbf{Y})$ using standard tree-structured Bayesian networks.

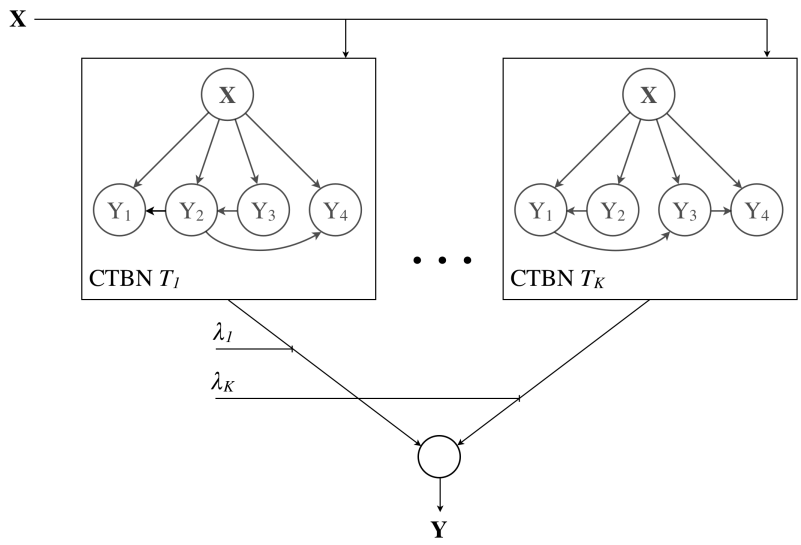


Figure 3.4: An example MC.

2. *Structure learning*: Our structure learning algorithm optimizes $P(\mathbf{Y}|\mathbf{X})$ using weighted conditional log-likelihood criterion; while MT relies on the standard Chow-Liu algorithm [Koller and Friedman, 2009] that optimizes $P(\mathbf{Y})$ using mutual information.
3. *Parameter learning*: Not surprisingly, both our parameter learning method and that of MT rely on the EM algorithm. However, the criteria and optimization techniques are very different. For example, the M-step of our algorithm corresponds to learning of instance-weighted logistic regression classifiers; while that of MT is based on simple (weighted) counting.

3.3.2 Representation

By following the definition of MT in Equation (3.7), MC defines the multivariate posterior distribution of class vector $\mathbf{y} = (y_1, ..., y_d)$ as:

$$P(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} \lambda_k P(\mathbf{y}|\mathbf{x}, T_k), \tag{3.8}$$

where $\lambda_k \geq 0, \forall k$; and $\sum_{k=1}^{K} \lambda_k = 1$. Here each *mixture component* $P(\mathbf{y}|\mathbf{x}, T_k)$ is the distribution defined by CTBN T_k (as in Equation (3.2)) and *mixture coefficients* are denoted by λ_k. Figure 3.4 depicts an example MC model, which consists of K CTBNs and the mixture coefficients λ_k.

3.3.3 Parameter Learning

In this section, we describe how to learn the parameters of MC with the assumption that the structures of individual CTBNs are known and fixed. The parameters of the MC model are the mixture coefficients $\{\lambda_1, ..., \lambda_K\}$ as well as the parameters of each CTBN in the mixture $\{\theta_1, ..., \theta_K\}$.

Given training data $D = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\} : n \in 1, ..., N$, the objective is to optimize the log-likelihood of D, which we refer to as the *observed log-likelihood*.

$$\sum_{n=1}^{N} \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \lambda_k P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k)$$

However, this is very difficult to directly optimize because it contains the log of the sum. Hence, we cast this optimization in the expectation-maximization (EM) framework. Let us associate each instance $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ with a hidden variable $z^{(n)} \in \{1, ..., K\}$ indicating which CTBN it belongs to. The *complete log-likelihood* (assuming $z^{(n)}$ are observed) is:

$$\sum_{n=1}^{N} \log P(\mathbf{y}^{(n)}, z^{(n)}|\mathbf{x}^{(n)}) \tag{3.9}$$

$$= \sum_{n=1}^{N} \log \prod_{k=1}^{K} P\left(\mathbf{y}^{(n)}, T_k|\mathbf{x}^{(n)}\right)^{\mathbb{1}[z^{(n)}=k]} \tag{3.10}$$

$$= \sum_{n=1}^{N} \log \prod_{k=1}^{K} \left[\lambda_k P\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k\right)\right]^{\mathbb{1}[z^{(n)}=k]}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{1}[z^{(n)} = k] \left[\log \lambda_k + \log P\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k\right)\right],$$

where $\mathbb{1}[z^{(n)} = k]$ is the indicator function, which is one if the n-th instance belongs to the k-th CTBN and zero otherwise; and λ_k is the mixture coefficient of CTBN T_k, which can be interpreted as its prior probability in the data.

The EM algorithm iteratively optimizes the *expected complete log-likelihood*, which is always a lower bound to the observed log-likelihood [Moon, 1996]. In the *E-step*, the expectation is computed with the current set of parameters; in the *M-step*, the parameters of the mixture $(\lambda_k, \theta_k : k = \{1, ..., K\})$ are relearned to maximize the expected complete log-likelihood. In the following, we describe our parameter learning algorithm by deriving the E-step and the M-step for MC.

E-step In the E-step, we compute the expectation of the hidden variables. Let $\gamma_k(n)$ denote $P(z^{(n)} = k|\mathbf{y}^{(n)}, \mathbf{x}^{(n)})$, the posterior of the hidden variable $z^{(n)}$ given the observations and the current parameters. Using Bayes rule, we write:

$$\gamma_k(n) = \frac{\lambda_k P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k)}{\sum_{k'} \lambda_{k'} P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_{k'})} \tag{3.11}$$

M-step In the M-step, we learn the model parameters $\{\lambda_1, ..., \lambda_K, \theta_1, ..., \theta_K\}$ that maximize the expected complete log-likelihood, which is a lower bound of the observed log-likelihood. Let us first define the following two quantities:

$$\Gamma_k = \sum_{n=1}^{N} \gamma_k(n), \qquad w_k(n) = \frac{\gamma_k(n)}{\Gamma_k}$$

Γ_k can be interpreted as the number of observations that belongs to the k-th CTBN (hence, $\sum_{k=1}^{K} \Gamma_k = N$), and $w_k(n)$ is the renormalized posterior $\gamma_k(n)$, which can be interpreted as the weight of the n-th instance on the k-th CTBN.

Note that when taking the expectation of the complete log-likelihood (Equation (3.9)), only the indicator $\mathbb{1}[z^{(n)} = k]$ is affected by the expectation. By using the notations introduced above, we rewrite the expected complete log-likelihood:

$$\begin{aligned} &\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k(n) \left[\log \lambda_k + \log P\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k\right)\right] \\ &= \sum_{k=1}^{K} \Gamma_k \log \lambda_k + \sum_{k=1}^{K} \Gamma_k \sum_{n=1}^{N} w_k(n) \log P\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k\right) \end{aligned} \tag{3.12}$$

We wish to maximize Equation (3.12) with respect to $\{\lambda_1, ..., \lambda_K, \theta_1, ..., \theta_K\}$ subject to the constraint $\sum_{k=1}^{K} \lambda_k = 1$. Notice that Equation (3.12) consists of two terms and each term

Algorithm 3 Learn-MC-parameters

Input: Training data D; base CTBNs $T_1, ..., T_K$
Output: Model parameters $\{\theta_1, ..., \theta_K, \lambda_1, ..., \lambda_K\}$

repeat
  E-step:
  for $k = 1$ **to** K, $n = 1$ **to** N **do**
    Compute $\gamma_k(n)$ using Equation (3.11)
  end for
  M-step:
  for $k = 1$ **to** K **do**
    $\Gamma_k = \sum_{n=1}^{N} \gamma_k(n)$
    $w_k(n) = \gamma_k(n)/\Gamma_k$
    $\lambda_k = \Gamma_k/N$
    $\theta_k = \arg\max \sum_{n=1}^{N} w_k(n) \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k)$
  end for
until *convergence*

has a disjoint subset of parameters – which allows us to maximize Equation (3.12) term by term. By maximizing the first term with respect to λ_j (the mixture coefficient of T_j), we obtain:

$$\lambda_j = \frac{\Gamma_j}{\sum_{k=1}^{K} \Gamma_k} = \frac{\Gamma_j}{N}$$

To maximize the second term, we train θ_j (the parameters of T_j) to maximize:

$$\theta_j = \arg\max \sum_{n=1}^{N} w_j(n) \log P(y^{(n)}|x^{(n)}, T_j) \tag{3.13}$$

It turns out Equation (3.13) is the instance-weighted log-likelihood, and we use instance-weighted logistic regression to optimize it. Algorithm 3 outlines our parameter learning algorithm.

3.3.3.1 Complexity E-step: We compute $\gamma_k(n)$ for each instance on every CTBN. To compute $\gamma_k(n)$, we should estimate $P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, T_k)$, which requires applying the logistic regression classifiers for each node of T_k, which requires $O(md)$ multiplications. Hence, the complexity of the E-step is $O(KNmd)$.

M-step: The major computational cost of the M-step is to learn the instance-weighted logistic regression models for the nodes of every CTBN. Hence, the complexity is $O(Kd)$ times the complexity of learning logistic regression.

3.3.4 Structure Learning

In this section, we describe how to automatically learn multiple CTBN structures from data. We apply a sequential boosting-like heuristic, where in each iteration we learn the structure that focuses on the instances that are not well predicted by the previous structures (i.e., the MC model learned so far). In the following, we first describe how to learn a single CTBN structure from instance-weighted data. After that, we describe how to re-weight the instances and present our algorithm for learning the overall MC model.

Learning a Single CTBN Structure on Weighted Data The goal here is to discover the CTBN structure that maximizes the weighted conditional log-likelihood (WCLL) on $\{D, \Omega\}$, where $D = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$ is the data and $\Omega = \{\omega^{(n)}\}_{n=1}^{N}$ is the weight for each instance. We do this by partitioning D into two parts: training data D_{tr} and hold-out data D_h. Given a CTBN structure T, we train its parameters using D_{tr} and the corresponding instance weights. On the other hand, we use WCLL of D_h to score T.

$$\begin{aligned} Score(T) &= \sum_{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in D_h} \omega^{(n)} \log P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, T) \\ &= \sum_{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in D_h} \sum_{i=1}^{d} \omega^{(n)} \log P(y_i^{(n)} | \mathbf{x}^{(n)}, y_{\pi(i,T)}^{(n)}) \end{aligned} \tag{3.14}$$

In the following, we describe our algorithm for obtaining the CTBN structure that optimizes Equation (3.14) without having to evaluate all of the exponentially many possible tree structures.

Let us first define a weighted directed graph $G = (V, E)$, which has one vertex V_i for each class label Y_i and a directed edge $E_{j \to i}$ from each vertex V_j to each vertex V_i (i.e., G is complete). In addition, each vertex V_i has a self-loop $E_{i \to i}$. The weight of edge $E_{j \to i}$, denoted as $W_{j \to i}$, is the WCLL of class Y_i conditioned on $\mathbf{X}$ and Y_j:

$$W_{j \to i} = \sum_{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in D_h} \omega^{(n)} \log P(y_i^{(n)} | \mathbf{x}^{(n)}, y_j^{(n)}) \tag{3.15}$$

The weight of self-loop $E_{i\to i}$, denoted as $W_{\phi\to i}$, is the WCLL of class Y_i conditioned only on $\mathbf{X}$. Using the definition of edge weights, Equation (3.14) can be simplified as the sum of the edge weights:

$$Score(T) = \sum_{n=1}^{d} W_{\pi(i,T)\to i}$$

Now we have transformed the problem of finding the optimal tree structure into a problem of finding a tree in G that has the maximum sum of edge weights. The solution can be obtained by solving the maximum branching (arborescence) problem [Edmonds, 1967], which finds the maximum weight tree in a weighted directed graph.

Learning Multiple CTBN Structures In order to obtain multiple CTBN structures for the MC model, we apply the algorithm described above multiple times with different sets of instance weights. We assign the weights such that we give higher weights for poorly predicted instances and lower weights for well-predicted instances.

We start by assigning all instances uniform weights (i.e., all instances are equally important a priori).

$$\omega^{(n)} = 1/N : n = 1, ..., N$$

Using this initial set of weights, we find the initial CTBN structure T_1 (and its parameters θ_1) and set the current model M to be T_1. We then estimate the prediction error margin $\omega^{(n)} = 1 - P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, M)$ for each instance and renormalize such that $\sum_{n=1}^{N} \omega^{(n)} = 1$. We use $\{\omega^{(n)}\}$ to find the next CTBN structure T_2. After that, we set the current model to be the MC model learned by mixing T_1 and T_2 according to Algorithm 3.

We repeat the process by incrementally adding trees to the mixture. To stop the process, we use internal validation approach. Specifically, the data used for learning are split into internal train and test sets. The structure of the trees and parameters are always learned on the internal train set. The quality of the current mixture is evaluated on the internal test set. The mixture growth stops when the log-likelihood on the internal test set for the new mixture is worse than that of the previous mixture. The trees included in the previous mixture are then fixed, and the parameters of the mixture are relearned on the full training data.

3.3.4.1 Complexity

In order to learn a single CTBN structure, we compute edge weights for the complete graph G, which requires estimating $P(Y_i|\mathbf{X}, Y_j)$ for all d^2 pairs of classes. Finding the maximum branching in G can be obtained in $O(d^2)$ using [Tarjan, 1977]. To learn K CTBN structures for the mixture, we repeat these steps K times. Therefore, the overall complexity is $O(d^2)$ times the complexity of learning logistic regression.

3.3.5 Prediction

In order to make a prediction for a new instance $\mathbf{x}$, we want to find the MAP assignment of the class variables (see Equation (3.1)). In general, this requires evaluating all possible assignments of values to d class variables, which is exponential in d.

One important advantage of the CTBN model is that the MAP inference can be done more efficiently by avoiding blind enumeration of all possible assignments. More specifically, the MAP inference on a CTBN is linear in the number of classes ($O(d)$) when implemented using a variant of the max-sum algorithm [Koller and Friedman, 2009] on a tree structure.

However, our MC model consists of multiple CTBNs and the MAP solution may, at the end, require enumeration of exponentially many class assignments. To address this problem, we rely on approximate MAP inference. Two commonly applied MAP approximation approaches are convex programming relaxation via dual decomposition [Sontag, 2010], and simulated annealing using a Markov chain [Yuan et al., 2004]. In this work, we use the latter approach. Briefly, we search the space of all assignments by defining a Markov chain that is induced by local changes to individual class labels. The annealed version of the exploration procedure [Yuan et al., 2004] is then used to speed up the search. We initialize our MAP algorithm using the following heuristic: first, we identify the MAP assignments for each CTBN in the mixture individually, and after that, we pick the best assignment from among these candidates. We have found this (efficient) heuristic to work very well and it often results in the true MAP assignment.

3.3.6 Experiments

3.3.6.1 Datasets We perform experiments on ten publicly available multi-label datasets. These datasets are obtained from different domains such as music recognition (emotions [Trohidis et al., 2008]), semantic image labeling (scene [Boutell et al., 2004] and image [Dembczynski et al., 2010]), biology (yeast [Elisseeff and Weston, 2001]) and text classification (enron and RCV1 [Lewis et al., 2004] datasets). Table 3.1 summarizes the characteristics of the datasets. We show the number of instances (N), number of feature variables (m) and number of class variables (d). In addition, we show two statistics: label cardinality (LC), which is the average number of labels per instance, and distinct label set (DLS), which is the number of all distinct configurations of classes that appear in the data. Note that, for RCV1 datasets, we have used the ten most common labels.

3.3.6.2 Methods We compare the performance of our two algorithms, CTBN that uses a single model (SC) and the mixture-of-CTBNs (MC) model, to multiple MLC baselines. These baselines include: simple binary relevance (BR) independent classification [Clare and King, 2001, Boutell et al., 2004], classification with heterogeneous features (CHF) [Godbole and Sarawagi, 2004], multi-label k-nearest neighbor (MLKNN) [Zhang and Zhou, 2007], instance-based learning by logistic regression (IBLR) [Cheng and Hüllermeier, 2009], classi-

Dataset	N	m	d	**LC**	**DLS**	**Domain**
Emotions	593	72	6	1.87	27	music
Yeast	2,417	103	14	4.24	198	biology
Scene	2,407	294	6	1.07	15	image
Image	2,000	135	5	1.24	20	image
Enron	1,702	1,001	53	3.38	753	text
RCV1_subset1	6,000	8,394	10	1.31	69	text
RCV1_subset2	6,000	8,304	10	1.21	70	text
RCV1_subset3	6,000	8,328	10	1.22	74	text
RCV1_subset4	6,000	8,332	10	1.22	79	text
RCV1_subset5	6,000	8,367	10	1.31	76	text

Table 3.1: Datasets characteristics (N: number of instances, m: number of features, d: number of classes, LC: label cardinality, DLS: distinct label set, DM: domain).

fier chains (CC) [Read et al., 2009], ensemble of classifier chains (ECC) [Read et al., 2009], probabilistic classifier chains (PCC) [Dembczynski et al., 2010], ensemble of probabilistic classifier chains (EPCC) [Dembczynski et al., 2010], multi-label conditional random fields (MLCRF) [Pakdaman et al., 2014], and maximum margin output coding (MMOC) [Zhang and Schneider, 2012].

For all methods, we use the same parameter settings as suggested in their papers: For MLKNN and IBLR, which use the k-nearest neighbor (KNN) method, we use Euclidean distance to measure similarity of instances and we set the number of nearest neighbors to 10 [Zhang and Zhou, 2007, Cheng and Hüllermeier, 2009]; for CC, we set the order of classes to $Y_1 < Y_2, ... < Y_d$ [Read et al., 2009]; for ECC and EPCC, we use 10 CCs in the ensemble [Read et al., 2009, Dembczynski et al., 2010]; finally for MMOC, we set the decoding parameter to 1 [Zhang and Schneider, 2012]. Also note that all of these methods except MLKNN and MMOC are considered as meta-learners because they can work with several base classifiers. To eliminate additional effects that may bias the results, we use L_2-penalized logistic regression for all of these methods and choose their regularization parameters by cross validation. For our MC model, we decide the number of mixture components using our stopping criterion (Section 3.3.4) and we use 150 iterations of simulated annealing for prediction.

3.3.6.3 Evaluation Metrics Evaluating the performance of MLC methods is more difficult than evaluating simple classification methods. The most suitable performance measure is the *exact match accuracy* (EMA), which computes the percentage of instances whose predicted label vectors are exactly the same as their true label vectors:

$$EMA = \sum_{n=1}^{N} \delta(\mathbf{y}^{(n)}, h(\mathbf{x}^{(n)}))$$

However, this measure could be too harsh, especially when the output dimensionality is high. Another very useful measure is the *conditional log-likelihood loss* (CLL-loss), which computes the negative conditional log-likelihood of the test instances:

$$CLL\text{-}loss = \sum_{n=1}^{N} -\log\left(P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)})\right)$$

CLL-loss evaluates how much probability mass is given to the true label vectors (the higher the probability, the smaller the loss).

Other evaluation measures used commonly in MLC literature are based on F1 scores. *Micro F1* aggregates the number of true positives, false positives and false negatives for all classes and then calculates the overall F1 score. On the other hand, *macro F1* computes the F1 score for each class separately and then averages these scores. Note that both measures are not the best for MLC because they do not account for the correlations between classes (see [Dembczynski et al., 2010] and [Zhang and Zhang, 2010]). However, we report them in our performance comparisons as they have been used in other MLC literature [Tsoumakas et al., 2010].

3.3.6.4 Results We have performed *ten-fold cross validation* for all of our experiments. To evaluate the statistical significance of performance difference, we apply paired t-tests at 0.05 significance level. We use markers $*/\circledast$ to indicate whether MC is significantly better/worse than the compared method.

Tables 3.2, 3.3, 3.4 and 3.5 show the performance of the methods in terms of EMA, CLL-loss, micro F1 and macro F1, respectively. We only show the results of MMOC on four datasets (emotions, yeast, scene and image) because it did not finish on the remaining data (MMOC did not finish one round of the learning within a 24 hours time limit). For the same reason, we do not report the results of PCC, EPCC and MLCRF on the enron dataset. Also note that we do not report CLL-loss for MMOC, ECC and EPCC because they do not compute a probabilistic score for a given class assignment.

In terms of EMA (Table 3.2), MC clearly outperforms the other methods on most datasets. MC is significantly better than BR, CHF, MLKNN and CC on all ten datasets, significantly better than IBLR, ECC and MLCRF on nine datasets, significantly better than EPCC and SC on five datasets and significantly better than PCC on four datasets (see the last row of Table 3.2). Although not statistically significant, MC performs better than MMOC on all datasets MMOC is able to finish. MLKNN and IBLR perform poorly on the high-dimensional ($m > 1,000$) datasets because Euclidean distances between data instances become indiscernible in high dimensions.

EMA	BR	CHF	MLKNN	IBLR	CC	ECC	PCC	EPCC	MLCRF	MMOC	SC	MC
Emotions	0.265 ∗	0.300 ∗	0.283 ∗	0.335	0.268 ∗	0.288 ∗	0.317	0.344	0.303 ∗	0.332	0.322	0.346
Yeast	0.151 ∗	0.163 ∗	0.179 ∗	0.204 ∗	0.193 ∗	0.204 ∗	0.230	0.219	0.180 ∗	0.219	0.192 ∗	0.235
Scene	0.541 ∗	0.605 ∗	0.629 ∗	0.644 ∗	0.632 ∗	0.658 ∗	0.666	0.671	0.583 ∗	0.664	0.625 ∗	0.680
Image	0.280 ∗	0.360 ∗	0.346 ∗	0.387 ∗	0.426 ∗	0.413 ∗	0.449	0.442	0.377 ∗	0.448	0.414 ∗	0.463
Enron	0.164 ∗	0.170 ∗	0.078 ∗	0.163 ∗	0.173 ∗	0.180	-	-	-	-	0.167 ∗	0.187
Rcv1_subset1	0.334 ∗	0.357 ∗	0.205 ∗	0.279 ∗	0.429 ∗	0.410 ∗	0.432 ∗	0.420 ∗	0.344 ∗	-	0.441 ∗	0.457
Rcv1_subset2	0.439 ∗	0.465 ∗	0.288 ∗	0.417 ∗	0.516 ∗	0.509 ∗	0.523 ∗	0.516 ∗	0.475 ∗	-	0.531	0.536
Rcv1_subset3	0.466 ∗	0.486 ∗	0.327 ∗	0.446 ∗	0.539 ∗	0.539 ∗	0.548 ∗	0.544 ∗	0.489 ∗	-	0.560	0.561
Rcv1_subset4	0.510 ∗	0.531 ∗	0.354 ∗	0.491 ∗	0.579 ∗	0.569 ∗	0.588	0.576 ∗	0.550 ∗	-	0.592	0.591
Rcv1_subset5	0.439 ∗	0.456 ∗	0.276 ∗	0.411 ∗	0.497 ∗	0.494 ∗	0.519 ∗	0.513 ∗	0.457 ∗	-	0.539	0.540
#win/#tie/#loss	**10/0/0**	**10/0/0**	**10/0/0**	**9/1/0**	**10/0/0**	**9/1/0**	**4/5/0**	**5/4/0**	**9/0/0**	**0/4/0**	**5/5/0**	

Table 3.2: Performance of each method on the benchmark datasets in terms of exact match accuracy (EMA; higher value is better). Marker ∗/⊛ indicates whether MC is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level). The last row shows the total number of win/tie/loss for MC against the compared method (e.g., #win is how many times MC significantly outperforms that method).

CLL-loss	BR	CHF	MLKNN	IBLR	CC	PCC	MLCRF	SC	MC
Emotions	153.5 ∗	147.5 ∗	151.7 ∗	143.0 ∗	169.6 ∗	134.9	139.2 ∗	147.4 ∗	128.8
Yeast	1500.3 ∗	1491.7 ∗	1464.9 ∗	1434.2 ∗	2303.8 ∗	932.1 ⊛	1175.4 ∗	1097.0 ∗	1000.0
Scene	344.7 ∗	318.4 ∗	310.9 ∗	283.9 ∗	395.0 ∗	258.9	313.2 ∗	306.3 ∗	260.1
Image	432.5 ∗	415.9 ∗	425.3 ∗	395.6 ∗	480.3 ∗	354.7	401.4 ∗	388.4 ∗	347.1
Enron	1287.3 ∗	1272.5 ∗	1301.2 ∗	1287.4 ∗	1293.5 ∗	-	-	1437.9 ∗	1224.4
Rcv1_subset1	1443.8 ∗	2144.2 ∗	1873.7 ∗	1379.5 ∗	1701.3 ∗	1034.3 ∗	1369.4 ∗	962.7	951.1
Rcv1_subset2	1207.4 ∗	2223.6 ∗	1687.8 ∗	1172.6 ∗	1398.8 ∗	923.0 ∗	1123.6 ∗	893.5 ∗	855.8
Rcv1_subset3	1207.4 ∗	2156.0 ∗	1674.6 ∗	1168.2 ∗	1500.5 ∗	896.7 ∗	1116.4 ∗	939.7 ∗	837.2
Rcv1_subset4	1072.9 ∗	1759.9 ∗	1532.9 ∗	1034.8 ∗	1282.1 ∗	823.0 ∗	951.4 ∗	790.7 ∗	770.6
Rcv1_subset5	1267.0 ∗	2283.6 ∗	1795.5 ∗	1234.7 ∗	1422.0 ∗	1009.0 ∗	1192.4 ∗	924.0 ∗	894.3
#win/#tie/#loss	**10/0/0**	**10/0/0**	**10/0/0**	**10/0/0**	**10/0/0**	**5/3/1**	**9/0/0**	**9/1/0**	

Table 3.3: Performance of each method in terms of conditional log-likelihood loss (CLL-loss; smaller value is better). Marker ∗/⊛ indicates whether MC is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level). The last row shows the total number of win/tie/loss for MC against the compared method.

Micro F1	BR	CHF	MLKNN	IBLR	CC	ECC	PCC	EPCC	MLCRF	MMOC	SC	MC
Emotions	0.645 ∗	0.672	0.656 ∗	0.692	0.621 ∗	0.652 ∗	0.664 ∗	0.688	0.684	0.687	0.678	0.693
Yeast	0.635	0.637	0.646	0.661 ⊛	0.628	0.631	0.645	0.650	0.619 ∗	0.651	0.631	0.640
Scene	0.696 ∗	0.722 ∗	0.736	0.758	0.697 ∗	0.724 ∗	0.722 ∗	0.743	0.713 ∗	0.711 ∗	0.717 ∗	0.745
Image	0.479 ∗	0.541 ∗	0.504 ∗	0.573	0.550	0.563	0.565	0.577	0.558	0.572	0.561	0.573
Enron	0.551	0.569 ⊛	0.450 ∗	0.566 ⊛	0.577 ⊛	0.583 ⊛	-	-	-	-	0.552	0.556
Rcv1_subset1	0.503 ∗	0.516	0.257 ∗	0.459 ∗	0.511 ∗	0.525	0.510 ∗	0.529	0.505 ∗	-	0.512 ∗	0.525
Rcv1_subset2	0.568 ∗	0.584	0.317 ∗	0.546 ∗	0.586	0.589	0.588	0.591	0.582 ∗	-	0.591	0.587
Rcv1_subset3	0.576 ∗	0.592	0.364 ∗	0.564 ∗	0.594	0.610 ⊛	0.594	0.613 ⊛	0.590	-	0.596	0.599
Rcv1_subset4	0.622 ∗	0.637	0.404 ∗	0.606 ∗	0.640	0.646 ⊛	0.644 ⊛	0.650 ⊛	0.635	-	0.638	0.635
Rcv1_subset5	0.582 ∗	0.597	0.314 ∗	0.566 ∗	0.595	0.603	0.600	0.605 ⊛	0.589 ∗	-	0.598	0.597
#win/#tie/#loss	**8/2/0**	**2/7/1**	**8/2/0**	**5/3/2**	**3/6/1**	**2/5/3**	**3/5/1**	**0/6/3**	**5/4/0**	**1/3/0**	**2/8/0**	

Table 3.4: Performance of each method in terms of micro F1 (higher value is better). Marker ∗/⊛ indicates whether MC is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level). The last row shows the total number of win/tie/loss for MC against the compared method.

Macro F1	BR	CHF	MLKNN	IBLR	CC	ECC	PCC	EPCC	MLCRF	MMOC	SC	MC
Emotions	0.632 ∗	0.667	0.656	0.690	0.620 ∗	0.643 ∗	0.659	0.683	0.667	0.679	0.670	0.686
Yeast	0.457 ∗	0.461 ∗	0.478	0.498 ⊛	0.467	0.477	0.486	0.496 ⊛	0.451 ∗	0.473	0.467	0.477
Scene	0.703 ∗	0.730 ∗	0.743	0.765	0.709 ∗	0.740	0.729 ∗	0.753	0.721 ∗	0.721 ∗	0.728 ∗	0.755
Image	0.486 ∗	0.546 ∗	0.516 ∗	0.581	0.562	0.571	0.575	0.586	0.560 ∗	0.578	0.572	0.584
Enron	0.478 ⊛	0.479	0.411 ∗	0.475	0.484 ⊛	0.482 ⊛	-	-	-	-	0.470	0.470
Rcv1_subset1	0.495 ∗	0.511	0.273 ∗	0.463 ∗	0.506 ∗	0.516	0.504 ∗	0.521	0.500 ∗	-	0.507	0.517
Rcv1_subset2	0.503 ∗	0.526	0.264 ∗	0.475 ∗	0.531	0.539	0.531	0.538	0.516 ∗	-	0.536	0.531
Rcv1_subset3	0.513 ∗	0.536	0.278 ∗	0.497 ∗	0.547	0.558 ⊛	0.548	0.561 ⊛	0.531	-	0.543	0.542
Rcv1_subset4	0.499 ∗	0.519	0.269 ∗	0.477 ∗	0.534 ⊛	0.540 ⊛	0.534 ⊛	0.539 ⊛	0.515	-	0.526	0.522
Rcv1_subset5	0.500 ∗	0.526	0.257 ∗	0.487 ∗	0.536	0.538 ⊛	0.534	0.538 ⊛	0.513 ∗	-	0.536	0.527
#win/#tie/#loss	**9/0/1**	**3/7/0**	**7/3/0**	**5/4/1**	**3/5/2**	**1/5/4**	**2/6/1**	**0/5/4**	**6/3/0**	**1/3/0**	**1/9/0**	

Table 3.5: Performance of each method in terms of macro F1 (higher value is better). Marker ∗/⊛ indicates whether MC is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level). The last row shows the total number of win/tie/loss for MC against the compared method.

Interestingly, MC shows significant improvements over SC (a single CTBN) on five datasets, while SC produces competitive results as well. We attribute the improved performance of MC to the ability of mixtures to compensate for the restricted dependences modeled by CTBNs, and that of individual CTBNs to better fit the data with different weight sets. On the contrary, ECC and EPCC do not show consistent improvements over their base methods (CC and PCC, respectively) and sometimes even deteriorate the accuracy. This is due to the ad-hoc nature of their ensemble learning and prediction (see Section 2.1.7) that limits the potential improvement and disturbs the prediction of the ensemble classifiers.

Table 3.3 compares MC to other probabilistic MLC methods using CLL-loss. The results show that MC outperforms all other methods. This is expected because MC is tailored to optimize the conditional log-likelihood. Among the compared probabilistic methods, only PCC produces comparable results with MC because PCC explicitly evaluates all possible class assignments to compute the entire class conditional distribution. On the other hand, CC greedily seeks the mode of the class conditional distribution and results in large losses. In addition, CHF and MLKNN perform very poorly because they apply ad-hoc classification heuristics without performing proper probabilistic inference. Again, MC shows consistent improvements over SC because mixing multiple CTBNs allows us to account for different patterns in the data and, hence, improves the generalization of the model.

Lastly, Tables 3.4 and 3.5 show that MC is also very competitive in terms of micro and macro F1 scores, although optimizing them was not our immediate objective. One noteworthy observation is that ECC and EPCC do particularly well in terms of F1 scores. We consider averaging out the predictions on each class variable enhances BR-like characteristics in their ensemble decision. In the future, we will crossbreed these two different ensemble approaches (e.g., MCC/MPCC by applying our mixture framework and algorithms to CC/PCC; ECTBN using randomly structured CTBNs and simple averaging) and compare the performances.

3.3.7 Discussion

In this section, we presented a probabilistic ensemble approach to the MLC problem based on the Mixtures-of-Trees [Meilă and Jordan, 2000] and Conditional Tree-structured Bayesian Networks (CTBNs; Section 3.2) frameworks. We devised and presented algorithms for learning the parameters of the mixture, finding multiple tree structures and inferring the maximum a posteriori (MAP) output label configurations for the model.

Our experiments on a broad range of datasets revealed several interesting properties of the base CTBN model (SC) as well as the mixture-of-CTBNs (MC) method. First, the tree assumption of SC let us define efficient yet powerful learning and prediction algorithms, which result in the most competitive performance among the non-ensemble methods (*i.e.*, BR, CHF, MLKNN, IBLR, CC, and PCC). This suggests that SC can be a favorable candidate as a general off-the-shelf MLC solution, when one needs good predictive accuracy within a short period of time. On the other hand, as a mixture ensemble, MC considerably improved the accuracy of the outcomes over SC at the cost of longer parameter optimization (Section 3.3.3.1). We conclude that MC would be preferred when one needs more accurate multi-label prediction and probability estimates at the extra expense of time.

In the next section, we further improve this ensemble approach by developing a Mixtures-of-Experts [Jacobs et al., 1991] framework for MLC. We present a generalized representation of the class posterior distribution $P(\mathbf{Y}|\mathbf{X})$ that covers a number of existing MLC models [Boutell et al., 2004, Clare and King, 2001, Batal et al., 2013, Read et al., 2009] as special cases. We show how this generalized representation is incorporated with our extended Mixtures-of-Experts framework, which combines the decisions from multiple base MLC models using an input-dependent gate function (instead of a fixed set of mixture coefficients).

3.4 MULTI-LABEL MIXTURES-OF-EXPERTS

In this section, we develop a generalized probabilistic ensemble framework for the MLC problem that is based on the *Mixtures-of-Experts* [Jacobs et al., 1991] architecture. This novel

framework combines multiple MLC models in the *classifier chains family* (see Section 2.1) that decompose the class posterior distribution $P(Y_1, ..., Y_d|\mathbf{X})$ using a product of posterior distributions over components of the output space. Our approach captures different input–output and output–output relations that tend to change across data. As a result, we can recover a rich set of dependence relations among inputs and outputs that a single multi-label classification model cannot capture due to its modeling simplifications. We develop and present algorithms for learning the Mixtures-of-Experts models from data and for performing multi-label predictions on unseen data instances.

3.4.1 Preliminary: Mixtures-of-Experts Framework

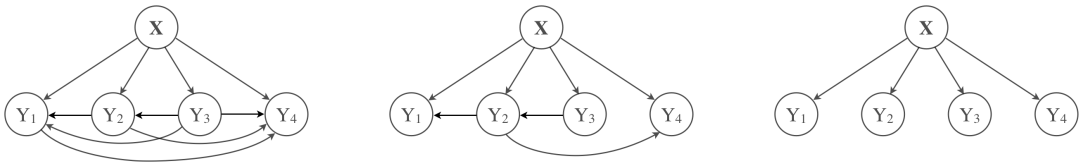
The MLC solution we propose combines multiple MLC classifiers using the *Mixtures-of-Experts* (ME) [Jacobs et al., 1991] architecture. While in general the ME architecture may combine many different types of probabilistic MLC models, this work focuses on the models that belong to the *classifier chains family* (CCF). In the following we briefly review the basics of ME and CCF.

The ME architecture is a mixture model that consists of a set of *experts* combined by a *gating function* (or *gate*). The model represents the conditional distribution $P(y|\mathbf{x})$ by the following decomposition:

$$\begin{aligned} P(y|\mathbf{x}) &= \sum_{k=1}^{K} P(E_k|\mathbf{x}) P(y|\mathbf{x}, E_k), \\ &= \sum_{k=1}^{K} g_k(\mathbf{x}) P(y|\mathbf{x}, E_k), \end{aligned} \tag{3.16}$$

where $P(y|\mathbf{x}, E_k)$ is the output distribution defined by the k-th expert E_k; and $P(E_k|\mathbf{x})$ is the context-sensitive prior of the k-th expert, which is implemented by the gating function $g_k(\mathbf{x})$. Generally speaking, depending on the choice of the expert model, ME can be used for either regression or classification [Yuksel et al., 2012].

Note that the gating function in ME defines a soft-partitioning of the input space, on which the K experts represent different input-output relations. The ability to switch among the experts in different input regions allows to compensate for the limitation of individual



(a) CC (b) CTBN (c) BR

Figure 3.5: Example models in the *classifier chains family*.

experts and improve the overall model accuracy. As a result, ME is especially useful when individual expert models are good in representing local input-output relations but may fail to accurately capture the relations on the complete input space.

ME has been successfully adopted in a wide range of applications, including handwriting recognition [Ebrahimpour et al., 2009], text classification [Estabrooks and Japkowicz, 2001], bioinformatics [Qi et al., 2007b, Cao et al., 2010], and climate prediction [Lu, 2006]. In addition, ME has been used in time series analysis, such as speech recognition [Mossavat et al., 2010], financial forecasting [Weigend and Shi, 2000] and dynamic control systems [Jacobs and Jordan, 1993, Weigend et al., 1995]. Recently, ME was used in social network analysis, in which various social behavior patterns are modeled through a mixture [Gormley and Murphy, 2011].

Here we apply the ME architecture to solve the MLC problem. In particular, we explore how to combine ME with MLC models that belong to the classifier chains family (CCF). The CCF models decompose the multivariate class posterior distribution $P(\mathbf{Y}|\mathbf{X})$ using a product of the posteriors over individual class variables as follows:

$$P(\mathbf{Y}|\mathbf{X}, M) = \prod_{i=1}^{d} P(Y_i|\mathbf{X}, \mathbf{Y}_{\boldsymbol{\pi}(i,M)}), \tag{3.17}$$

where $\mathbf{Y}_{\boldsymbol{\pi}(i,M)}$ denotes the parent classes of class variable Y_i defined by model M. An important advantage of the CCF models is that they give us a well-defined model of posterior class probabilities. That is, the models let us calculate $P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$ for any $(\mathbf{x}, \mathbf{y})$ input-output pair. This is extremely useful not only for prediction, but also for decision making

[Raiffa, 1997, Berger, 1985], conditional outlier analysis [Hauskrecht et al., 2007, Hauskrecht et al., 2010, Hauskrecht et al., 2013], or performing any inference over subsets of output class variables. In contrast, the majority of existing MLC methods aim to only identify the best output configuration for the given $\mathbf{x}$.

The original *classifier chains* (CC) model was introduced by Read et al. [Read et al., 2009]. Due to the efficiency and effectiveness of the model, CC has quickly gained large popularity in the machine learning community. Briefly, it defines the class posterior distribution $P(\mathbf{Y}|\mathbf{X})$ using a collection of classifiers that are tied together in a chain structure. To capture the dependence relations among features and class variables, CC allows each class variable to have only the classes that precede it along the chain as parents ($\mathbf{Y}_{\boldsymbol{\pi}(i,M)}$ in Equation (3.17)).

Figure 3.5(a) shows an example CC, whose chain order is $Y_3 \rightarrow Y_2 \rightarrow Y_1 \rightarrow Y_4$. Hence, the example defines the conditional joint distribution of class assignment (y_1, y_2, y_3, y_4) given $\mathbf{x}$ as:

$$
\begin{aligned}
&P(y_1, y_2, y_3, y_4|\mathbf{x}, M_{Fig.3.5(a)}) \\
&= P(y_3|\mathbf{x}) \cdot P(y_2|\mathbf{x}, y_3) \cdot P(y_1|\mathbf{x}, y_3, y_2) \cdot P(y_4|\mathbf{x}, y_3, y_2, y_1)
\end{aligned}
$$

Likewise, CCF is defined by a collection of classifiers, $P(Y_i|\mathbf{X}, \mathbf{Y}_{\boldsymbol{\pi}(i,M)}) : i = 1, ..., d$, one classifier for each output variable Y_i in the chain (Equation (3.17)). Theoretically, the CCF decomposition lets us accurately represent the complete conditional distribution $P(\mathbf{Y}|\mathbf{X})$ using a fully connected graph structure of $\mathbf{Y}$ (see Figure 3.5(a)). However, this property does not hold in practice [Dembczynski et al., 2010]. First, the choice of the univariate classifier model in CC (such as logistic regression), or other structural restrictions placed on the model, limit the types of multivariate output relations one can accurately represent. Second, the model is learned from data, and the data we have available for learning may be limited, which in turn may influence the model quality in some parts of the input space. As a result, a specific CC model is best viewed as an approximation of $P(\mathbf{Y}|\mathbf{X})$. In such a case, a more accurate approximation of $P(\mathbf{Y}|\mathbf{X})$ may be obtained by combining multiple CCs, each optimized for a different input subspace.

Conditional Tree-structured Bayesian networks (CTBN) (Section 3.2) is another model in CCF. The model is defined by an additional structural restriction: the number of parents is set to at most one (using the notation in Equation (3.17), $\mathbf{Y}_{\boldsymbol{\pi}(i,M)} := Y_{\pi(i,M)}$) and the dependence relations among classes form a tree:

$$P(\mathbf{y}|\mathbf{x}, M) = \prod_{i=1}^{d} P(y_i|\mathbf{x}, y_{\pi(i,M)}),$$

where $y_{\pi(i,M)}$ denotes the parent class of class Y_i in M. Figure 3.5(b) shows an example CTBN that defines: $P(\mathbf{y}|\mathbf{x}, M_{Fig.3.5(b)}) = P(y_3|\mathbf{x}) \cdot P(y_2|\mathbf{x}, y_3) \cdot P(y_1|\mathbf{x}, y_2) \cdot P(y_4|\mathbf{x}, y_2)$. The advantage of the tree-structured restriction is that it permits efficient structure learning and exact MAP inference [Batal et al., 2013].

The *binary relevance* (BR) [Clare and King, 2001, Boutell et al., 2004] model is a special case of CC that assumes all class variables are conditionally independent of each other ($\mathbf{Y}_{\boldsymbol{\pi}(i,M)} = \{\} : i = 1, ..., d$)[3]. Figure 3.5(c) illustrates BR when $d = 4$.

Finally, we would like to note that Section 3.3 extends the Mixtures-of-Trees framework [Meilă and Jordan, 2000, Šingliar and Hauskrecht, 2007] for multi-label prediction tasks using multiple CTBNs (Section 3.2). In this section, we further generalize the approach using ME and CCF.

3.4.2 Representation

By following the definition of ME (Equation (3.16)), ML-ME defines the multivariate posterior distribution of class vector $\mathbf{y} = (y_1, ..., y_d)$ by employing K CCF models described in the previous section:

$$P(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} g_k(\mathbf{x}) P(\mathbf{y}|\mathbf{x}, M_k) \tag{3.18}$$

$$= \sum_{k=1}^{K} g_k(\mathbf{x}) \prod_{i=1}^{d} P(y_i|\mathbf{x}, \mathbf{y}_{\boldsymbol{\pi}(i,M_k)}), \tag{3.19}$$

where $P(\mathbf{y}|\mathbf{x}, M_k) = \prod_{i=1}^{d} P(y_i|\mathbf{x}, \mathbf{y}_{\boldsymbol{\pi}(i,M_k)})$ is the joint conditional distribution defined by the k-th CCF model M_k and $g_k(\mathbf{x}) = P(M_k|\mathbf{x})$ is the gate reflecting how much M_k should

[3]By convention, $\mathbf{Y}_{\boldsymbol{\pi}(i,M)} = \{\}$ if Y_i in M does not have a parent class.

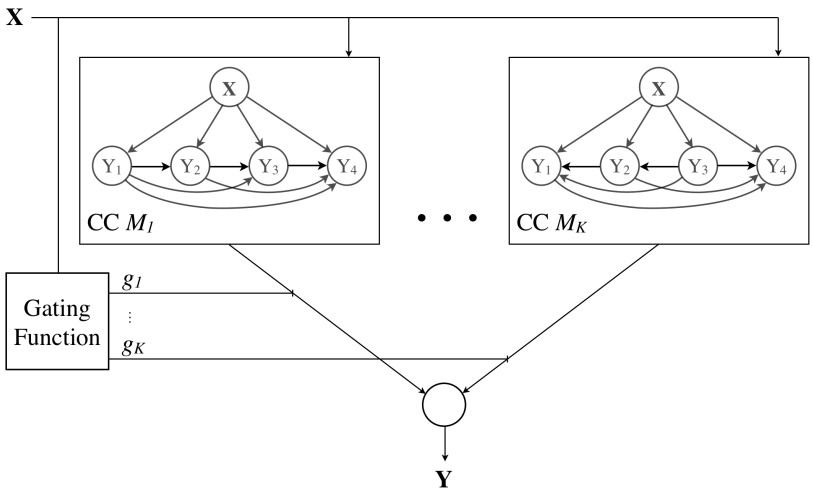


Figure 3.6: An example of ML-ME.

contribute towards predicting classes for input $\mathbf{x}$. We model the gate using the Softmax function, also known as normalized exponential:

$$g_k(\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_{G_k}\mathbf{x})}{\sum_{k'=1}^{K}\exp(\boldsymbol{\theta}_{G_{k'}}\mathbf{x})}, \tag{3.20}$$

where $\boldsymbol{\Theta}_G = \{\boldsymbol{\theta}_{G_k}\}_{k=1}^K$ is the set of Softmax parameters. Figure 3.6 illustrates an example of ML-ME model, which consists of K CCFs whose outputs are probabilistically combined by the gating function.

Parameters Let $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_G, \boldsymbol{\Theta}_M\}$ denote the set of parameters for an ML-ME model, where $\boldsymbol{\Theta}_G = \{\boldsymbol{\theta}_{G_k}\}_{k=1}^K$ are the gate parameters and $\boldsymbol{\Theta}_M = \{\boldsymbol{\theta}_{M_k}\}_{k=1}^K$ are the parameters of the CCF models defining individual experts. We define a gate output for each expert by a linear combination of inputs, which requires $|\boldsymbol{\theta}_{G_k}| = (m+1) = O(m)$ parameters. On the other hand, we parameterize each CCF expert by learning a set of classifiers. This in turn requires $|\boldsymbol{\theta}_{M_k}| = d(m + O(d) + 1) = O(dm + d^2)$ parameters.

In summary, the total number of parameters for our ML-ME model is $|\boldsymbol{\Theta}_G| + |\boldsymbol{\Theta}_M| = O(Kmd + Kd^2)$.

3.4.3 Parameter Learning

In this section, we describe how to learn the parameters of ML-ME when the structures of individual CCF models are known and fixed. We return to the structure learning problem in Section 3.4.4. Our objective here is to find the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_G, \boldsymbol{\Theta}_M\}$ that optimize the log-likelihood of the training data:

$$\begin{aligned} l(D; \boldsymbol{\Theta}) &= \sum_{n=1}^{N} \log P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}) \\ &= \sum_{n=1}^{N} \log \sum_{k=1}^{K} g_k(\mathbf{x}^{(n)}) P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, M_k) \end{aligned} \tag{3.21}$$

We refer to Equation (3.21) as the *observed log-likelihood*. However, direct optimization of this function is very difficult because the summation inside the log results in a non-convex function. To avoid this, we instead optimize the *complete log-likelihood*, which is defined by associating each instance $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ with a hidden variable $z^{(n)} \in \{1, ..., K\}$ indicating which expert it belongs to:

$$\begin{aligned} l_c(D; \boldsymbol{\Theta}) &= \sum_{n=1}^{N} \log P(\mathbf{y}^{(n)}, z^{(n)} | \mathbf{x}^{(n)}) \\ &= \sum_{n=1}^{N} \log \prod_{k=1}^{K} P(\mathbf{y}^{(n)}, M_k | \mathbf{x}^{(n)})^{\mathbb{1}[z^{(n)}=k]} \\ &= \sum_{n=1}^{N} \log \prod_{k=1}^{K} \left[g_k(\mathbf{x}^{(n)}) P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, M_k) \right]^{\mathbb{1}[z^{(n)}=k]} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{1}[z^{(n)} = k] \log \left(g_k(\mathbf{x}^{(n)}) P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}, M_k) \right), \end{aligned} \tag{3.22}$$

where $\mathbb{1}[z^{(n)} = k]$ is the indicator function that evaluates to one if the n-th instance belongs to the k-th expert and to zero otherwise. We use the EM framework that iteratively optimizes the *expected complete log-likelihood* ($E[l_c(D; \boldsymbol{\Theta})]$), which is always a lower bound of the observed log-likelihood [Dempster et al., 1977]. In the following, we derive an EM algorithm for ML-ME.

Each EM iteration consists of E-step and M-step. In the *E-step*, we compute the expectation of the complete log-likelihood. This reduces to computing the expectation of the hidden

variable $z^{(n)}$, which is equivalent to the posterior of the k-th expert given the observation and the current set of parameters.

$$\begin{aligned} E\left[\mathbb{1}[z^{(n)}=k]\right] &= P(z^{(n)}=k|\mathbf{y}^{(n)},\mathbf{x}^{(n)}) \\ &= \frac{g_k(\mathbf{x}^{(n)})P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)},M_k)}{\sum_{k'=1}^{K} g_{k'}(\mathbf{x}^{(n)})P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)},M_{k'})} \end{aligned} \tag{3.23}$$

In the *M-step*, we learn the model parameters $\{\boldsymbol{\Theta}_G, \boldsymbol{\Theta}_M\}$ that maximize the expected complete log-likelihood. Let $h_k^{(n)}$ denote $E\left[\mathbb{1}[z^{(n)}=k]\right]$. Then we can rewrite the expectation of Equation (3.22) using $h_k^{(n)}$ and by switching the order of summations:

$$\sum_{k=1}^{K}\sum_{n=1}^{N} h_k^{(n)} \log g_k(\mathbf{x}^{(n)}) + h_k^{(n)} \log P\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)},M_k\right)$$

As $h_k^{(n)}$ is fixed in the M-step, we can decompose this into two parts, which respectively involves the gate parameters $\boldsymbol{\Theta}_G$ and the CCF model parameters $\boldsymbol{\Theta}_M$:

$$\begin{aligned} f_G(D;\boldsymbol{\Theta}_G) &= \sum_{k=1}^{K}\sum_{n=1}^{N} h_k^{(n)} \log g_k(\mathbf{x}^{(n)}) \\ f_M(D;\boldsymbol{\Theta}_M) &= \sum_{k=1}^{K}\sum_{n=1}^{N} h_k^{(n)} \log P\left(\mathbf{y}^{(n)}|\mathbf{x}^{(n)},M_k\right) \end{aligned}$$

By taking advantage of this modular structure, we optimize $f_G(D;\boldsymbol{\Theta}_G)$ and $f_M(D;\boldsymbol{\Theta}_M)$ individually to learn $\boldsymbol{\Theta}_G$ and $\boldsymbol{\Theta}_M$, respectively. We first optimize $f_G(D;\boldsymbol{\Theta}_G)$, which we rewrite as (using Equation (3.20)):

$$\begin{aligned} &f_G(D;\boldsymbol{\Theta}_G) \\ &= \sum_{k=1}^{K}\sum_{n=1}^{N} h_k^{(n)} \boldsymbol{\theta}_{G_k}\mathbf{x}^{(n)} - h_k^{(n)} \log \sum_{k'=1}^{K} \exp(\boldsymbol{\theta}_{G_{k'}}\mathbf{x}^{(i)}) \end{aligned}$$

Since $f_G(D;\boldsymbol{\Theta}_G)$ is concave in $\boldsymbol{\Theta}_G$, we can find the optimal solution using a gradient-based method. The derivative of the log-likelihood with respect to $\boldsymbol{\theta}_{G_j}$ is:

$$\nabla_{\theta_j} f_G(D;\boldsymbol{\Theta}_G) = \sum_{n=1}^{N} \left\{ h_j^{(n)} - g_j(\mathbf{x}^{(n)}) \right\} \mathbf{x}^{(n)} \tag{3.24}$$

Note that this equation has an intuitive interpretation as the derivative becomes zero when $g_j(\mathbf{x}^{(n)}) = P(M_k|\mathbf{x}^{(n)})$ and $h_j^{(n)} = P(M_k|\mathbf{y}^{(n)},\mathbf{x}^{(n)})$ are equal.

Algorithm 4 Learn-ML-ME-parameters

Input: Training data D; base CCF experts $M_1, ..., M_K$
Output: Model parameters $\{\mathbf{\Theta}_G, \mathbf{\Theta}_T\}$

repeat
  E-step:
  for $k = 1$ **to** K, $n = 1$ **to** N **do**
    Compute $h_k^{(n)}$ using Equation (3.23)
  end for
  M-step:
  $\mathbf{\Theta}_G = \arg\max_{\mathbf{\Theta}_G} f_G(D; \mathbf{\Theta}_G) - R(\mathbf{\Theta}_G)$
  for $k = 1$ **to** K **do**
    $\boldsymbol{\theta}_{M_k} = \arg\max \sum_{n=1}^{N} h_k^{(n)} \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, M_k)$
  end for
until *convergence*

In our experiments, we solve this optimization using the L-BFGS algorithm [Liu and Nocedal, 1989], which is a quasi-Newton method that uses a sparse approximation to the inverse Hessian matrix to achieve a faster convergence rate even with a large number of variables. To prevent overfitting in high-dimensional space, we regularize with the L_2-norm of the parameters $R(\mathbf{\Theta}_G) = \frac{\lambda}{2} \sum_{k=1}^{K} ||\boldsymbol{\theta}_{G_k}||_2^2$.

Now we optimize $f_M(D; \mathbf{\Theta}_M)$, which can be further broken down into learning K individual CCF models. Note that f_M forms the weighted log-likelihood where $h_k^{(n)}$ serves as the instance weight. In our experiments, we optimize this by applying L_2-regularized instance-weighted logistic regression models.

3.4.3.1 Complexity Algorithm 4 summarizes our parameter learning algorithm. The E-step computes $h_k^{(n)}$ for each instance on each expert. This requires $O(md)$ multiplications. Hence, the complexity of a single E-step is $O(KNmd)$. The M-step optimizes the parameters $\mathbf{\Theta}_G$ and $\mathbf{\Theta}_M$. Optimizing $\mathbf{\Theta}_G$ computes the derivative (Equation (3.24)) which requires $O(mN)$ multiplications. Denoting the number of L-BFGS steps by l, this requires $O(mNl)$ operations. Optimizing $\mathbf{\Theta}_M$ learns K CCF models. We do this by learning $O(Kd)$ instance-weight logistic regression models.

3.4.4 Structure Learning

We previously described the parameter learning of ML-ME by assuming we have fixed the individual structures. In this section, we present how to obtain useful structures for learning a mixture from data. We first show how to obtain CCF structures from weighted data. Then, we present our sequential boosting-like heuristic that, on each iteration, learns a structure by focusing on "hard" instances that previous mixture tends to misclassify.

Learning a Single CCF Structure on Weighted Data To learn the structure that best approximates weighted data, we find the structure that maximizes the weighted conditional log-likelihood (WCLL) on $\{D, \Omega\}$, where $\Omega = \{\omega^{(n)}\}_{n=1}^{N}$ is the instance weight. Note that we further split D into training data D_{tr} and hold-out data D_h for internal validation.

Given a CCF structure M, we train its parameters using D_{tr}, which corresponds to learning instance-weighted logistic regression using D_{tr} and their weights. On the other hand, we use WCLL of D_h to define the score that measures the quality of M.

$$score(M) = \sum_{n \in D_h} \omega^{(n)} \sum_{i=1}^{d} \log P(y_i^{(n)} | \mathbf{x}^{(n)}, \mathbf{y}_{\boldsymbol{\pi}(i,M)}^{(n)}) \tag{3.25}$$

The original CC [Read et al., 2009] generates the underlying dependence structure (chain order) by a random permutation. In theory, this would not affect the model accuracy as CC still considers the complete relations among the class variables. However, in practice, using a randomly generated structure may degrade the model performance due to the modeling and algorithmic simplifications (see section 3.4.1). In order to alleviate the issue, Read et al. [Read et al., 2009] suggested to use Ensembles of CC (ECC) that averages the predictions of multiple randomly ordered CCs trained on random subsets of the data. However, this is not a viable option because simply averaging the multi-dimensional output predictions may result in inconsistent estimates (does not correctly solve Equation (3.1)).

Instead, we use a structure learning algorithm that learns a chain order greedily by maximizing WCLL. That is, starting from an empty ordered set ρ, we iteratively add a class

Algorithm 5 Find-an-optimal-chain-structure

Input: Training data D
Output: Chain order ρ

1: Split D into D_{tr} and D_h
2: Initialize an ordered set $\rho = \{\}$
3: **for** $i = 1$ **to** d **and** $j \notin \rho$ **do**
4: **for** $j = 1$ **to** d **do**
5: $\boldsymbol{\theta}_j = \arg\max_{\boldsymbol{\theta}_j} \omega^{(n)} \log P(y_j^{(n)} | \mathbf{x}^{(n)}, \mathbf{y}_\rho^{(n)}) : n \in D_{tr}$
6: **end for**
7: $\rho = \rho \cup \arg\max_j \omega^{(n)} \log P(y_j^{(n)} | \mathbf{x}^{(n)}, \mathbf{y}_\rho^{(n)}; \boldsymbol{\theta}_j) : n \in D_h$
8: **end for**

index j to ρ by optimizing:

$$score_j(\rho) = \sum_{n \in D_h} \omega^{(n)} \log P(y_j^{(n)} | \mathbf{x}^{(n)}, \mathbf{y}_{i \in \rho}^{(n)}), \tag{3.26}$$

where $\mathbf{y}_{i \in \rho}^{(n)}$ denotes the classes previously selected in ρ. We formalize our method in Algorithm 5. Note that this algorithm can be seen as a special case of [Kumar et al., 2012] that optimizes the chain order using the beam search.

We would like to note that by incorporating additional restriction on the CC model, the optimal (restricted) CC structure may be efficiently computable. An example of such a model is the Conditional Tree-structured Bayesian Network (CTBN; Section 3.2). Briefly, the optimal CTBN structure may be found using the maximum branch (weighted maximum spanning tree) [Edmonds, 1967] out of a weighted complete digraph, whose vertices represent class variables and the edges between them represent pairwise dependences between classes.

Learning Multiple CCF Structures To obtain multiple, effective CCF structures for ML-ME, we apply the above described algorithms multiple times with different sets of instance weights. This section explains how we assign the weights such that poorly predicted instances have higher weights; and well-predicted instances have lower weights.

To start with, we assign all instances uniform weights ($\omega^{(n)} = 1/N : n = 1, ..., N$; i.e., all instances are equally important a priori). Using this initial set of weights, we first obtain a CCF structure ρ_1 (i.e., either a CC or CTBN structure) and train a model M_1 that follows

ρ_1. Then, by setting the current mixture $\boldsymbol{\mathcal{M}}$ to be M_1, we compute the new instance weights to be the normalized prediction error:

$$\omega^{(n)} \propto 1 - P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, \boldsymbol{\mathcal{M}}), \;\; s.t. \;\; \sum_{n=1}^{N} \omega^{(n)} = 1$$

With the updated weights $\{\omega^{(n)}\}$, we obtain another structure ρ_2, and train $\boldsymbol{\mathcal{M}}$ with M_1 and M_2 that follow ρ_1 and ρ_2, respectively (Algorithm 4).

We incrementally inject new models to the mixture by repeating this process. To stop the process, we use internal validation approach. Specifically, the data used for learning are split into internal train and test sets. The structure of the trees and parameters are always learned on the internal train set. The quality of the current mixture is evaluated on the internal test set. The mixture growth stops when the log-likelihood on the internal test set for the new mixture does not improve any more. The structures included in the previous mixture are then fixed, and the parameters of the mixture are re-learned on the full training data.

3.4.4.1 Complexity To learn a single CCF using our greedy algorithm, we need to estimate $P(Y_i|\mathbf{X}, \mathbf{Y}_{\pi(i,M)})$ for $O(d^2)$ times. Since we learn K CCF structures for a mixture, the overall complexity is $O(Kd^2)$ times the complexity of learning logistic regression.

3.4.5 Prediction

In order to make a prediction for a new instance $\mathbf{x}$, we want to find the MAP assignment of the class variables (see Equation (3.1)). Our ML-ME model consists of multiple CCF models and the MAP solution may, at the end, require enumeration of exponentially many class assignments. To address this problem, we rely on approximate MAP inference. The two commonly applied MAP approximation approaches in the literature are: convex programming relaxation via dual decomposition [Sontag, 2010], and simulated annealing using a Markov chain [Yuan et al., 2004]. Here we use the latter approach. Briefly, we search the space of all assignments by defining a Markov chain that is induced by local changes to individual class labels. The annealed version of the exploration procedure [Yuan et al.,

2004] is then used to speed up the search. We initialize our MAP algorithm using the following heuristic: first, we identify the MAP assignments for each CCF model in the mixture individually [Dembczynski et al., 2010, Batal et al., 2013, Boutell et al., 2004]. After that, we pick the best assignment among these candidates. We have found this heuristic to work very well and often results in the true MAP assignment.

3.4.6 Experiments

3.4.6.1 Datasets We use seven publicly available MLC datasets obtained from different domains. Table 3.6 summarizes the characteristics of the datasets, including dataset size, label cardinality (the average number of labels per instance), distinct label set (the number of distinct class configurations that appear in the data) and data domain.

3.4.6.2 Methods To demonstrate the benefits of our mixture framework, we compare the performance of the following eight methods: *binary relevance (BR)* [Clare and King, 2001, Boutell et al., 2004], *conditional tree-structured Bayesian networks (CTBN)* [Batal et al., 2013], *classifier chains (CC) and their ensembles (ECC)* [Read et al., 2009], *probabilistic classifier chains (PCC) and their ensembles (EPCC)* [Dembczynski et al., 2010], *ML-ME with CTBN (MCTBN)* and *ML-ME with CC (MCC)*.

For a fair comparison of the methods, we fix the following parameters throughout all

DATASET	N	m	d	LC	DLS	DM
Image	2,000	135	5	1.24	20	image
Scene	2,407	294	6	1.07	15	image
Emotions	593	72	6	1.87	27	music
Flags	194	19	7	3.39	54	image
Yeast	2,417	103	14	4.24	198	biology
Medical	978	1,449	45	1.25	94	text
Enron	1,702	1,001	53	3.38	753	text

Table 3.6: Datasets characteristics (N: number of instances, m: number of features, d: number of classes, LC: label cardinality, DLS: distinct label set, DM: domain).

experiments:

• We use L_2-penalized logistic regression for all of the methods and choose their regularization parameters by cross validation.

• We set the maximum number of experts to 10 for MCTBN/MCC. We use our heuristic (section 3.4.4) to stop early if possible; ECC/EPCC use 10 fixed number of base models in an ensemble.

• We use our structure learning algorithm (Algorithm 5) for CC/PCC; we use random chain orders for ECC/EPCC.

• For predictions on MCTBN/MCC, we use 150 iterations of simulated annealing.

3.4.6.3 Evaluation Metrics To compare different MLC methods, we use *Exact match accuracy* (EMA) and *Conditional log-likelihood loss* (CLL-loss) (see Section 3.3.6.3 for detailed description).

3.4.6.4 Results Tables 3.7 and 3.8 show the performance of all methods in terms of EMA and CLL-loss, respectively. All results are obtained using *ten-fold cross validation*. In parentheses, we indicate the relative ranking of the methods on each dataset. We do not report the results of PCC/EPCC on Medical and Enron because evaluating all $O(2^d)$ class assignments is clearly infeasible. Also, we do not report CLL-loss for ECC and EPCC because they do not produce probabilistic output.

Based on the results, our ML-ME framework clearly improves the performance of the base models. In terms of EMA (Table 3.7), the prediction accuracy of MCC is not only the highest but also the most stable. Although not as good as MCC, MCTBN also shows a large improvement compared with CTBN. The results demonstrate that ML-ME compensates for the restrictions that the base MLC models have using their combinations. In addition, this is in contrast to simple averaging, which often leads to inconsistent estimation (ECC and EPCC). The model fitness of MCC measured by CLL-loss (Table 3.8) also indicates that MCC is competitive, followed by MCTBN, CTBN, BR and CC. Although PCC is recording the highest average ranking, it is computationally very expensive and does not scale up to large data.

In summary, the experimental results show that our ML-ME method with the CCF

EMA	BR	CTBN	CC	PCC	ECC	EPCC	MCTBN	MCC
Image	0.279±0.036 (8)	0.407±0.036 (6)	0.445±0.038 (2)	0.452±0.032 (2)	0.413±0.028 (6)	0.442±0.019 (2)	0.444±0.035 (2)	**0.486±0.038** (1)
Scene	0.542±0.028 (8)	0.624±0.035 (7)	0.694±0.023 (3)	**0.701±0.028** (1)	0.658±0.027 (5)	0.681±0.030 (3)	0.645±0.028 (5)	**0.708±0.028** (1)
Emotions	0.265±0.056 (8)	0.334±0.065 (4)	0.341±0.061 (4)	0.343±0.073 (4)	0.288±0.086 (4)	**0.344±0.072** (1)	**0.370±0.063** (1)	**0.356±0.062** (1)
Flags	0.139±0.042 (7)	0.155±0.069 (7)	**0.196±0.067** (1)	0.191±0.075 (6)	**0.212±0.089** (1)	**0.222±0.062** (1)	**0.216±0.063** (1)	**0.227±0.071** (1)
Yeast	0.151±0.024 (8)	0.195±0.026 (7)	0.220±0.027 (3)	0.242±0.023 (2)	0.204±0.024 (3)	0.219±0.015 (3)	0.218±0.025 (3)	**0.259±0.026** (1)
Medical	0.641±0.075 (6)	0.667±0.079 (4)	0.688±0.056 (4)	- (-)	**0.701±0.035** (1)	- (-)	**0.712±0.065** (1)	**0.711±0.055** (1)
Enron	0.173±0.024 (4)	0.184±0.016 (4)	**0.197±0.032** (1)	- (-)	0.181±0.030 (4)	- (-)	**0.196±0.023** (1)	**0.192±0.026** (1)
Avg.Rank	7.0	5.6	2.6	3.0	3.4	2.0	2.0	1.0

Table 3.7: Performance of each method on the benchmark datasets in terms of exact match accuracy (EMA; higher value is better). Numbers in parentheses show the relative ranking of the method on each dataset. The best methods (by paired t-test at $\alpha = 0.05$) are shown in **bold**. The last row shows the average ranking of the methods.

CLL-loss	BR	CTBN	CC	PCC	MCTBN	MCC
Image	432.6±20.8 (5)	391.0±22.2 (4)	475.9±34.9 (6)	**347.0±24.4** (1)	378.3±20.7 (3)	**342.6±30.7** (1)
Scene	343.6±22.5 (5)	287.2±16.4 (4)	371.8±32.3 (6)	**230.1±15.6** (1)	277.8±12.0 (3)	**234.2±18.4** (1)
Emotions	153.9±10.7 (5)	135.8±8.6 (4)	155.2±10.1 (5)	**130.1±10.1** (1)	134.2±9.9 (3)	**132.0±10.0** (1)
Flags	68.6±11.5 (2)	66.5±11.2 (2)	80.9±19.7 (6)	**57.6±11.4** (1)	66.6±12.1 (2)	66.3±9.1 (2)
Yeast	1502.4±45.1 (5)	1075.3±46.5 (3)	2233.4±126.4 (6)	932.1±72.6 (2)	1077.5±52.7 (3)	**915.7±38.6** (1)
Medical	155.9±25.2 (4)	145.4±23.8 (2)	152.7±35.6 (4)	- (-)	**133.3±34.8** (1)	140.6±36.6 (2)
Enron	1441.3±85.7 (5)	1316.4±78.6 (4)	1230.9±72.4 (3)	- (-)	**1127.4±63.8** (1)	1156.2±71.3 (2)
Avg.Rank	4.4	3.3	5.1	1.2	2.3	1.4

Table 3.8: Performance of each method on the benchmark datasets in terms of conditional log-likelihood loss (CLL-loss; smaller value is better). Numbers in parentheses show the relative ranking of the method on each dataset. The best methods (by paired t-test at $\alpha = 0.05$) are shown in **bold**. The last row shows the average ranking of the methods.

experts is able to outperform or match the existing state-of-the-art methods across a broad range of benchmark MLC datasets. We attribute this improvement to the ability of the CCF mixture that simultaneously compensates for the restricted dependencies modeled by an individual CCF, and to its ability that better fits the different regions of the input space with new expert models.

3.4.7 Discussion

We presented a novel probabilistic ensemble framework for multi-label classification. Our approach attempts to capture different input-output and output-output relations that tend to change across data. We integrated the Mixtures-of-Experts architecture and the multi-label classification models in the classifier chains family, which decompose the class posterior distribution $P(Y_1, ..., Y_d|\mathbf{X})$ using a product of posterior distributions over components of the output space. We developed the learning and prediction algorithms for our mixture framework, and showed that our approach recovers a rich set of dependence relations among inputs and outputs that a single multi-label classification model cannot capture due to its modeling simplifications.

Through the experiments on multiple benchmark datasets, we found that our generalized mixture approach achieves highly competitive results and outperforms the existing state-of-the-art multi-label classification methods. We conclude that our mixture solutions would be useful when one prefers superior predictive accuracy to a manageable amount of additional learning and prediction time. The results also showed that our solutions can be applied when precise probability estimates are demanded. Unlike our approach, other existing solutions revealed some limitations. For example, although PCC and EPCC showed competitive prediction performance, they could not finish on some datasets due to the exhaustive search involved in their prediction algorithms. Also, because of the the simplicity of the ensemble prediction algorithm used in ECC and EPCC, the methods are only able to output binary prediction and are incapable of producing probability estimates.

Lastly, note that MC (Section 3.3) is a special case of ML-ME where the base MLC method is CTBN and the gating function is fixed to return $g_k(\mathbf{x}) = 1/K$ for all $k = 1, ..., K$.

As a result, while the parameters of an MC can be optimized faster than that of an ML-ME, the performance of ML-ME is generally superior or similar to that of MC.

3.5 SUMMARY

We studied the multi-label classification problem, where our goal is to predict the maximum a posteriori (MAP) output $\mathbf{y}$ for a given input $\mathbf{x}$.

First, we developed *Conditional Tree-structured Bayesian Networks* (CTBN) that restricts the dependence relations among the response variables to follow a *directed tree*. Our model represents the conditional dependence relations between classes using a special tree-structured Bayesian network, whose conditional distributions are defined using probabilistic classifiers. We presented an efficient algorithm to learn the tree structure that maximizes the conditional log likelihood. We provided an efficient exact inference algorithm that has a linear-time complexity in the number of class variables.

Next, we presented *Mixtures-of-Conditional Tree-structured Bayesian Networks* (MC) that builds a mixture ensemble of multiple tree-structured models (CTBNs) to better represent the dependence relations among the response variables. We devised algorithms for learning the parameters of the mixture, finding multiple tree structures, and inferring the MAP output label configurations for a given input. Consequently, we developed a new mixture framework that can learn various dependence relations, which a single tree-structured model cannot capture, and combine them to make ensemble predictions that achieve a higher predictive accuracy.

Last, we developed the *Multi-Label Mixtures-of-Experts* (ML-ME) framework that combines MLC models in the *classifier chains family* — our generalization of structured MLC models that decompose the class posterior distribution $P(Y_1, ..., Y_d|\mathbf{X})$ using a product of posterior distributions over components of the output space. We overviewed the classifier chains family and the Mixtures-of-Experts [Jacobs et al., 1991] framework. We presented algorithms for learning the ML-ME models from data and for performing multi-label predictions on unseen data instances.

Our experimental evaluation on a range of datasets showed that our approaches outperform several state-of-the-art methods and produce much more reliable probabilistic estimates.

4.0 CONDITIONAL OUTLIER DETECTION

This chapter explores and develops solutions for the *conditional outlier detection* (COD) problem that aims to identify data instances with unusual input-output associations. More specifically, we seek data instances with unusual output given its input (or context). We assume the dataset in which we search for outliers is formed by input-output pairs; hence, both the input and output attributes are given a priori.

In terms of conditional outlier detection methodology, we investigate two main directions, (1) probabilistic methods and (2) outlier scoring methods derived from unconditional outlier methods:

Probabilistic methods The idea behind probabilistic methods is to first build a model of $P(\mathbf{Y}|\mathbf{X})$ from data, and then use it to identify conditional outliers. Briefly, conditional outliers are data instances with a low probability $P(\mathbf{Y}|\mathbf{X})$. We note that the meaning of 'low probability' should not be interpreted in absolute terms (absolute probability values) but in relative terms – that is, relative to probabilities associated with other outcomes. For example, assuming a binary case, the probability of 0.1 for an outcome is low relative to the opposite outcome, which is 0.9. However, 0.1 may not be low if there are many different outcomes. For example, when there are 10 outcomes and three of these are assigned probability 0.02, 0.1 should not be considered low.

We note that there are many different ways of defining and learning $P(\mathbf{Y}|\mathbf{X})$ models from data. Regardless of the approach used, the key here is to assure that the model of $P(\mathbf{Y}|\mathbf{X})$ is as accurate as possible, so that low probabilities are as precise as possible. This is especially challenging when the model is learned from a limited finite-size data.

Outlier scoring methods derived from unconditional outlier methods As seen in Chapter 2, there has been a tremendous amount of work in outlier detection research in recent

years. The effort of the community has focused primarily on unconditional outlier detection methods that seek unusual instances in data where all attributes are treated equally; that is, there is no division of attributes between inputs and outputs. A large spectrum of different unconditional methods have been developed. However, an important open issue is how one can benefit from these methods to support conditional outlier detection. To address this, we develop and explore a new general framework for defining conditional outlier scores in terms of unconditional outlier scores. The new framework bridges the gap in the development of conditional and unconditional outlier methods.

Throughout this chapter, we consider two types of conditional outlier detection (COD) problems: *univariate* and *multivariate*. The two problems differ in the dimensionality of the output. In the univariate COD (UCOD) problem, each input vector is associated with a single output variable, whereas in the multivariate COD (MCOD) problem, each input vector is associated with multiple output variables. In other words, UCOD is a special case of the MCOD problem where the output dimensionality is one. For the sake of simplicity, we will first tackle the UCOD problem to build our outlier detection solutions (Section 4.2). After that, we will explore and develop solutions for the MCOD problem (Section 4.3) which is a much harder version of the conditional outlier detection problem especially when dealing with a high-dimensional output space. We show how one can tackle this problem by developing outlier detection approaches based on the MLC-like decompositions (see previous chapter). We develop a new multivariate conditional outlier framework that utilizes the decomposable structure of the MLC models to acquire a set of projections representing the dimension-wise conditional outlier scores. These scores are then combined to produce the final multivariate conditional outlier score.

The rest of the chapter is organized as follows. Section 4.1 reviews the formal definition of the COD problem. Section 4.2 presents solutions for the UCOD problem. Section 4.3 extends the univariate approaches to the MCOD problem. The experimental results that demonstrate the validity and effectiveness of our approaches to successfully identify conditional outliers are presented at the end of each section.

4.1 PROBLEM DEFINITION AND NOTATION

The *conditional outlier detection* (COD) problem is a special type of the outlier detection problem, where we are interested in finding data instances that show unusual (or irregular) output patterns given their input (or context).

More formally, let us assume we have a dataset $D = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$ that consists of N data instances, such that each instance consists of an m-dimensional input vector (or context) $\mathbf{x}^{(n)} = (x_1^{(n)}, ..., x_m^{(n)})$ and corresponding d-dimensional binary output vector (or response) $\mathbf{y}^{(n)} = (y_1^{(n)}, ..., y_d^{(n)})$. We seek to identify an instance $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ for which the output $\mathbf{y}^{(n)}$ is unusual for the given $\mathbf{x}^{(n)}$ when compared to the rest of data instances in D. Example problems are the identification of images with unusual annotation given the image content, or an order of a medication that is unusual given the patient condition.

Please note that the above definition of the COD problem considers the multivariate output space. We referred to this type of problem as the *multivariate COD* (MCOD) problem. The *univariate COD* (UCOD) problem is a special case of the MCOD problem with $d = 1$.

In this chapter, we use the following notations:

- D — Dataset
- $\mathbf{X}$, $\mathbf{x}$ — Input (context) variable and value
- $\mathbf{Y}$, $\mathbf{y}$ — Output (response) variable and value
- m — Input dimensionality
- d — Output dimensionality
- N — Number of data instances
- n — Index of data instance
- $\mathcal{M}$ — A trained data model
- $\boldsymbol{\rho} = \{\rho_1, ..., \rho_d\}$ — Conditional probability estimates on individual output dimensions ($i.e.$, $\rho_i = P(y_i|\mathbf{x}, \boldsymbol{\pi}(y_i); \mathcal{M})$)
- $f(\bullet)$, θ_f — A discriminative function and its parameters (Sections 4.2.2 and 4.3.2)
- ϕ — A discriminative projection of a data instance (Sections 4.2.2 and 4.3.2)
- $o_U(\bullet)$ — An unconditional outlier scoring function (Sections 4.2.2 and 4.3.2)

4.2 UNIVARIATE CONDITIONAL OUTLIER DETECTION

This section explores the solutions for the univariate conditional outlier detection (UCOD) problem where the dataset analyzed contains only one output variable. We focus our attention on two approaches for scoring and ranking outliers explored throughout this thesis: the probabilistic scoring approach and the outlier scoring approach where scores are derived from unconditional outlier scores.

The objective of this section is to present the key ideas behind the two approaches on a simpler (univariate) conditional outlier detection problem. The univariate solutions will also develop the basic building blocks of our more general solutions for the MCOD problem (Section 4.3).

4.2.1 Probabilistic Approach to Univariate Conditional Outlier Detection

The basic idea of the probabilistic approach is to build a data model of conditional probability $P(Y|\mathbf{X})$ from data D. The model is then used to identify outliers by calculating $\widetilde{P}(Y = y^{(n)}|\mathbf{X} = \mathbf{x}^{(n)})$, where $(\mathbf{x}^{(n)}, y^{(n)})$ denotes a data instance being examined[1]. In general, the instance is an outlier when it leads to a low conditional probability. Since it may be hard to define a fixed "low probability" threshold, the conditional probability can be also used to score and rank the different data instances in terms of their outlier strength. We note that the probabilistic approach has been successfully applied to solve multiple UCOD problems in the literature [Hauskrecht et al., 2007, Song et al., 2007, Hauskrecht et al., 2010, Hauskrecht et al., 2016].

Figure 4.1 illustrates the basics of the probabilistic approach and its two phases: *data modeling* and *outlier scoring*. In the following, we review methods one can use for building the data model and discuss the outlier scoring step.

4.2.1.1 Data Modeling The first phase of the probabilitic approach regards model building that produces a probabilistic model $\mathcal{M}$ that captures stochastic dependence rela-

[1]The vertical bar (|) denotes conditioning; variables to the left of the symbol are conditioned on those to the right of the symbol.

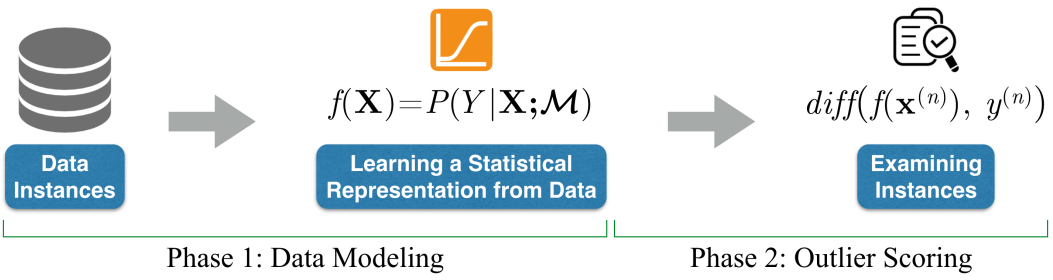


Figure 4.1: Probabilistic conditional outlier detection.

tions among data attributes. The goal of this phase is to obtain a precise data representation that can efficiently estimate the conditional probability $\widetilde{P}(y|\mathbf{x};\mathcal{M})$ for any observed input and output pair $(\mathbf{x}, y)$. Various statistical machine learning models and methods can be used for this purpose. These include generative and discriminative models.

Generative models compute the conditional probability using Bayes rule (*i.e.*, $P(Y|\mathbf{X}) = P(\mathbf{X}, Y)/P(\mathbf{X})$). That is, approaches based on generative models first learn the joint distribution of both input and output, $P(\mathbf{X}, Y)$, using a set of model parameters. Subsequently, the joint distribution is used to estimate the conditional distribution through an algebraic transformation defined by Bayes rule. This approach was applied to COD by [Hauskrecht et al., 2007] to detect unusual emergency room admissions from emergency room observations and findings. The authors built a probabilistic model of the admission action conditioned on the current patient status (such as symptoms, observations) using Bayesian belief networks (BBN) [Pearl, 1988, Lauritzen and Spiegelhalter, 1988, Cooper and Herskovits, 1992]. A similar approach for COD was also used in [Song et al., 2007]. This work tackled a slightly different type of COD problem where both input and output attributes are continuous. The generative model used in the work was based on the Gaussian mixture model (GMM) [Nowlan, 1991, Titterington et al., 1985] which was used to represent the conditional probability by modeling the correlations among the input and output spaces respectively.

In contrast to generative models, *discriminative* models directly learn the conditional distribution $P(Y|\mathbf{X})$ by optimizing a likelihood or loss function expressed by a set of parameters. The discriminative models were used to support COD for identification of unusual

patient management actions (medication and lab orders) in clinical workflow [Hauskrecht et al., 2010, Hauskrecht et al., 2013, Hauskrecht et al., 2016]. More specifically, the approach applied in this work used calibrated support vector machines (SVM) models that first learn a discriminative projection of the input attributes that reflect the associated output values. Then, a transformation from the projection to a probability estimate is obtained using a post-hoc recalibration approach [Platt, 1999, DeGroot and Fienberg, 1983].

Apparently, generative and discriminative models have very different properties as well as complementary strengths and weaknesses (*e.g.*, generative models allow one to generate new data similar to existing data; whereas discriminative models generally outperform generative models in classification tasks). Detailed discussion on the comparison of generative and discriminative models could be found in [Ng and Jordan, 2002, Ulusoy and Bishop, 2006, Bishop and Lasserre, 2007].

To represent the probabilistic approach in our empirical studies, we implement the baseline probabilistic model using the discriminative approach, where the L_2-regularized logistic regression model is used to directly learn the conditional probability from D.

4.2.1.2 Outlier Scoring The second phase of the probabilistic approach aims to compute outlier scores using the obtained data model. The goal is to assign each instance an outlier score such that the higher the score is, the more likely the instance is an outlier.

Since outliers are associated with low probabilities, we can convert probabilities to outlier scores (where stronger outliers are associated with a higher score) using one of the following transformations:

$$Score_{\text{PROB}}(y^{(n)}|\mathbf{x}^{(n)}) = 1 - \widetilde{P}(y^{(n)}|\mathbf{x}^{(n)};\mathcal{M}) \tag{4.1}$$

or

$$Score_{\text{PROB}}(y^{(n)}|\mathbf{x}^{(n)}) = \frac{1}{\widetilde{P}(y^{(n)}|\mathbf{x}^{(n)};\mathcal{M})} \tag{4.2}$$

In the following discussion, we use a minor modification of the second score, where we take the logarithm of the inverse probability to rank the conditional outliers:

$$Score_{\text{PROB}}(y^{(n)}|\mathbf{x}^{(n)}) = -\log \widetilde{P}(y^{(n)}|\mathbf{x}^{(n)};\mathcal{M}) \tag{4.3}$$

Note that the logarithm is a monotonous function. Therefore, the order of scores before and after the transformation is preserved.

4.2.1.3 Limitations of Probabilistic Models

Probabilistic outlier detection approaches, however, has several fundamental drawbacks that may affect the COD performance. This mainly regards the accuracy of the underlying data models that produce the probability estimates, with which we compute the outlier score.

More specifically, standard parameter optimization criteria for generative models, such as the Bayesian belief networks, naïve Bayes model, and linear discriminant analysis, assume that data instances are drawn independently from an unknown population (*i.e.*, independently and identically distributed or i.i.d.). Accordingly, they treat all instances equally important and minimize the expected loss under the i.i.d. assumption. However, this assumption is often violated in many practical problems [Dundar et al., 2007].

Although discriminative models, such as logistic regression, are less strict with the i.i.d. assumption, the models still often fail to produce well calibrated probabilities for sparse regions of the input ($\mathbf{X}$) space (*i.e.*, the regions where $\mathbf{X}$ has a low support) [MacKay, 2003]. In addition to that, the fixed representation of parametric data models may constrain accuracy in the estimates. A parametric approach relies on a set of model parameters that reflect the underlying assumptions about the population. When the assumptions are correct, the approach will produce accurate and precise probability estimates. However, if the assumptions are not correct, the approach has a large chance of failing; *e.g.*, when one trains a linear model for nonlinear domains, the assumption that the probability is monotonously increasing along the discriminative projection does not hold and leads to imprecise probability estimation.

Apparently, the above described issue may have a crucial impact on the outlier detection performance. Unfortunately, there are no rules of thumb for avoiding the issue. For example, it is possible to consider local (instance-based) models instead of building a global model, such as the work by [Valko and Hauskrecht, 2008]. However, this approach typically reduces the sample size, and thus the resulting probability estimates may still be inaccurate. Alternatively, calibration via binning [Tukey, 1961, Bella et al., 2009, Pakdaman, 2017] might

address the general issues with imprecise probability estimates. However, this again would not be a good solution for outlier detection, in which we want to correctly estimate very small probabilities. Since outlier detection is often done on a finite dataset, this becomes particularly hard as binning reduces sample size.

4.2.2 Univariate Conditional Outlier Detection with Unconditional Outlier Detection Methods

Now we switch our focus to conditional outlier detection methods that do not require probability estimation. This switch is partly driven by the fact that a large spectrum of successful outlier detection models, which were developed by the machine learning and data mining communities for unconditional outlier detection, are non-probabilistic. In addition, the existence of many unconditional outlier detection methods (see Section 2.2) raises another important question: *Is it possible to take advantage of these methods when defining the conditional outliers and when building the conditional outlier detection methods?* The current state-of-the-art conditional outlier detection does not take much advantage of the progress and solutions developed in unconditional approaches. To bridge this gap we propose, develop and test a new conditional outlier detection framework that defines the conditional outlier score by combining the results of multiple unconditional outlier scores.

4.2.2.1 Ratio of Outlier Scores Briefly, our new conditional outlier score for a data instance works by comparing (via ratio) two unconditional outlier scores: one score calculated against data instances with the same observed output value; and another calculated against instances with the opposite output value. We refer to the new conditional outlier scoring approach as Ratio of Outlier Scores (or Ratio-based Outlier Scoring; ROS) approach. It comes with a couple of important advantages. First, it allows us to utilize a wide variety of unconditional outlier scores. Also, it lets us effectively avoid the cases where instances with rare $\mathbf{x}$ (but properly associated with $\mathbf{y}$) undesirably receive a high conditional outlier score.

More formally, let us consider a binary-labeled dataset $D = \{\mathbf{x}^{(n)}, y^{(n)}\}_{n=1}^{N}$ where each instance in D consists of a continuous input vector $\mathbf{x}^{(n)} \in \mathbb{R}^m$ and an associated output value

$y^{(n)} = \{0, 1\}$. For notational convenience, let us also define $D_{\text{Agree}(n)}$ and $D_{\text{Disagree}(n)}$, subsets of D based on the value of $y^{(n)}$:

$D_{\text{Agree}(n)} = \{\mathbf{x}^* \mid y^* = y^{(n)}\}$	A subset of D whose output value is equal to $y^{(n)}$ ($D_{\text{Agree}(n)}$ does not include $\mathbf{x}^{(n)}$)
$D_{\text{Disagree}(n)} = \{\mathbf{x}^* \mid y^* \neq y^{(n)}\}$	A subset of D whose output value is not equal to $y^{(n)}$

We define $Score_{\text{ROS}}(\mathbf{x}^{(n)}, y^{(n)})$ as the ratio between two unconditional outlier scores evaluated on $D_{\text{Agree}(n)}$ and $D_{\text{Disagree}(n)}$, respectively:

$$Score_{\text{ROS}}(y^{(n)}|\mathbf{x}^{(n)}) := \frac{o_U(\mathbf{x}^{(n)}; D_{\text{Agree}(n)})}{o_U(\mathbf{x}^{(n)}; D_{\text{Disagree}(n)})} \tag{4.4}$$

where $o_U(\mathbf{x}^{(n)}; D)$ denotes an unconditional outlier score calculated for $\mathbf{x}^{(n)}$ on the dataset D.

ROS measures the unusualness in the input $\mathbf{x}^{(n)}$ being associated with its output $y^{(n)}$. For normal instances $Score_{\text{ROS}}(\mathbf{x}^{(n)}, y^{(n)})$ will be low, which in turn indicates the outlier score from $D_{\text{Agree}(n)}$ is low and that of $D_{\text{Disagree}(n)}$ is high. On the other hand, instances with a high $Score_{\text{ROS}}(\mathbf{x}^{(n)}, y^{(n)})$ are deemed as outliers, because $Score_{\text{ROS}}(\mathbf{x}^{(n)}, y^{(n)})$ is high if the outlier score from $D_{\text{Agree}(n)}$ is high and that of $D_{\text{Disagree}(n)}$ is low.

Note that Equation (4.4) easily turns many existing unconditional outlier scores to conditional outlier scores. That is, we can compute and compare the conditional outlier score of the data instances by simply applying any unconditional outlier score – such as density-based outlier scores [Breunig et al., 2000, Papadimitriou et al., 2003], distance-based outlier scores [Knorr and Ng, 1997], or other unconditional outlier score reviewed in Section 2.2.1 – to the subsets of D and computing their ratio. Another advantage of this approach is that it can properly handle instances that fall in regions of the input space with low support. For a data instance that does not have enough support (*i.e.*, the instance falls in a sparse neighborhood of $\mathbf{X}$), it is not straightforward to come up with an outlier score that is confident. However, our outlier score suffers less from the issue, because both $o_U(\mathbf{x}^{(n)}; D_{\text{Agree}(n)})$ and $o_U(\mathbf{x}^{(n)}; D_{\text{Disagree}(n)})$ will be high in such a sparse region and, as a result, by cancelling each other out in Equation (4.4), the resulting conditional outlier score will not be high.

In summary, our new conditional outlier detection approach based on the ratio-score defines a general and flexible framework that allows one to plug in an unconditional outlier score and use it to perform conditional outlier detection. In the remainder of the chapter we will use our new conditional score in combination with the Local Outlier Factor (LOF) method and its score definition [Breunig et al., 2000], which we briefly summarize next.

4.2.2.2 Local Outlier Factor Recall that LOF is a nonparametric approach used to detect unconditional outliers based on the density of the local neighborhood of the target data instance. More specifically, it computes the outlier score of an instance by comparing the local density of the instance to the average local density of its k nearest neighbors:

$$o_U(\mathbf{x}^{(n)}; D) = LOF\left(\mathbf{x}^{(n)}, k; D\right) = \frac{\sum_{\mathbf{x}' \in N_k(\mathbf{x}^{(n)}; D)} \frac{lrd_k(\mathbf{x}'; D)}{lrd_k(\mathbf{x}^{(n)}; D)}}{|N_k(\mathbf{x}^{(n)}; D)|} \tag{4.5}$$

where $o_U(\mathbf{x}^{(n)}; D)$ is the unconditional outlier score for the instance $\mathbf{x}^{(n)}$ and dataset D, $N_k(\mathbf{x}^{(n)}; D)$ denotes the k-nearest neighborhood of the instance $\mathbf{x}^{(n)}$ in D and

$$lrd_k(\xi; D) := \frac{|N_k(\xi; D)|}{\sum_{o \in N_k(\xi; D)} \max(dist_k(o), dist(\xi, o))} \tag{4.6}$$

is the local reachability density, which measures the geometric dispersion of the k-nearest neighborhood, where $dist(\xi, o)$ denotes the distance between two instances ξ and o; and $dist_k(o)$ denotes the distance to the k-th nearest neighbor of o. We will use the Mahalanobis distance to compute the pairwise distances.

4.2.2.3 Ratio of Outlier Scores on Discriminative Projections Our newly designed conditional outlier score is defined by a ratio of two unconditional outlier scores defined over the input space. Hence, any issues affecting the quality of the unconditional scores are likely to be inherited by the new ratio score.

Recall that one of the recurring challenges of many unconditional outlier detection approaches is that they tend to exhibit poor performance when the data dimensionality is high (see Section 2.2). This is because in high-dimensional data spaces, with many (random) dimensions, all data objects appear to be sparse, and many of the distance metrics and density estimators become analytically ineffective and computationally intractable [Weber

et al., 1998, Hinneburg et al., 2000, Aggarwal et al., 2001]. As a result, outliers are hard to define and detect. It is reasonable to assume that these limitations also translate to the COD score based on the ratio outlier score, and devising a solution to improve the robustness of the method is appropriate.

One common way to resolve the problem of a high-dimensional space (in unconditional settings) is to reduce the dimensionality of the space via various dimensionality reduction methods, such as principal component or independent component analysis [Jolliffe, 1986, Hyvärinen et al., 2004], before the detection. However, the conditional outlier detection is different, since the importance of the input space and its individual dimensions depends on how important the dimensions are in defining (or predicting) the output. In such a case, various supervised space transformations or supervised metric learning approaches can be applied.

To cope with the dimensionality problem, in this work, we adopt a relatively simple supervised dimensionality reduction approach that relies on a discriminative model and its output to define a lower-dimensional projection of the original (high-dimensional) input data. In principle, one can use one or more such discriminative projections. In this work, we focus on and experiment with one dimensional projections, in which a high dimensional input space is reduced to a one-dimensional discriminative space. These projections can be built with the help of various classification learning methods. For example, by applying the logistic regression to the dataset D, we can obtain a probabilistic projection of $(\mathbf{x}^{(n)}, y^{(n)})$ representing $P(Y = y^{(n)}|\mathbf{X} = \mathbf{x}^{(n)})$. Similarly, by taking a raw output of the support vector machines model, we can obtain non-probabilistic discriminative projections. We denote such a discriminative projection function as f and the projection of the function as ϕ.

$$f : \mathbf{x}^{(n)} \rightarrow \phi^{(n)} \tag{4.7}$$

Obtaining a discriminative projection function f and its projections ϕ is equivalent to training (learning) of a model on the input-output instances in D. Assuming the logistic regression model, the parameters of the projection function are optimized as:

$$\theta_f = \arg\max_{\theta} \sum_{n=1}^{N} \log P(y^{(n)}|\mathbf{x}^{(n)}; \theta) \tag{4.8}$$

After θ_f is obtained from data, we define the projection function f on an observed input $\mathbf{x}^{(n)}$ as follows:

$$\phi^{(n)} = f\left(\mathbf{x}^{(n)}\right) = \frac{1}{1+\exp(-\mathbf{x}^{(n)}\theta_f)} \tag{4.9}$$

To combine the discriminative projections with our ROS approach, we first map the original data to the projected discriminative space. After that, we compute the ROS score (Equation (4.4)) only on the new projected space. We refer to this new score and the associated approach as Ratio of Outlier Scores on Discriminative Projections (ROS-DP) approach. Given the parameters of the disriminative projection f, the ROS-DP score is defined as:

$$Score_{\text{ROS-DP}}(y^{(n)}|\mathbf{x}^{(n)}, f) := \frac{o_U(f(\mathbf{x}^{(n)}); D_{\text{Agree}(n)})}{o_U(f(\mathbf{x}^{(n)}); D_{\text{Disagree}(n)})} = \frac{o_U(\phi^{(n)}; D_{f:\text{Agree}(n)})}{o_U(\phi^{(n)}; D_{f:\text{Disagree}(n)})} \tag{4.10}$$

where

$D_{f:\text{Agree}(n)} = \{\phi^* \mid y^* = y^{(n)}\}$	A subset of the projections of f on D whose output value is equal to $y^{(n)}$ ($D_{\text{Agree}(n)}$ does not include $(\phi^{(n)})$)
$D_{f:\text{Disagree}(n)} = \{\phi^* \mid y^* \neq y^{(n)}\}$	A subset of the projections of f on D whose output value is not equal to $y^{(n)}$

In the following, we show the advantages of our conditional outlier approach based on the ROS score and its combination with discriminative projections (ROS-DP) dimensionality reduction approach on multiple datasets. To assure fair comparison, we use local outlier factor (LOF) approach to calculate the unconditional outlier scores throughout the experiments.

4.2.3 Experiments

In this section, we test the above UCOD methods, including the newly designed ROS and ROS-DP approaches, by performing experiments on multiple synthetic datasets. After that we focus on real-world data based on the digit recognition (*MNIST*) dataset [LeCun et al., 1998].

Tested Methods We perform experiments with the following methods:

- *Local Outlier Factor on the Joint Space* (LOF) [Breunig et al., 2000] (Sections 2.2.1.2 and 4.2.2.2) – applies LOF to the joint space of all data attributes (both input and output)
- *Conditional outlier scoring based on the probabilistic model* (PROB) (Equation (4.3)) – uses $-\log P(y|\mathbf{x})$ estimated from the logistic regression model as the outlier score
- *Ratio of Outlier Scores* (ROS) (Equation (4.4)) – scoring based on our ratio-based outlier score calculated on the original input space
- *Ratio of Outlier Scores on Discriminative Projection* (ROS-DP) (Equation (4.10)) – scoring based on our ratio-based outlier score combined with a discriminative projection

For ROS and ROS-DP, we use LOF score as the base unconditional outlier score. For every instance of LOF (LOF and LOF used in ROS and ROS-DP), we set the number of neighbors to $k = 50$; and use Mahalanobis distance to measure the distance between pairs of instances. To obtain data models/discriminative projection functions in PROB and ROS-DP, we use L_2-regularized logistic regression and choose regularization parameters using the internal cross validation.

Experiment Setup The evaluation and comparisons of COD methods are often very challenging because outlier validation may be ambiguous and may require additional human feedback. For the purpose of our comparative evaluation, we conduct experiments on simulated outliers. In the following experiments on UCOD, we will use a fixed simulation process such that

1. In each simulation, select 1.0% of instances uniformly at random (*Outlier ratio* = 1.0%)
2. For each selected instance, invert the output value ($y_{\text{outlier}} = |y_{\text{original}} - 1|$)

The resulting outliers can be interpreted as contextually abnormal output signals (errors or mistakes). For example, in an annotated image dataset, the outliers would be perceived as images with incorrect labels.

We would like to stress that all methods (including their model building and outlier scoring stages) are run on data with simulated outliers. In other words, we never train a model on the original (clean) data and perform a test on the data with simulated outliers. Such a setup would be impractical since in real world applications where we do not know a priori which data instances should be excluded from training a model in order to obtain outlier-free data.

Evaluation Metrics We use *precision-alert rate* (PAR) curves [Hauskrecht et al., 2016] and *precision-recall* (PR) curves as our primary evaluation metrics. PAR measures the percentage of true outliers over all predicted outliers (precision) at different alert rates [Hauskrecht et al., 2016]. We report PAR in two ways: We present the *average PAR* (APAR) in $[0.00, 0.01]$ range, which coincides with the simulated outlier ratio, on all experiments. We also provide the PAR curves and compare the performance of different outlier scores on selected experiments.

The PR curves plot the overall performance on all ranges of the alert rate in terms of precision and recall tradeoff [Davis and Goadrich, 2006]. We report the area under the PR curve (AUPRC) and summarize the tradeoff over the entire range of threshold values.

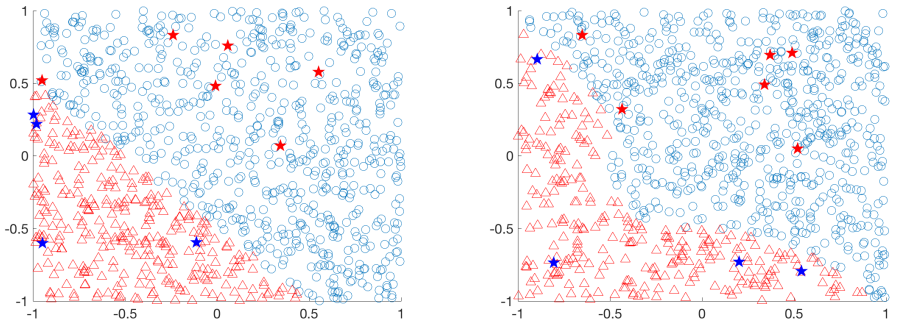
Briefly, precision and recall are defined as below:

$$\textit{Precision} = \textit{(True positive outliers)/(Predicted outliers)}$$
$$\textit{Recall} = \textit{(True positive outliers)/(True outliers)}$$

Note that the recall can be computed only when true outliers are known a priori (as in our simulated study) and may not be available in some real-world data analysis.

4.2.3.1 Synthetic Datasets 1

We first consider two synthetic datasets shown in Figure 4.2, referred to respectively as *SD1* and *SD2*. Each dataset consists of 1,000 instances



(a) Synthetic dataset 1 (*SD1*) (b) Synthetic dataset 2 (*SD2*)

Figure 4.2: Two synthetic datasets (*SD1* and *SD2*) with example conditional outliers (marked with a star). Colors represent the output assignment ($red = 1$; $blue = 0$).

with *two*-dimensional input data; each random variable X_i is generated by the uniform distribution between -1 and 1.

$$x_i \sim unif(\alpha, \beta); \qquad \alpha = -1, \;\; \beta = 1 \;\; (i = 1, 2)$$

The output Y is determined depending on the input values. *SD1* (Figure 4.2(a)) establishes a linear discriminative boundary, whereas *SD2* (Figure 4.2(b)) sets up a nonlinear discriminative boundary.

$$\textit{(SD1)} \qquad y^{(n)} = \begin{cases} 1, & \text{if } x_1^{(n)} + x_2^{(n)} < -0.5 \\ 0, & \text{otherwise.} \end{cases}$$

$$\textit{(SD2)} \qquad y^{(n)} = \begin{cases} 1, & \text{if } (x_1^{(n)})^3 + (x_2^{(n)})^3 < -0.5^3 \\ 0, & \text{otherwise.} \end{cases}$$

Conditional Outliers Each dataset contains 1.0% of conditional outliers, which are simulated by the above described process that flips the label of the instance. The instances marked with a star in Figure 4.2 indicate outliers.

	SD1		*SD2*	
	$\text{APAR}_{[0.00,0.01]}$	AUPRC	$\text{APAR}_{[0.00,0.01]}$	AUPRC
LOF	0.88 ± 0.08	0.69 ± 0.12	0.88 ± 0.08	0.66 ± 0.14
PROB	**0.99 ± 0.00**	**0.90 ± 0.00**	0.91 ± 0.06	0.64 ± 0.14
ROS	0.93 ± 0.03	0.81 ± 0.04	0.95 ± 0.04	0.79 ± 0.10
ROS-DP	**0.99 ± 0.01**	**0.89 ± 0.02**	0.94 ± 0.05	0.72 ± 0.14

Table 4.1: Average precision-alert rate in alert rate = $[0.00, 0.01]$ range ($\text{APAR}_{[0.00,0.01]}$) and area under the precision-recall curve (AUPRC) for the conditional outlier detection on synthetic datasets *SD1* and *SD2*. Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Higher APAR/AUPRC is better.

Results The results of the outlier detection experiments on *SD1* and *SD2* data are in Table 4.1. The results are averages over *five* repetitions of outlier simulations. We compare APAR and AUPRC of the tested methods. In both metrics, we find consistent improvements when we use the COD methods (PROB, ROS, or ROS-DP). On the dataset with a linear discriminative boundary (*SD1*), PROB and ROS-DP show statistically significant improvements over LOF, whereas ROS also shows a slight improvement over the unconditional outlier detection method. On the dataset with a nonlinear discriminative boundary (*SD2*), although there are no statistically significant differences, the COD methods generally beat LOF. The only exception is the AUPRC of PROB. This is because the base probabilistic model (*i.e.*, logistic regression) cannot properly learn the nonlinear decision boundary of *SD2*. However, we find ROS-DP has a favorable property in that it recovers the discriminability that the linear model lost. That is, while ROS-DP examines data instances on the same linear projection space as PROB, it outperforms PROB and performs comparably to the best method (ROS) on the dataset.

4.2.3.2 Synthetic Datasets 2 (Higher-dimensional Input)

Now consider two new synthetic datasets, referred to respectively as *SD3* and *SD4*. Each dataset consists of 1,000 instances with *ten*-dimensional input data; each random variable X_i is generated by the

	SD3		*SD4*	
	$\text{APAR}_{[0.00,0.01]}$	AUPRC	$\text{APAR}_{[0.00,0.01]}$	AUPRC
LOF	0.35 ± 0.23	0.14 ± 0.12	0.27 ± 0.17	0.06 ± 0.04
PROB	0.84 ± 0.06	0.57 ± 0.13	**0.68 ± 0.13**	**0.36 ± 0.16**
ROS	0.88 ± 0.07	0.59 ± 0.08	0.55 ± 0.24	**0.27 ± 0.18**
ROS-DP	**0.99 ± 0.01**	**0.88 ± 0.02**	**0.72 ± 0.13**	**0.43 ± 0.15**

Table 4.2: Average precision-alert rate in alert rate = $[0.00, 0.01]$ range ($\text{APAR}_{[0.00,0.01]}$) and area under the precision-recall curve (AUPRC) for the conditional outlier detection on synthetic datasets *SD3* and *SD4*. Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Higher APAR/AUPRC is better.

uniform distribution between -1 and 1.

$$x_i \sim unif(\alpha, \beta); \qquad \alpha = -1, \;\; \beta = 1 \;\; (i = 1, ..., 10)$$

The output Y is determined depending on the input values. As in the previous datasets, *SD3* establishes a linear discriminative boundary, whereas *SD4* defines a nonlinear discriminative boundary.

$$\textit{(SD3)} \qquad y^{(n)} = \begin{cases} 1, & \text{if } \sum_{i=1}^{10} x_i^{(n)} < -0.5 \\ 0, & \text{otherwise.} \end{cases}$$

$$\textit{(SD4)} \qquad y^{(n)} = \begin{cases} 1, & \text{if } \sum_{i=1}^{10} (x_i^{(n)})^3 < -0.5^3 \\ 0, & \text{otherwise.} \end{cases}$$

Conditional Outliers Each dataset contains 1.0% of conditional outliers, which are simulated by the above describe process.

Results Table 4.2 shows the results on *SD3* and *SD4* in terms of APAR and AUPRC. Again, the results are averages calculated based on *five* simulation rounds. The numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$). In general, we find the performance differences become more distinguished on high-dimensional data. On *SD3*,

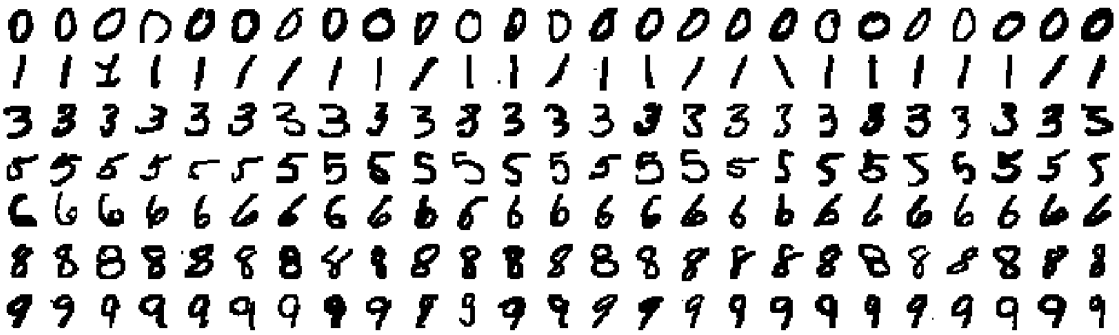


Figure 4.3: *MNIST* dataset [LeCun et al., 1998].

while ROS-DP shows a statistically superior performance, other COD methods (PROB and ROS) outperform LOF with a larger difference. This is because LOF equally examines all data attributes and hence fails to identify the abnormality that occurred only in the output space. On the other hand, our ratio-based counterpart (ROS) shows much competitive performance even though it relies on the same underlying LOF score.

On *SD4*, again, all the COD methods outperform LOF by a large margin. Also, we reaffirm ROS-DP can improve the performance of PROB on the datasets with a nonlinear discriminative boundary. Another interesting point is that, in a high-dimensional space, the performance difference between ROS-DP and ROS becomes larger. This demonstrates the benefit of dimensionality reduction via discriminative projection in our ratio-based (ROS) framework.

4.2.3.3 Public Image Datasets with Simulated Outliers Now we evaluate the methods on the *MNIST* dataset [LeCun et al., 1998]. The dataset contains images of handwritten digits (as in Figure 4.3), along with the ground truth labels telling what digits are scanned.

Experiment Setup We use *six* subsets of *MNIST*; each subset includes 2,000 randomly sampled images of two pre-selected digits (1,000 images per digit). The pre-selected pairs of digits are:

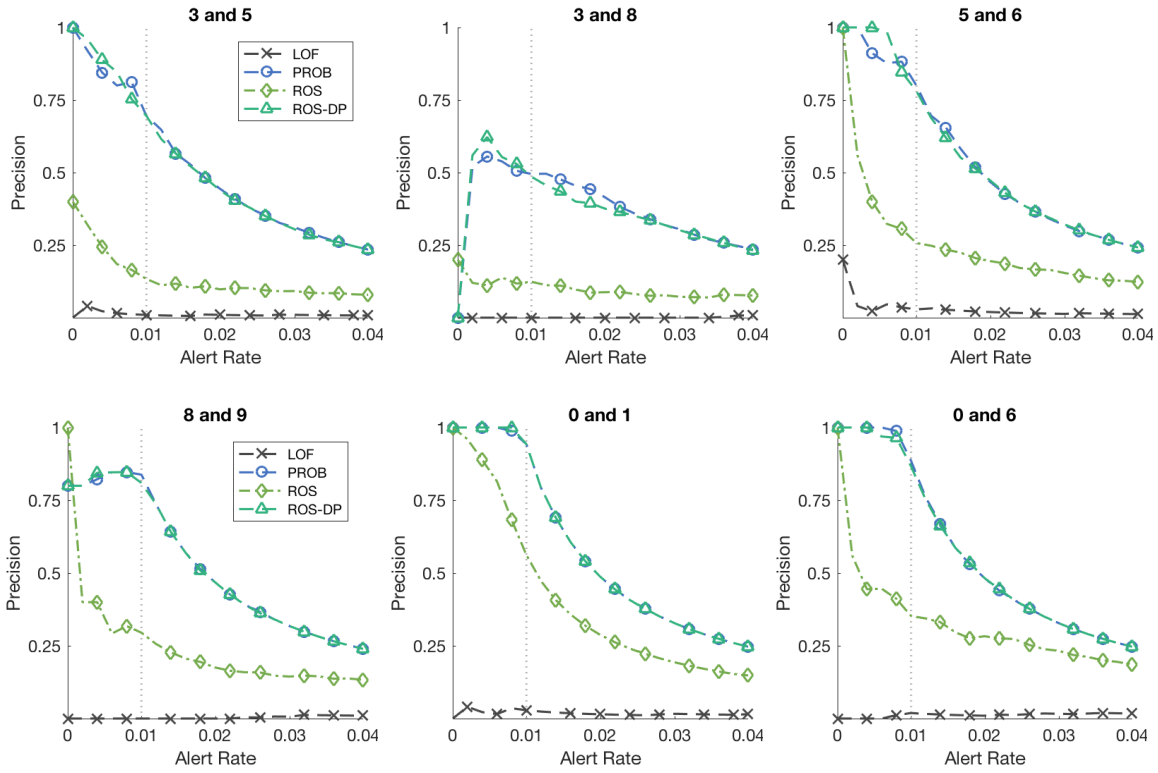


Figure 4.4: Precision-alert rates (PAR) at alert rates (detection thresholds) between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the simulated outlier ratio.

- 3 and 5
- 3 and 8
- 5 and 6
- 8 and 9
- 0 and 1
- 0 and 6

We note that we chose the digit pairs that are hard to distinguish based on the similarity in their visual patterns. Given the images as input, we create a new binary output label, such that it indicates the matching digit; *e.g.*, in the first subset, output values 0 and 1 indicate 3 and 5, respectively.

Simulating Outliers On each experiment, we simulate conditional outliers by selecting 1.0% of the dataset instances and by inverting their output labels. Given that each dataset contains two classes of handwritings with subtle differences, the simulated outliers can be

$APAR_{[0.00,0.01]}$	LOF	PROB	ROS	ROS-DP
3 and 5	0.02 ± 0.04	**0.83 ± 0.04**	0.22 ± 0.06	**0.86 ± 0.02**
3 and 8	0.00 ± 0.00	**0.52 ± 0.09**	0.13 ± 0.08	**0.55 ± 0.07**
5 and 6	0.04 ± 0.05	**0.92 ± 0.02**	0.41 ± 0.08	**0.94 ± 0.02**
8 and 9	0.00 ± 0.00	**0.83 ± 0.04**	0.35 ± 0.08	**0.83 ± 0.09**
0 and 1	0.03 ± 0.04	**0.99 ± 0.01**	0.82 ± 0.03	**1.00 ± 0.00**
0 and 6	0.00 ± 0.01	**0.98 ± 0.01**	0.48 ± 0.05	**0.97 ± 0.03**

Table 4.3: Average precision-alert rate in alert rate = $[0.00, 0.01]$ range ($APAR_{[0.00,0.01]}$). Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Higher APAR is better.

AUPRC	LOF	PROB	ROS	ROS-DP
3 and 5	0.01 ± 0.00	**0.71 ± 0.05**	0.08 ± 0.01	**0.73 ± 0.04**
3 and 8	0.01 ± 0.00	**0.46 ± 0.07**	0.07 ± 0.01	**0.45 ± 0.06**
5 and 6	0.01 ± 0.00	**0.83 ± 0.01**	0.20 ± 0.06	**0.84 ± 0.03**
8 and 9	0.01 ± 0.00	**0.76 ± 0.03**	0.18 ± 0.06	**0.76 ± 0.08**
0 and 1	0.01 ± 0.00	**0.94 ± 0.01**	0.52 ± 0.06	**0.95 ± 0.01**
0 and 6	0.01 ± 0.00	**0.92 ± 0.01**	0.32 ± 0.05	**0.91 ± 0.02**

Table 4.4: Area under the precision-recall curve (AUPRC). Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Higher AUPRC is better.

interpreted as errors or mistakes in image labels.

Results Figure 4.4 and Tables 4.3 and 4.4 present the results on the six digit subsets. All the results are averages over *five* simulation runs. The numbers shown in boldface in the tables indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set.

The PAR curves (Figure 4.4) show that the precision of the tested methods at different alert rates (detection thresholds) ranging between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the ratio of simulated outliers. Note that the PAR curves show the precision of the outlier detection can be controlled by tightening the alert rate. Also, in all experiments, the COD methods offer more precise outlier scores than unsupervised LOF over the joint input-output space. The

same information is observed in Tables 4.3 and 4.4, where all three COD methods clearly improve APAR and AUPRC over LOF. Among the COD methods, PROB and ROS-DP show superior performance. Albeit not as good as other two COD methods, ROS consistently improves the results over LOF. All in all, the results on the real-world image datasets confirm the validity of the COD methods in addressing their intended problem.

4.2.4 Discussion

In this section, we studied the UCOD problem, which is a special case of MCOD where the output dimensionality is one. We focused on and presented the key ideas of two approaches: the probabilistic model-based (PROB) approach, that is extensively used in the literature, and the new ratio-based outlier scoring (ROS) approach, that is newly proposed. To begin with, we revisited the existing probabilistic model-based methods and defined our own solution that will be used in our following discussion. We then switched to the ratio-based approach and defined a new method that computes the conditional score based on unconditional outlier detection methods and their scores. We also presented a dimensionality reduction technique for the ratio-based approach using discriminative projection. The importance of this new approach is that it bridges the gap between the development of unconditional and conditional outlier detection methodologies and, hence, provides users with more flexibility when designing conditional outlier detection solutions.

The experiments on synthetic and public image datasets with simulated outliers demonstrated that our COD methods achieved good performance. The results showed that a probabilistic score can be used to identify conditional outliers, which an unconditional outlier detection method cannot discover. The ratio-based score also showed competitive or superior performance, especially when data are high-dimensional and have a nonlinear discriminative boundary.

Lastly, we note that we will use the methods presented here as basic building blocks for developing the solutions for a more general multivariate conditional outlier detection (MCOD) problem in the next section.

4.3 MULTIVARIATE CONDITIONAL OUTLIER DETECTION

In this section, we turn our focus to the multivariate conditional outlier detection (MCOD) problem that concerns data with multi-dimensional output variables. Generally, the MCOD problem is more complex than the UCOD problem primarily in that the inter-dependences with respect to the output variables, as well as the contextual dependences, should be taken into account when identifying outliers.

To cope with the increased complexities, we take two UCOD approaches presented in the previous section and extend them to handle the MCOD problem. In doing so, one straightforward approach is to divide and solve an MCOD problem as d independent UCOD problems, such that each UCOD problem focuses on one output dimension (analogous to the Binary Relevance approach in MLC; Section 2.1.1). This simple solution, however, inevitably ignores all inter-dependence relations and does not suffice to fully address the MCOD problem. That is, in the UCOD setting, only dependences between input and output (contextual dependences) are considered when computing the outlier scores, but in the general MCOD setting, the output variables may be dependent on each other (output inter-dependences).

Accordingly, our discussion throughout this section focuses on how to properly extend COD approaches to effectively identify multivariate conditional outliers in data. Our proposed solutions are largely inspired by the structured (decomposable) models used for multi-label classification (MLC) which we studied in Chapter 3. More specifically, we utilize the decomposable MLC models and methods to capture stochastic dependence relations among input and output attributes, and incorporate them into our COD approaches to compute outlier scores for multivariate conditional outliers. Through the experimental results, we demonstrate the performance of our proposed methods that successfully identify multivariate conditional outliers.

4.3.1 Probabilistic Approach to Multivariate Conditional Outlier Detection

This section describes a probabilistic approach that tackles the MCOD problem by building a probabilistic model of $P(\mathbf{Y}|\mathbf{X})$. Similarly to the probabilistic UCOD solution, the model

is built (learned) from all available data, aiming to capture and summarize all relevant dependences among data attributes and their strength as observed in the data. Conditional outliers are then identified with the help of this model. More specifically, a conditional outlier corresponds to a data instance that is assigned a low probability by the model.

To convert the above idea into a workable MCOD framework, multiple issues need to be resolved. First, it is unclear how the probabilistic model $P(\mathbf{Y}|\mathbf{X})$ should be represented and parameterized. To address this issue, we resort to and adapt structured probabilistic data models of $P(\mathbf{Y}|\mathbf{X})$ that provide an efficient representation of input-output relations by decomposing the model using the chain rule into a product of univariate probabilistic factors $P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i)) : i = 1, ..., d$; *i.e.*, each response Y_i is dependent on $\mathbf{X}$ and a subset of the other responses $\boldsymbol{\pi}(Y_i)$. The univariate conditional probability models and their learning are rather common and well-studied, and multiple probabilistic models (*e.g.*, logistic regression or naïve Bayes) can be applied to implement them. These were also reviewed and discussed in Subsection 4.2.1 devoted to probabilistic approaches supporting UCOD. We note that the structured probabilistic data models were originally proposed and successfully applied to support structured output prediction problems [Zhang and Zhou, 2013]. However, their application to outlier detection problems has not been formally investigated. The key difference between the two tasks is that while in prediction we seek to find outputs that maximize the probability given the inputs, in conditional outlier detection we aim to identify abnormal (or low probability) associations in between observed inputs and outputs.

The second issue concerns the question of how to define and score multivariate outliers with the help of the probabilistic models. For example, the outliers (in varied application contexts) may manifest themselves differently across the different output dimensions. Take for instance analysis of network attacks on multiple network nodes by exploring their normal and saturated traffic state given the context. In that case, one may want to identify unusual data instances with abnormal outputs (many saturated states across network nodes). Other applications may prefer outliers that are manifested in just one or a few output dimensions. These different outlier definitions may lead to different outlier scoring methods. Another issue that may affect identification of outliers is the quality of probability estimates trained on finite size data and inaccuracies in probability estimates that may lead to, which may affect

the identification of outliers. To address this concern, we present outlier scoring methods that combine probability estimates with the help of weights reflecting their reliability in assessment of outliers.

4.3.1.1 Data Modeling Our probabilistic approach works by analyzing data instances using a statistical model representing the conditional distribution $P(\mathbf{Y}|\mathbf{X})$. In general, the representation and learning of such a model from data may be very costly because the number of possible output combinations grows exponentially with d. To avoid such inefficiencies and yet achieve an accurate data representation for outlier detection, we first decompose the conditional joint into a product of conditional univariate distributions using the chain rule of probability:

$$P(Y_1, ..., Y_d|\mathbf{X}) = \prod_{i=1}^{d} P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i)) \tag{4.11}$$

where $\boldsymbol{\pi}(Y_i)$ denotes the parents of Y_i; *i.e.*, all the output variables preceding Y_i [Read et al., 2009]. This decomposition lets us represent $P(\mathbf{Y}|\mathbf{X})$ in terms of d univariate conditional factors, $P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i))$, each factor representing one output dimension. We note that similarly to UCOD, multiple probabilistic models (*e.g.*, logistic regression, naïve Bayes, relevance vector machine [Tipping, 2001], or support vector machine with probabilistic output [Platt, 1999]) can be used to represent these factors and learn them from data.

In the remainder of this chapter, we assume a logistic regression model is used to represent each of these factors. This choice of the model allows us to handle input spaces defined by a mixture of continuous and discrete variables (*i.e.*, $\mathbf{X}$ and $\boldsymbol{\pi}(Y_i)$ that predict Y_i). Each model can be learned by optimizing the likelihood based loss function. When the input dimensionality is high the learning can be enhanced through regularization techniques [Ng, 2004, Cetin and Karl, 2001] .

4.3.1.2 Outlier Scoring Once the model of $P(\mathbf{Y}|\mathbf{X})$ is learned from data, it can be applied to calculate conditional probability for any data instance $< \mathbf{x}^{(n)}, \mathbf{y}^{(n)} >$. Similarly to the previous section, outliers are data instances that are assigned a low probability $\widetilde{P}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)})$.

To match the definition of the outlier score (higher score implies stronger outlier) for the univariate probabilistic score, we define the multivariate probabilistic conditional outlier score for model $\mathcal{M}$ as:

$$Score_{\text{MPROB}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = -\log \widetilde{P}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}; \mathcal{M}) \tag{4.12}$$

Rewriting the multivariate conditional probability using the chain rule we get:

$$Score_{\text{MPROB}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} -\log \widetilde{P}(y_i^{(n)}|\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)}); \mathcal{M}). \tag{4.13}$$

Now using the outlier scoring for UCOD (see Equation (4.3)) the probabilistic MCOD score can be rewritten as:

$$Score_{\text{MPROB}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} Score_{\text{PROB}(i)} \left(y_i^{(n)} \middle| \mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)}) \right) \tag{4.14}$$

In other words, the multivariate probabilistic COD score can be expressed as the sum of univariate probabilistic COD scores; one univariate COD score per output dimension.

This multivariate score decomposition is extremely important, since it can be the basis of many other multivariate outlier scoring methods formed by plugging in the definitions of other univariate COD scores, such as those that rely on ratio of outliers score (ROS) that were presented earlier in this chapter.

4.3.1.3 Decomposable Data Model with Circular Dependences

In theory, the product $P(Y_1, ..., Y_d|\mathbf{X}) = \prod_{i=1}^{d} P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i))$ in (Equation (4.11)) should be invariant regardless of the chain order (order of Y_i). Nevertheless, in practice, different chain orders produce different conditional joint distributions as they draw in models learned from different data [Dembczynski et al., 2010]. For this reason, several structure learning methods that determine the optimal set of parents have been proposed [Zhang and Zhang, 2010, Kumar et al., 2012]. However, these methods require at least $O(d^2 t_c)$ of time, where t_c denotes the time of learning a base statistical model (*e.g.*, logistic regression). Such a complexity would negatively affect many outlier detection applications, especially when the output dimensionality d is high. Instead, we address the issue of the chain order by relaxing Equation (4.11) and by permitting *circular dependences* among the output variables. More specifically, we let $\boldsymbol{\pi}(Y_i)$, the parents of Y_i, be all the remaining output variables. That is, we approximate $P(Y_1, ..., Y_d|\mathbf{X}) = \prod_{i=1}^{d} P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i))$ with:

$$\Psi(Y_1, ..., Y_d|\mathbf{x}) = \prod_{i=1}^{d} P(Y_i|\mathbf{X}, \mathbf{Y}_{-i}) \tag{4.15}$$

where $\mathbf{Y}_{-i}$ denotes the values of all other output variables except Y_i.

This new decomposition allows us to capture the interactions among the output variables, as well as the input-output relations, using a collection of individually trained probabilistic functions with a relaxed conditional independence assumption. We note that although the new conditioning set for each output dimension always includes other outputs, those outputs that do not contribute to the prediction, can be regularized out when learning the model from data, and hence controlling the complexity of the individual models. Finally, we note that the new decomposition can be substituted into the multivariate probabilistic COD score in Equation (4.13) and Equation (4.14) to define a slightly different yet very reasonable MCOD score:

$$Score_{\text{MPROB-RELAX}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} -\log \widetilde{P}(y_i^{(n)}|\mathbf{x}^{(n)}, \mathbf{y}_{-i}^{(n)}; \mathcal{M}) \tag{4.16}$$

4.3.1.4 Outlier Scoring with Reliability Weights

Note that our probabilistic scores and models assumed all our probability estimates and the models generating them are of high quality. However, in practice, the models that produce the probability estimates may not be all equally reliable as they are trained from a finite number of samples (this is important especially when the number of input and output variables is high, and the sample size is small). Also, some dimensions of $Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i)$ may not fit well the base statistical assumption (which in this section is a logistic curve) and result in miscalibrated estimations. Consequently, if we treat $P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i))$ for all $i = 1, ..., d$ equally and merely search for the regions with low probabilities, the resulting scores degenerate to a noisy vector, which makes the detection of true irregularities hard.

To alleviate the above issues, we propose to consider the reliability of each estimated conditional probability and incorporate it into the outlier score. Let $\rho_i^{(n)}$ define a conditional probability estimate for the data point $< \mathbf{x}^{(n)}, \mathbf{y}^{(n)} >$ and output dimension i that is generated either via the chain model or the proxy model with circular output dependences. Let $\boldsymbol{\rho}^{(n)}$ be a collection of all ρ_is, that is, $\boldsymbol{\rho}^{(n)} = \{\rho_1^{(n)}, \rho_2^{(n)}, \cdots, \rho_d^{(n)}\}$. In that case, the multivariate probabilistic score (Equation (4.13)) can be rewritten as:

$$Score_{\text{MPROB}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = -\sum_{i=1}^{d} \log \rho_i^{(n)} \tag{4.17}$$

One way to incorporate the reliability of each probability estimate and combine it with conditional probabilities is to define a weighted outlier score:

$$Score_{\text{RW}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = -\sum_{i=1}^{d} w_i \log \rho_i^{(n)} \tag{4.18}$$

where w_i denotes the reliability weight of the model used to score the i-th dimension. Trivially, when $w_i = 1$ for all dimensions $i = 1, ..., d$, the score becomes equivalent to Equation (4.17).

Calculating Reliability Weights One of the widely used metrics that assesses the reliability of a probabilistic predictive model is the Brier score [Brier, 1950] which measures

the quality of the model based on model's probability outputs. The Brier score is defined by averaging the squared errors of the probability estimates over all instances:

$$N^{-1}\sum_{n=1}^{N}(f^{(n)}-o^{(n)})^2 = N^{-1}\sum_{n=1}^{N}(1-\rho_i^{(n)})^2 \tag{4.19}$$

where $f^{(n)}$ and $o^{(n)}$ respectively denote the predicted probability and actual outcome of the n-th instance. For our weighting purpose (Equation (4.18)), however, direct application of the Brier score to the assessment of model quality would not be appropriate as it imposes different penalties for different errors and varies the distribution of errors [Willmott and Matsuura, 2005] (the mean squared error penalizes larger errors more than smaller errors). Therefore we compute the reliability using Equation (4.19) without squaring the error (*i.e.*, the mean estimation error), which allows us to estimate the quality of each estimate dimension ρ_i without distorting the distribution of errors. We finally define the reliability weight w_i by taking the inverse of this reliability measure.

More formally, let $\epsilon_i^{(n)} = 1-\rho_i^{(n)}$ be the estimation error in probability on the dimension i for the n-th data instance. We define the reliability weight w_i for the outlier score in Equation (4.18) as:

$$w_i = \frac{N}{\sum_{n=1}^{N}\epsilon_i^{(n)}} \tag{4.20}$$

This weight effectively prioritizes the components of the outlier score, such that contribution of outlier scores for more reliable partial models covering the output dimensions increases, whereas the contribution from noisy (unreliable) models and their dimensions decreases.

Local Reliability Weights The above weighting scheme (Equation (4.20)) implicitly assumes that the reliability of probability estimates (*i.e.*, the quality of a model) is invariant across all data regions. However, the assumption often does not hold because in most practical problems, especially in high-dimensional data spaces, data is not uniformly distributed in its attribute space. As a result, modeling and estimation of $P(Y_i|\mathbf{X}, \boldsymbol{\pi}(Y_i))$ cannot be achieved properly in data-sparse regions of the space.

We tackle such a sparsity issue by evaluating the reliability of each dimension of $\boldsymbol{\rho}$ locally in the region around the instance that we want to test. This localized approach can be implemented as follows:

$$Score_{\text{LRW}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = -\sum_{i=1}^{d} w_i^{(n)} \log \rho_i^{(n)} \tag{4.21}$$

where

$$w_i^{(n)} = \frac{|N_k(n)|}{\sum_{n \in N_k(n)} \epsilon_i^{(n)}} \tag{4.22}$$

and $N_k(n)$ denotes k-nearest neighbors of the n-th instance in the original input (context) space. In the next section, we show the benefits of our reliability weights and outlier scores through experimental results.

4.3.1.5 Experiments To validate and demonstrate the performance of our approach, we conduct experiments with multivariate data obtained from various domains. Through the empirical analysis in this section, we would like to verify the advantages of (1) adopting the COD approach, (2) considering the dependence relations among outputs, (3) weighting via reliability estimation, and (4) localized reliability weights and outlier scores. Below we describe our experimental design and present the evaluation results.

Tested Methods To achieve our objectives, we perform experiments with the following methods:

- *Local Outlier Factor for the Joint Input-Output Space* (LOF) [Breunig et al., 2000] – LOF is an unconditional method that estimates outliers using a relative local density measure and helps to find instances that fall in sparse regions of data. Here we apply it to the joint space of all data attributes (see Sections 2.2.1.2 and 4.2.2.2 for details).
- *Conditional outlier detection with d independent UCOD models* (I-PROD) – Solves the multivariate conditional outlier detection problem by considering d independent conditional probability models $P(Y_i|\mathbf{X})$ (where Y_i is not dependent on other output variables) and UCOD scores defined on these models.

Dataset	$N / m / d$	Domain	Value Description	
			Input	Output
Mediamill	43,907 / 120 / 101	Video	Video frames	Concepts
Yahoo-business	11,214 / 21,924 / 30	Text	News articles	Topics
Yahoo-arts	7,484 / 23,146 / 26	Text	News articles	Topics
Bibtex	7,395 / 1,836 / 159	Text	Paper metadata	Topics
Enron	1,702 / 1,001 / 53	Text	Emails	Properties
Yeast	2,417 / 103 / 14	Biology	Genes	Functionalities
Birds	645 / 276 / 19	Sound	Bird songs	Species
Cal500	502 / 68 / 174	Music	Waveforms	Annotations

Table 4.5: Dataset characteristics (N: number of instances, m: input dimensionality, d: output dimensionality).

- *MCOD without weighting* (M-PROD) – Solves the multivariate conditional outlier detection problem by considering d dependent conditional probability models (with circular dependences) and UCOD scores defined on these models. (Equation (4.17))
- *MCOD with Reliability Weights* (M-RW) Solves the multivariate conditional outlier detection problem by considering d dependent conditional probability models (with circular dependences) and reliability weighted UCOD scores defined on these models. (Equation (4.18))
- *MCOD with Local Reliability Weights* (M-LRW) Solves the multivariate conditional outlier detection problem by considering d dependent conditional probability models (with circular dependences) and local reliability weighted UCOD scores on these models (Equation (4.21))

To obtain data models in I-PROD, M-PROD, M-RW, and M-LRW, we use L_2-regularized logistic regression and choose their regularization parameters by cross validation. For LOF and M-LRW, we set the number of neighbors $k = 100$.

Datasets We use *eight* public datasets with multi-dimensional input and output.[2] These are collected from various application domains, including semantic video/image annotation (*Mediamill*), text categorization (*Yahoo* datasets, *Enron*), biology (*Yeast*), and sound recog-

[2]Datasets are available at http://mulan.sourceforge.net [Tsoumakas et al., 2010].

nition (*Birds*). Table 4.5 summarizes the characteristics of the datasets, such as dataset size, data domain, and short descriptions of the input and output variables.

Experiment Setup For our comparative evaluation, we simulate multivariate conditional outliers by perturbing the output space of data. There are two parameters in our simulation process. *Outlier ratio* specifies how many outliers per simulation are injected. We set this parameter to 1.0% throughout the experimental study. *Outlier dimensionality* specifies how many output dimensions per outlier to be perturbed. We vary this parameter relative to the output dimensionality by perturbing $\{2.5, 5.0, 10.0, 20.0\}$% of outputs. Therefore a dataset creates up to four sets of experiments.[3] To summarize, we simulate outliers as:

1. In each simulation, select 1.0% of instances uniformly at random
2. For each of the selected instances in Step 1, perturb the values in $\{2.5, 5.0, 10.0, 20.0\}$% of the output dimensions uniformly at random ($y_{\text{perturbed}} = |y_{\text{original}} - 1|$)

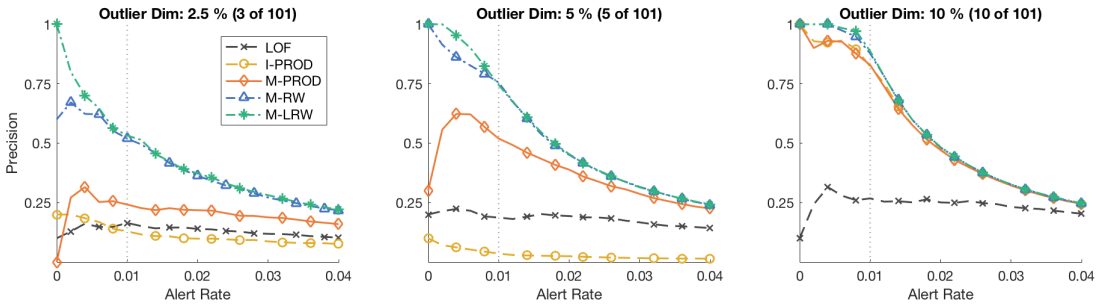
These simulated outliers can be interpreted as contextually abnormal output signals (errors or mistakes) in each application (see Table 4.5). For example, in semantic video annotation (*e.g.*, *Mediamill*), the outliers (perturbed output values) can be perceived as video frames with inaccurate concept tags. In text categorization (*e.g.*, *Yahoo-business*), the outliers can be seen as news articles with incorrectly assigned topics.

We would like to stress that all methods (including both the model building and outlier scoring stages) are run on data with simulated outliers. In other words, we never learn a model on the original (unperturbed) data and detect outliers on the simulated (perturbed) data. Such an experimental setting would be impractical since in real world applications we do not know a priori what data instances to remove to learn a model from outlier-free data.

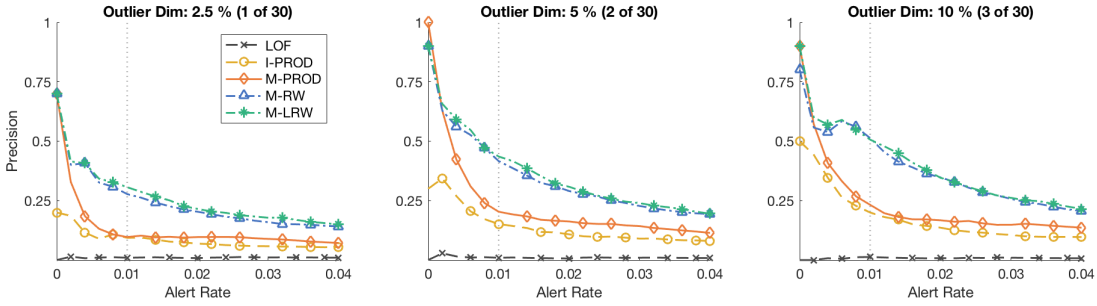
Evaluation Metrics We use the *precision-alert rate* (PAR) curves, *average PAR* (APAR) in $[0.00, 0.01]$ range, and *area under the precision-recall curve* (AUPRC) as our evaluation metric. See Section 4.2.3 for detailed description of the metrics.

Results Figures 4.5-4.7 and Tables 4.6-4.7 present the results of the five tested methods.

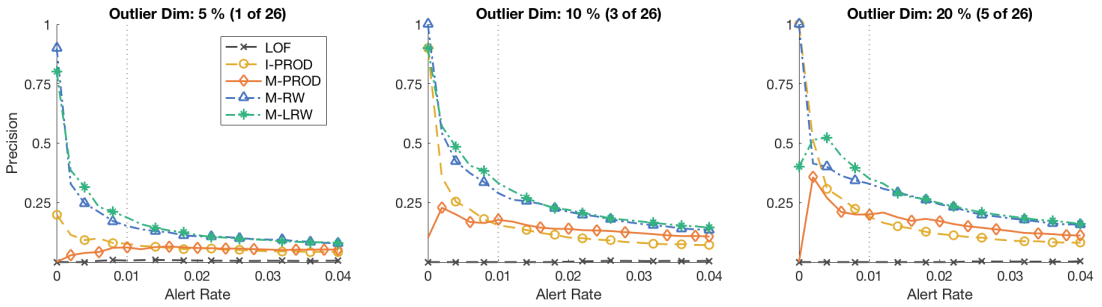
[3]For *Yahoo-arts*, *Yeast*, and *Birds*, outlier dimensionality = 2.5% cannot be applied due to the low output dimensionality.



(a) Mediamill (outlier dimensionality = $\{2.5, 5.0, 10.0\}\%$)



(b) Yahoo-business (outlier dimensionality = $\{2.5, 5.0, 10.0\}\%$)

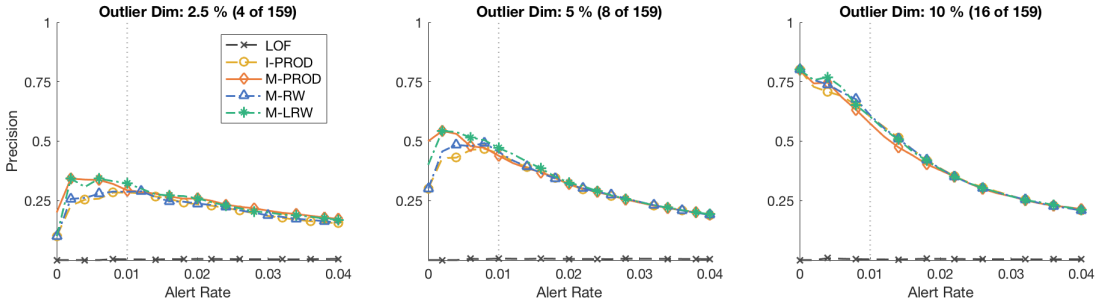


(c) Yahoo-arts (outlier dimensionality = $\{5.0, 10.0, 20.0\}\%$)

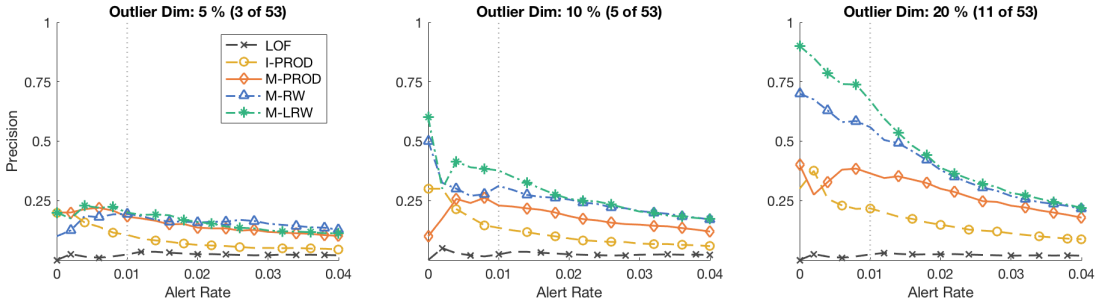
Figure 4.5: Precision-alert rate (PAR) at alert rates (detection thresholds) between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the simulated outlier ratio.

All results are obtained from *ten* repeats.

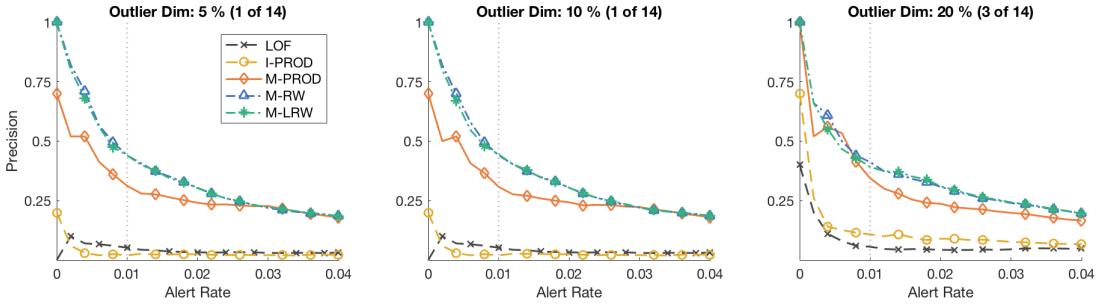
(1) *Precision-Alert Rate* Figures 4.5-4.7 show PARs at different alert rates (detection thresholds). X-axes show alert rate, ranging between 0.00 and 0.04; Y-axes show



(a) Bibtex (outlier dimensionality = $\{2.5, 5.0, 10.0\}\%$)



(b) Enron (outlier dimensionality = $\{5.0, 10.0, 20.0\}\%$)

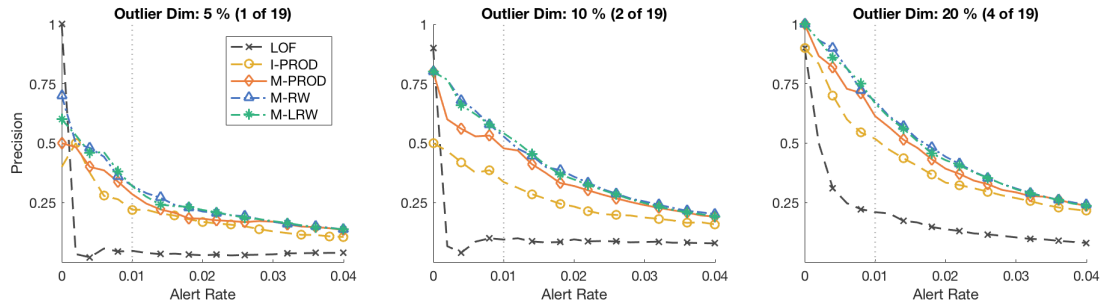


(c) Yeast (outlier dimensionality = $\{5.0, 10.0, 20.0\}\%$)

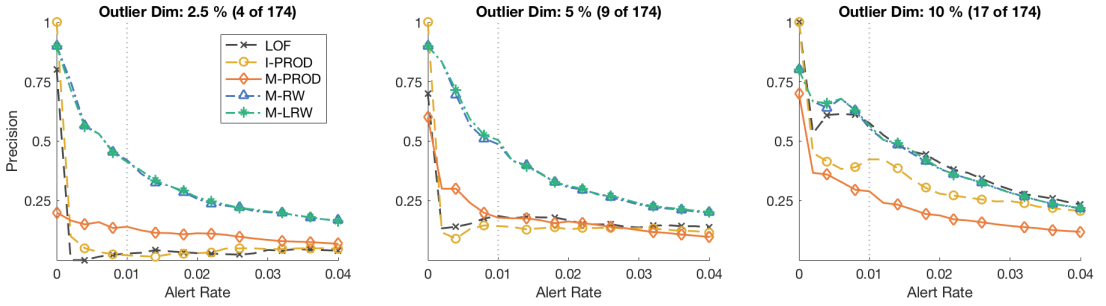
Figure 4.6: Precision-alert rate (PAR) at alert rates (detection thresholds) between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the simulated outlier ratio.

PAR. For each dataset, three sets of plots are displayed for different outlier dimensionality ($\{2.5, 5.0, 10.0\}\%$ or $\{5.0, 10.0, 20.0\}\%$).[4] Meanwhile, Table 4.6 presents the same result

[4]The plots of three configurations – *Mediamill* with outlier dimensionality = 20%, *Yahoo-business* with outlier dimensionality = 20%, *Bibtex* with outlier dimensionality = 20%, *Enron* with outlier dimensionality



(a) Birds (outlier dimensionality = $\{5.0, 10.0, 20.0\}$%)



(b) Cal500 (outlier dimensionality = $\{2.5, 5.0, 10.0\}$%)

Figure 4.7: Precision-alert rate at alert rates (detection thresholds) between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the simulated outlier ratio.

set in a different format – it shows the average PAR (APAR) in $[0.00, 0.01]$ range. The numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set. In general, PARs improve as the outlier dimensionality increases, because outliers with larger perturbations are easier to detect.

Comparing the COD approaches (I-PROD, M-PROD, M-RW, and M-LRW) to the unconditional approach (LOF), the conditional approaches clearly outperform. M-PROD, M-RW, and M-LRW almost always produce better PAR than LOF. Although I-PROD performs worse than LOF on *Mediamill*, *Yeast*, and *Cal500*, more often I-PROD outperforms LOF (Table 4.6 shows this more clearly). On the other hand, as expected, LOF hardly detects

= 2.5%, and *Cal500* with outlier dimensionality = 20% – are omitted for space limitation. Please see Table 4.6 for the omitted results.

$APAR_{[0.00,0.01]}$	Outlier dimensionality = 2.5%					Outlier dimensionality = 5.0%				
	Baselines		MCODE			Baselines		MCODE		
	LOF	I-PROD	M-PROD	M-RW	M-LRW	LOF	I-PROD	M-PROD	M-RW	M-LRW
Mediamill	0.14 ± 0.16	0.17 ± 0.09	0.26 ± 0.17	**0.61 ± 0.12**	**0.69 ± 0.09**	0.20 ± 0.17	0.06 ± 0.05	0.57 ± 0.14	**0.85 ± 0.05**	**0.90 ± 0.04**
Yahoo-business	0.01 ± 0.02	0.13 ± 0.06	**0.21 ± 0.10**	**0.36 ± 0.09**	**0.38 ± 0.07**	0.01 ± 0.03	0.25 ± 0.08	**0.43 ± 0.11**	**0.56 ± 0.08**	**0.58 ± 0.07**
Yahoo-arts	-	-	-	-	-	0.00 ± 0.01	0.11 ± 0.07	0.04 ± 0.04	**0.26 ± 0.06**	**0.29 ± 0.08**
Enron	0.01 ± 0.03	0.12 ± 0.19	0.11 ± 0.11	0.06 ± 0.11	0.05 ± 0.09	0.01 ± 0.03	0.15 ± 0.22	0.20 ± 0.17	0.17 ± 0.22	0.21 ± 0.26
Bibtex	0.00 ± 0.01	0.25 ± 0.28	0.32 ± 0.30	0.27 ± 0.29	0.33 ± 0.30	0.00 ± 0.01	**0.44 ± 0.27**	**0.49 ± 0.28**	**0.47 ± 0.28**	**0.51 ± 0.27**
Yeast	-	-	-	-	-	-	-	-	-	-
Birds	-	-	-	-	-	0.04 ± 0.08	**0.34 ± 0.22**	**0.39 ± 0.25**	**0.45 ± 0.21**	**0.46 ± 0.22**
Cal500	0.01 ± 0.02	0.05 ± 0.11	0.15 ± 0.18	**0.59 ± 0.20**	**0.59 ± 0.19**	0.13 ± 0.11	0.13 ± 0.16	0.27 ± 0.23	**0.66 ± 0.20**	**0.66 ± 0.19**

$APAR_{[0.00,0.01]}$	Outlier dimensionality = 10.0%					Outlier dimensionality = 20.0%				
	Baselines		MCODE			Baselines		MCODE		
	LOF	I-PROD	M-PROD	M-RW	M-LRW	LOF	I-PROD	M-PROD	M-RW	M-LRW
Mediamill	0.27 ± 0.16	**0.92 ± 0.03**	**0.91 ± 0.04**	**0.97 ± 0.03**	**0.98 ± 0.03**	0.30 ± 0.12	**0.99 ± 0.02**	**0.99 ± 0.01**	**1.00 ± 0.01**	**1.00 ± 0.00**
Yahoo-business	0.01 ± 0.01	0.32 ± 0.10	**0.42 ± 0.13**	**0.57 ± 0.06**	**0.57 ± 0.07**	0.01 ± 0.02	**0.36 ± 0.13**	0.25 ± 0.09	**0.39 ± 0.05**	**0.41 ± 0.04**
Yahoo-arts	0.00 ± 0.00	0.29 ± 0.05	0.18 ± 0.07	**0.44 ± 0.06**	**0.47 ± 0.05**	0.00 ± 0.00	0.36 ± 0.05	0.26 ± 0.07	0.39 ± 0.06	**0.45 ± 0.07**
Enron	0.03 ± 0.05	**0.21 ± 0.24**	**0.23 ± 0.17**	**0.31 ± 0.27**	**0.39 ± 0.29**	0.02 ± 0.04	0.26 ± 0.32	0.35 ± 0.21	**0.62 ± 0.24**	**0.78 ± 0.17**
Bibtex	0.00 ± 0.01	**0.70 ± 0.20**	**0.70 ± 0.18**	**0.71 ± 0.20**	**0.72 ± 0.17**	0.00 ± 0.00	**0.88 ± 0.11**	**0.86 ± 0.12**	**0.88 ± 0.11**	**0.87 ± 0.11**
Yeast	0.08 ± 0.07	0.04 ± 0.06	0.45 ± 0.11	**0.64 ± 0.06**	**0.63 ± 0.05**	0.13 ± 0.09	0.17 ± 0.11	**0.52 ± 0.08**	**0.56 ± 0.07**	**0.54 ± 0.08**
Birds	0.07 ± 0.11	**0.42 ± 0.31**	**0.56 ± 0.24**	**0.66 ± 0.18**	**0.66 ± 0.19**	0.32 ± 0.22	**0.67 ± 0.25**	**0.78 ± 0.19**	**0.85 ± 0.12**	**0.84 ± 0.13**
Cal500	**0.56 ± 0.20**	0.41 ± 0.31	0.35 ± 0.29	**0.66 ± 0.23**	**0.66 ± 0.23**	**0.97 ± 0.06**	**0.88 ± 0.22**	0.32 ± 0.24	**0.81 ± 0.18**	**0.81 ± 0.18**

Table 4.6: Average precision-alert rate in [0.00, 0.01] ($APAR_{[0.00,0.01]}$). Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Dashes (-) indicate the sets that we cannot create due to low output dimensionality.

AUPRC	Outlier dimensionality = 2.5%					Outlier dimensionality = 5.0%				
	Baselines		MCODE			Baselines		MCODE		
	LOF	I-PROD	M-PROD	M-RW	M-LRW	LOF	I-PROD	M-PROD	M-RW	M-LRW
Mediamill	0.14 ± 0.11	0.07 ± 0.03	0.21 ± 0.08	**0.54 ± 0.10**	**0.58 ± 0.10**	0.18 ± 0.12	0.02 ± 0.01	0.48 ± 0.09	**0.81 ± 0.08**	**0.86 ± 0.08**
Yahoo-business	0.01 ± 0.00	0.05 ± 0.01	0.07 ± 0.03	**0.20 ± 0.04**	**0.22 ± 0.03**	0.01 ± 0.00	0.10 ± 0.03	0.20 ± 0.06	**0.38 ± 0.07**	**0.40 ± 0.06**
Yahoo-arts	-	-	-	-	-	0.01 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	**0.09 ± 0.03**	**0.10 ± 0.03**
Bibtex	0.01 ± 0.00	**0.24 ± 0.24**	**0.28 ± 0.24**	**0.25 ± 0.24**	**0.28 ± 0.25**	0.01 ± 0.00	**0.38 ± 0.25**	**0.40 ± 0.25**	**0.39 ± 0.26**	**0.42 ± 0.25**
Enron	0.02 ± 0.02	0.07 ± 0.15	0.08 ± 0.09	0.07 ± 0.09	0.08 ± 0.07	0.02 ± 0.02	0.09 ± 0.15	0.13 ± 0.14	0.16 ± 0.18	0.17 ± 0.22
Yeast	-	-	-	-	-	-	-	-	-	-
Birds	-	-	-	-	-	0.03 ± 0.01	**0.17 ± 0.15**	**0.21 ± 0.17**	**0.24 ± 0.15**	**0.24 ± 0.16**
Cal500	0.03 ± 0.01	0.03 ± 0.02	0.08 ± 0.07	**0.35 ± 0.21**	**0.23 ± 0.18**	0.16 ± 0.10	0.11 ± 0.09	0.14 ± 0.14	**0.45 ± 0.22**	**0.27 ± 0.15**

AUPRC	Outlier dimensionality = 10.0%					Outlier dimensionality = 20.0%				
	Baselines		MCODE			Baselines		MCODE		
	LOF	I-PROD	M-PROD	M-RW	M-LRW	LOF	I-PROD	M-PROD	M-RW	M-LRW
Mediamill	0.26 ± 0.12	**0.85 ± 0.07**	**0.84 ± 0.08**	**0.93 ± 0.05**	**0.92 ± 0.06**	0.34 ± 0.06	**0.95 ± 0.04**	**0.95 ± 0.04**	**0.96 ± 0.02**	**0.96 ± 0.02**
Yahoo-business	0.01 ± 0.00	0.14 ± 0.04	0.22 ± 0.07	**0.42 ± 0.05**	**0.44 ± 0.06**	0.01 ± 0.00	0.16 ± 0.06	0.12 ± 0.03	**0.36 ± 0.03**	**0.34 ± 0.04**
Yahoo-arts	0.01 ± 0.00	0.09 ± 0.02	0.09 ± 0.02	**0.23 ± 0.03**	**0.24 ± 0.04**	0.01 ± 0.00	0.13 ± 0.02	0.12 ± 0.03	**0.23 ± 0.03**	**0.26 ± 0.04**
Enron	0.02 ± 0.02	**0.12 ± 0.16**	**0.17 ± 0.15**	**0.26 ± 0.21**	**0.30 ± 0.22**	0.02 ± 0.01	0.20 ± 0.22	0.29 ± 0.15	**0.51 ± 0.21**	**0.63 ± 0.18**
Bibtex	0.01 ± 0.00	**0.58 ± 0.24**	**0.56 ± 0.22**	**0.59 ± 0.24**	**0.59 ± 0.22**	0.01 ± 0.00	**0.78 ± 0.16**	**0.75 ± 0.17**	**0.78 ± 0.17**	**0.77 ± 0.16**
Yeast	0.04 ± 0.01	0.02 ± 0.00	0.29 ± 0.07	**0.42 ± 0.04**	**0.42 ± 0.04**	0.06 ± 0.01	0.06 ± 0.04	**0.30 ± 0.06**	**0.37 ± 0.05**	**0.37 ± 0.06**
Birds	0.05 ± 0.03	**0.29 ± 0.24**	**0.40 ± 0.24**	**0.47 ± 0.20**	**0.47 ± 0.20**	0.12 ± 0.09	**0.51 ± 0.25**	**0.62 ± 0.19**	**0.70 ± 0.13**	**0.71 ± 0.14**
Cal500	**0.54 ± 0.20**	**0.35 ± 0.25**	0.21 ± 0.20	**0.51 ± 0.17**	**0.43 ± 0.27**	**0.88 ± 0.05**	**0.78 ± 0.21**	0.20 ± 0.13	0.71 ± 0.16	0.63 ± 0.22

Table 4.7: Area under the precision-recall curve. Numbers shown in bold indicate the best results on each experiment set (by paired t-test at α=0.05). Dashes (-) indicate the sets that we cannot create due to low output dimensionality.

conditional outliers. In most experiments, its PAR is close to zero because it seeks unusual data patterns in the joint space of all attributes. One notable exception is on the results of *Mediamill* and *Cal500*, where the PARs of LOF are recorded unusually high. This is because the datasets have high-dimensional output (Table 4.5) and thus a perturbation in the output space can make the resultant conditional outliers obtrusive and noticeable even to the unconditional method.

Comparing the performance of M-PROD to I-PROD, M-PROD outruns I-PROD more often than vice versa. Recalling that the only difference between M-PROD and I-PROD is the type of data model they use, this verifies the advantages of adopting the decomposable probabilistic data model that is able to capture the conditional dependences among different output variables (Equation (4.15)). To account for the intervals where I-PROD rises (*i.e.*, certain intervals of alert rate on *Yahoo-arts*, *Enron*, and *Cal500*), we conjecture that the datasets do not have strong dependences in the output space and therefore that the data model used in M-PROD is less accurate than the model used in I-PROD.[5] (However, as the next paragraph discusses, M-RW and M-LRW can recover this inaccuracy of M-PROD.)

To validate our outlier scores with reliability weighting, we compare the performance of M-RW and M-LRW to that of M-PROD. Recall that all three methods are utilizing the same data representation; *i.e.*, the difference is only in the way they compute the outlier score. The figures show that, in many experiment sets, M-RW and M-LRW (methods with reliability weighting) improve PAR over M-PROD. In particular, the results on *Yahoo-arts*, *Enron*, and *Cal500* well illustrate the advantages of reliability weighting in that M-RW and M-LRW are able to recover the low PAR of M-PROD and demonstrate the best MCOD performance. Table 4.6 further shows that M-RW and M-LRW are not only capable of improving the outlier detection performance, but are also able to make PARs more consistent (the standard deviation in PAR decreases with reliability weighting). All in all, the results support that our proposed method can effectively assess the quality of each base model, and the resulting weights are useful for multivariate conditional outlier scoring.

Lastly, we compare the performance of M-LRW and M-RW and highlight the advantages

[5]Provided that there are no strong dependences in the output space, conditioning with other output variables (Equation (4.15)) would be analogous to adding noise to the model.

of the local outlier score. The results show that M-LRW can further improve PAR over M-RW using the reliability weights computed locally. Although the local outlier score does not always increase PAR, in several sets of experiments our local approach drastically improves the performance (*e.g.*, *Mediamill* with outlier dimensionality = 2.5 and 5%; *Yahoo-arts* with outlier dimensionality = 20%; *Enron* with outlier dimensionality = 10 and 20%; *Bibtex* with outlier dimensionality = 2.5 and 5%).

(2) *Area Under the Precision-Recall Curve* Table 4.7 shows the results in terms of AUPRC. Again, the numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set. Generally, the performance in AUPRC agrees with what we analyzed above. As with PAR, M-RW and M-LRW outperform the rest and produce the best results across all experiment sets. This confirms that our proposed approaches do not sacrifice recall to gain greater PAR but do well to balance them.

The other methods also show similar performance patterns as the above. AUPRC of LOF is very low in most cases. This conforms to the previous observation that the approach is not effective in addressing the conditional outlier problem. Comparing M-PROD to I-PROD, M-PROD results in better AUPRC in most experiment sets. Although M-PROD underperforms I-PROD on *Yahoo-business*, *Yahoo-arts* with outlier dimensionality = 20%, and *Cal500* with outlier dimensionality = 10 and 20%, M-RW and M-LRW are able to recover such performance drops through reliability weighting.

4.3.2 Multivariate Conditional Outlier Detection with Ratio-based Outlier Scoring

In this section, we explore the extension of our decomposable multivariate conditional outlier score schema to support a collection of univariate ratio-based outlier scoring methods (ROS and ROS-DP; Sections 4.2.2.1 and 4.2.2.3). Recall that ROS methods were introduced to complement probabilistic conditional outlier detection methods for univariate outputs (see Sections 4.2.2.1-4.2.2.3).

Briefly, ROS measures the relative unusualness of an input pattern conditioned on its output value. More specifically, it is defined as a ratio between two (unconditional) outlier

scores: one assessed within instances having the same output value and another assessed within instances having different output values.

Now consider the general multivariate conditional outlier decomposition schema, which can be written as follows.

$$Score_{\text{MPROB}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} Score_{\text{PROB}(i)}\left(y_i^{(n)}\middle|\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right) \tag{4.23}$$

Assuming the same decomposition, we can define the multivariate ROS schema $Score_{\text{ROS-M}}(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ by substituting $Score_{\text{PROB}(i)}$ for each output dimension i in the score with:

$$Score_{\text{ROS-M}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} Score_{\text{ROS}(i)}\left(y_i^{(n)}\middle|\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right) \tag{4.24}$$

$$= \sum_{i=1}^{d} \frac{o_U(\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)}); D_{\text{Agree}(i,n)})}{o_U(\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)}); D_{\text{Disagree}(i,n)})} \tag{4.25}$$

where $\boldsymbol{\pi}(y_i^{(n)})$ denotes the values of the parents of $y_i^{(n)}$ and

$D_{\text{Agree}(i,n)} = \left\{\mathbf{x}^*, \boldsymbol{\pi}(y_i^*)\middle| y_i^* = y_i^{(n)}\right\}$ A subset of D whose output value is equal to $y_i^{(n)}$ ($D_{\text{Agree}(i,n)}$ does not include $(\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)}))$)

$D_{\text{Disagree}(i,n)} = \left\{\mathbf{x}^*, \boldsymbol{\pi}(y_i^*)\middle| y_i^* \neq y_i^{(n)}\right\}$ A subset of D whose output value is not equal to $y_i^{(n)}$

Please note that the above decomposition considers the model with chain dependences and its decomposition. Equivalently, we can replace the chain dependency with the circular dependency decomposition (*i.e.*, $\boldsymbol{\pi}(Y_i) := \mathbf{Y}_{-i}$).

Also, note that ROS is itself a meta-score in that it allows a choice of a large spectrum of unconditional outlier scores. In all experiments in this section, we have adopted the local outlier factor (LOF) [Breunig et al., 2000], a nonparametric method that examines data for unconditional outliers using a relative local density measure (see Sections 2.2.1.2 and 4.2.2.2 for details), as our unconditional outlier score.

Finally, we note that although the conditioning set for each output dimension in ROS-M includes both inputs and other outputs, attributes that do not contribute to the prediction can be regularized out when learning the model from data and hence reduce complexity of the individual models. Below we discuss and formalize such a regularization step as a part of the ROS schema.

4.3.2.1 Ratio of Outlier Scores on Multi-dimensional Discriminative Projections

The ROS approach relies on unconditional outlier scores defined on the input space. For MCOD decompositions the input space is defined by the original input space, as well as, outputs that model dependences and follow the respective decompositions (either via chain or cicular decompositions). The problem with the new input space is that it may become very complex and methods for controling its complexity are needed. In Section 4.2.2.3 we introduced the univariate ROS-DP that lets us control the complexity of the input space via two mechanisms: regularization, and dimensionality reduction via discriminative projections. Hence, following Section 4.2.2.3, we present our last variant of the ROS score that works for data with multi-dimensional output.

The main idea is to employ a supervised discriminative function on top of the ROS-M framework. We define a set of discriminative projection functions $\{f_1, ..., f_d\}$ and their projections $\{\phi_1, ..., \phi_d\}$, such that we specify a projection function for each output dimension.

$$f_i : \left(\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right) \rightarrow \phi_i^{(n)} \tag{4.26}$$

To obtain such functions and projection, we train (learn) a model from the input-output instances in D. Note that the resulting projection also coincides with our probabilistic data representation discussed in Section 4.3.1.3.

To incorporate the discriminative projections with the ROS framework, we project the original data to the d-dimensional discriminative space (Equation (4.9)). We then compute the ROS score on the new projected space with respect to individual output dimension. Again, these steps result in a d-dimensional outlier score vector. We use the general multivariate condition outlier schema to compute the final outlier score. We refer to this score and the associated approach as *Ratio of Outlier Scores on Multi-dimensional Discriminative Projections* (ROS-MDP) approach. ROS-MDP can be written as below, by defining ROS-MDP as the sum over d ROS scores:

$$Score_{\text{ROS-MDP}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} Score_{\text{ROS-DP}(i)}\left(y_i^{(n)}\middle|(\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})), f_i\right) \tag{4.27}$$

$$= \sum_{i=1}^{d} Score_{\text{ROS}(i)}\left(y_i^{(n)}\middle| f_i\left(\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right)\right) \tag{4.28}$$

4.3.2.2 Alternative Multivariate Conditional Outlier Scoring Approaches

The MLC-based decompositions considered so far allowed us to define the multivariate conditional outlier score as a sum of univariate conditional scores. In general, the decomposition can be written as:

$$Score_{\text{MCOD}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} Score_{\text{UCOD}(i)}\left(y_i^{(n)}\middle|\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right)$$

where $Score_{\text{MCOD}}$ and $Score_{\text{UCOD}}$ represent multivariate and univariate conditional outlier score, respectively. $\boldsymbol{\pi}(Y_i)$ is defined either based on chain-rule or circular dependences.

The above schema assumes all univariate scores contribute equally to the final multivariate score. However, if there is any reason to prefer any output dimensions, this generalized multivariate conditional outlier decomposition can be easily extended to bias the score towards the different output dimensions using weighting:

$$Score_{\text{MCOD}_{\text{W}}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \sum_{i=1}^{d} w_i^{(n)}\ Score_{\text{UCOD}(i)}\left(y_i^{(n)}\middle|\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right)$$

The advantage of this new schema is that it remains decomposable to individual univariate conditional scores. Recall that we have used this approach in Section 4.3.1.4 when exploring probabilistic reliability weighting.

Finally, we note that decomposable MCOD scoring can be extended also to cover other types of scoring biases. For example, when we assume the outliers can occur in just one or a very few dimensions it may be appropriate to define the MCOD score by maximizing over the individual UCOD scores:

$$Score_{\text{MCOD}_{\text{MAX}}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) = \max_{i\in\{1,\ldots,d\}} Score_{\text{UCOD}(i)}\left(y_i^{(n)}\middle|\mathbf{x}^{(n)}, \boldsymbol{\pi}(y_i^{(n)})\right)$$

Briefly, unlike the standard MCOD scheme that weights all dimensions equally when defining the outliers, the max score focuses on the worst (or maximum) outlier score.

4.3.2.3 Experiments To validate and demonstrate the performance of our approach, we conduct experiments with synthetic and public datasets with simulated conditional outliers. Through the empirical analysis in this section, we would like to verify the advantages of ROS-M and ROS-MDP.

Tested Methods We compare the following outlier detection methods:

- *Local Outlier Factor for the Joint Input-Output Space* (LOF) [Breunig et al., 2000] – LOF is an unconditional method that estimates outliers using a relative local density measure and helps to find instances that fall in sparse regions of data. Here we apply it to the joint space of all data attributes covering both inputs and outputs.
- *Probability as Outlier Score* (PROB) – Solves the multivariate conditional outlier detection problem by considering d dependent conditional probability models (with circular dependences) and probabilistic UCOD scores defined on these models. (Equation (4.17))
- *Ratio of Outlier Scores on Multi-dimensional Output* (ROS-M) – Solves the multivariate conditional outlier detection problem by considering d dependent conditional probability models (with circular dependences) and ratio of outliers UCOD scores defined on these models.
- *Ratio of Outlier Scores on Multi-dimensional Discriminative Projections* (ROS-MDP) – Solves the multivariate conditional outlier detection problem by considering d dependent conditional probability models (with circular dependences) and ratio of outliers UCOD scores with discriminative projections (based on the logistic regression) defined on these models.

To obtain data models/discriminative projections in PROB and ROS-MDP, we use L_2-regularized logistic regression and choose their regularization parameters by cross validation. In LOF, ROS-M, and ROS-MDP, we set the number of neighbors $k = 50$.

Recall that PROB, ROS-M, and ROS-MDP require a multivariate function that combines multiple outlier scores into one. In our experiments, we use *max* and *average* operations. Accordingly, we use the following labels for the tested methods: PROB_{MAX}, PROB_{SUM}, $\text{ROS-M}_{\text{MAX}}$, $\text{ROS-M}_{\text{SUM}}$, $\text{ROS-MDP}_{\text{MAX}}$, and $\text{ROS-MDP}_{\text{SUM}}$.

Experiment Setup We continue to use the same experiment setup as in the previous section, with small modifications in *outlier dimensionality*. To highlight, in our experimental study, we simulate outliers as:

1. In each simulation, select [*outlier ratio*]% of instances uniformly at random
2. For each of the selected instances, perturb the values in [*outlier dimensionality*]% of the output dimensions uniformly at random ($y_{\text{perturbed}} = |y_{\text{original}} - 1|$)

Note that, in all experiments in this section, we fix *outlier ratio* to 1.0%. See Section 4.3.1.5 for detailed description on the experiment setup.

Evaluation Metrics We use the *precision-alert rate* (PAR) curves, *average PAR* (APAR) in $[0.00, 0.01]$ range, and *area under the precision-recall curve* (AUPRC) as our evaluation metric. For all three metrics, higher is better. See Section 4.2.3 for detailed description of the metrics.

4.3.2.3.1 Synthetic Datasets We first conduct experiments on two synthetic datasets, *SD5* and *SD6* (Figure 4.8). Each dataset contains 1,000 instances with *two*-dimensional output data ($d = 2$); each random variable Y_i is generated by the Bernoulli distribution.

$$y_i \sim Bern(\theta); \qquad \theta = 0.3 \;\; (i = 1, 2)$$

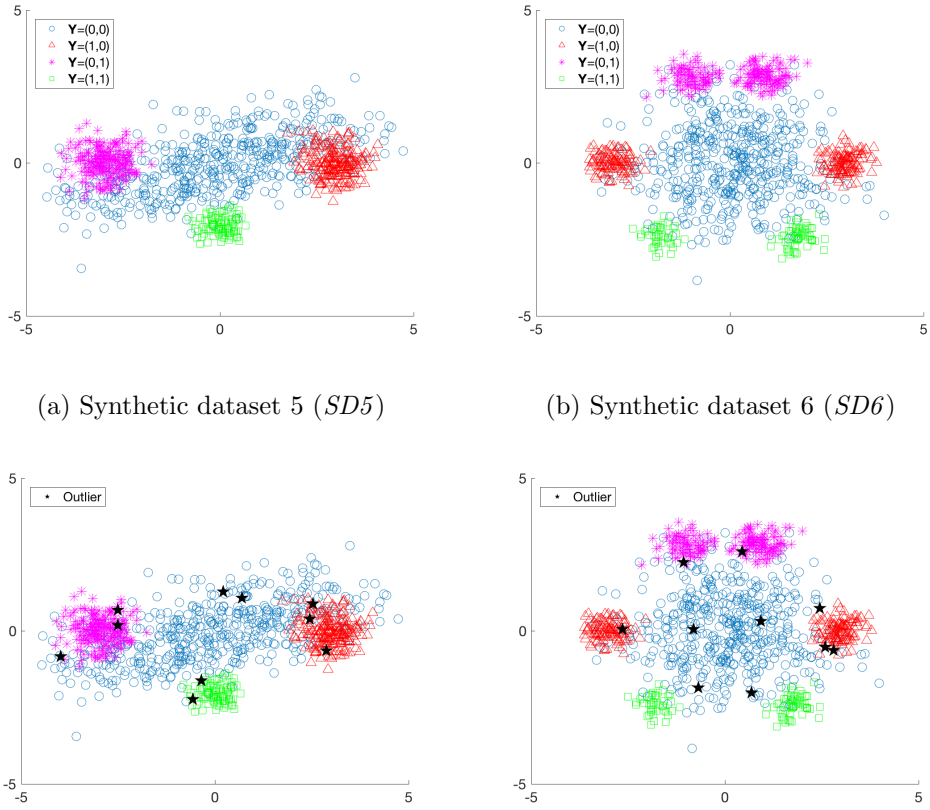
Now we create $\{2, 5, 10, 30\}$-dimensional input data. Depending on the values of $\mathbf{Y}$, the first *two* dimensions of input X_1, X_2 are generated by multivariate Gaussian.

$$x_1, x_2 \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are determined based on the values of $\mathbf{Y}$. The actual parameter values used for the data generation of *SD5* and *SD6* are listed in Table 4.8. Note that this process defines nonlinear discriminative boundaries among different output values.

For the rest of the input dimensions, we generate some values that are irrelevant to the output. This adds noisy attributes to the input space.

$$x_3, ..., x_m \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}'); \qquad \boldsymbol{\mu}' = \mathbf{0}, \;\; \boldsymbol{\Sigma}' = 0.25 \cdot \sqrt{m} \cdot I_{(m-2)}$$



(a) Synthetic dataset 5 (*SD5*)

(b) Synthetic dataset 6 (*SD6*)

(c) *SD5* with outliers

(d) *SD6* with outliers

Figure 4.8: Synthetic datasets 5, 6 (*SD5* and *SD6*; the first row) and example conditional outliers (marked with a star; the second row).

Simulated Outliers Each experiment run contains 1.0% of conditional outliers. We use *outlier dimensionality* $= \{50.0, 100.0\}$; *i.e.*, one or two output dimensions are perturbed for outliers.

Results Figures 4.9-4.10 and Tables 4.9-4.10 present the results of the tested methods on the three synthetic datasets, *SD5* and *SD6*. The results are averages over *five* repetitions of outlier simulations.

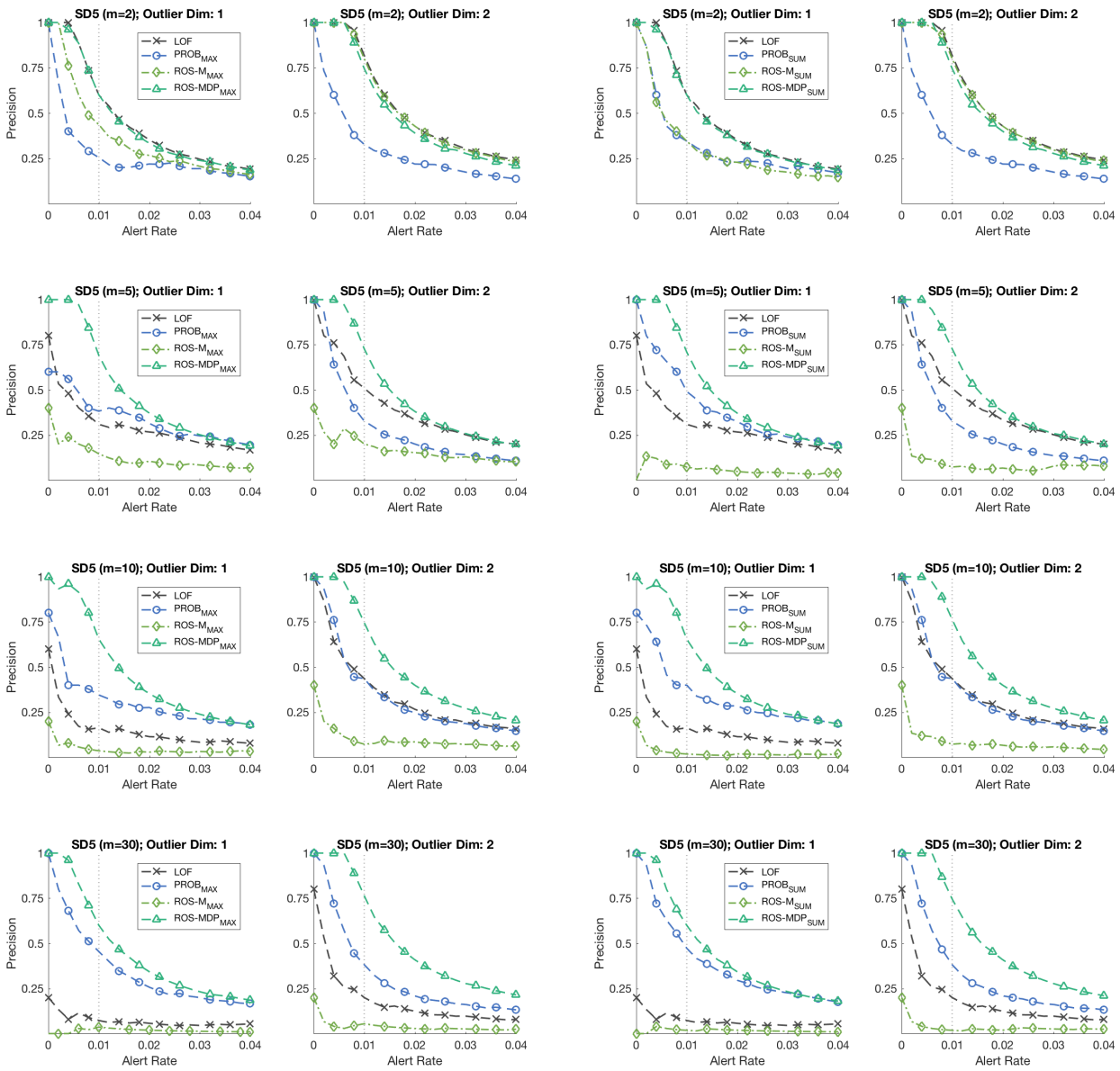
(1) *Precision-Alert Rate* Figures 4.9 and 4.10 show the precision of the tested methods at different alert rates (precision-alert rate (PAR) curves) ranging between 0.00 and 0.04,

	$\mathbf{Y} = \{0, 0\}$	$\mathbf{Y} = \{0, 1\}$	$\mathbf{Y} = \{1, 0\}$	$\mathbf{Y} = \{1, 1\}$
SD5	$\boldsymbol{\mu} = \begin{bmatrix} 0 & 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 5 & 1.5 \\ 1.5 & 1 \end{bmatrix}$	$\boldsymbol{\mu} = \begin{bmatrix} 3 & 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$	$\boldsymbol{\mu} = \begin{bmatrix} -3 & 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$	$\boldsymbol{\mu} = \begin{bmatrix} 0 & -2 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$
SD6	$\boldsymbol{\mu} = \begin{bmatrix} 0 & 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}$	Let $U \sim unif(0, 1)$; if $U < 0.5$: $\boldsymbol{\mu} = \begin{bmatrix} 3 & 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ otherwise: $\boldsymbol{\mu} = \begin{bmatrix} -3 & 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	Let $U \sim unif(0, 1)$; if $U < 0.5$: $\boldsymbol{\mu} = \begin{bmatrix} 0.91 & 2.86 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ otherwise: $\boldsymbol{\mu} = \begin{bmatrix} -0.91 & 2.86 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	Let $U \sim unif(0, 1)$; if $U < 0.5$: $\boldsymbol{\mu} = \begin{bmatrix} -1.74 & -2.44 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ otherwise: $\boldsymbol{\mu} = \begin{bmatrix} 1.74 & -2.44 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

Table 4.8: Parameters for the data generation of *SD5* and *SD6*.

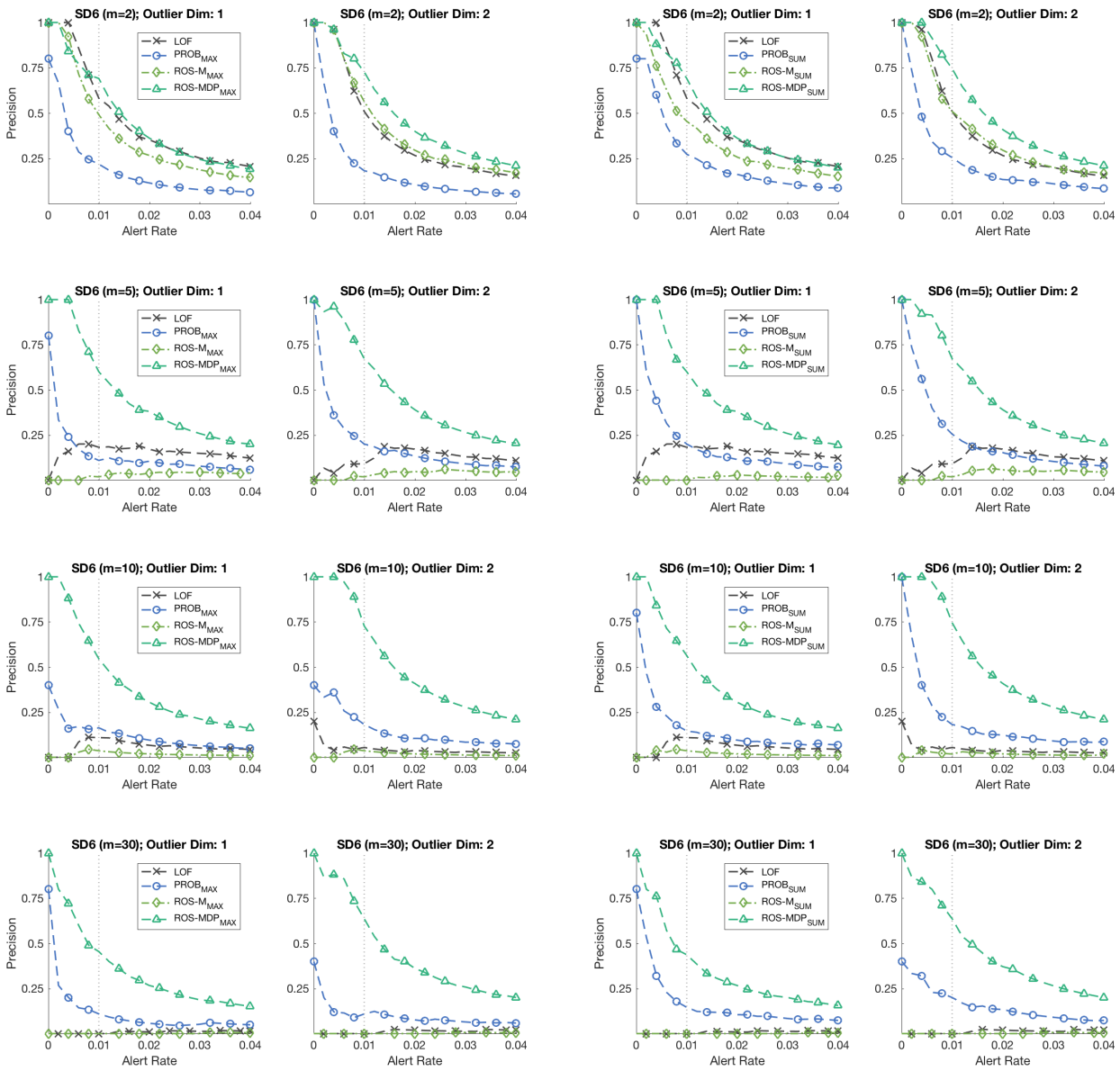
on *SD5* and *SD6*, respectively. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the ratio of simulated outliers. Notice that, for each dataset, the PAR curves are shown in two groups according to the type of the multivariate combine function used by the MCOD methods. Within each group, the curves are organized in such a way that each column shows an outlier dimensionality and each row shows an input dimensionality. In general, PAR improves as the outlier dimensionality increases, because outliers with larger perturbations are easier to detect. Table 4.9 presents the *average PAR* (APAR) in $[0.00, 0.01]$ range. The numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set.

Overall, the PAR curves (Figures 4.9 and 4.10) show that ROS-MDP achieves superior results across all experiments. It maintains the best precision and controllability regardless of the type of combine function (MAX or AVG), input or outlier dimensionality. In terms of APAR over alert rate $[0.00, 0.01]$ (Table 4.9), ROS-MDP also shows significantly better



(a) Combine function: *MAX* (b) Combine function: *SUM*

Figure 4.9: Precision-alert rate (PAR) on *SD5*. Each plot draws PAR at alert rates (detection thresholds) between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the simulated outlier ratio.



(a) Combine function: *MAX* (b) Combine function: *SUM*

Figure 4.10: Precision-alert rate (PAR) on *SD6*. Each plot draws PAR at alert rates (detection thresholds) between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the simulated outlier ratio.

$APAR_{[0.00,0.01]}$		Outlier dimensionality=1						
		LOF	$PROB_{MAX}$	$PROB_{SUM}$	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
SD5	(m=2)	**0.88 ± 0.04**	0.47 ± 0.11	0.58 ± 0.16	0.71 ± 0.11	0.59 ± 0.11	**0.87 ± 0.09**	**0.87 ± 0.10**
	(m=5)	0.46 ± 0.20	0.53 ± 0.29	0.70 ± 0.25	0.22 ± 0.20	0.09 ± 0.13	**0.93 ± 0.04**	**0.94 ± 0.04**
	(m=10)	0.24 ± 0.25	0.48 ± 0.29	0.57 ± 0.29	0.07 ± 0.16	0.05 ± 0.11	**0.89 ± 0.11**	**0.89 ± 0.11**
	(m=30)	0.10 ± 0.13	0.65 ± 0.17	0.72 ± 0.11	0.01 ± 0.03	0.02 ± 0.04	**0.87 ± 0.09**	**0.85 ± 0.10**
SD6	(m=2)	**0.87 ± 0.06**	0.42 ± 0.25	0.54 ± 0.31	0.79 ± 0.05	0.71 ± 0.11	**0.83 ± 0.07**	**0.87 ± 0.06**
	(m=5)	0.16 ± 0.11	0.26 ± 0.17	0.44 ± 0.17	0.01 ± 0.01	0.00 ± 0.00	**0.87 ± 0.06**	**0.86 ± 0.05**
	(m=10)	0.05 ± 0.03	0.21 ± 0.20	0.32 ± 0.24	0.02 ± 0.04	0.03 ± 0.06	**0.81 ± 0.09**	**0.80 ± 0.10**
	(m=30)	0.00 ± 0.00	0.23 ± 0.13	0.34 ± 0.22	0.00 ± 0.00	0.00 ± 0.00	**0.68 ± 0.08**	**0.66 ± 0.07**

$APAR_{[0.00,0.01]}$		Outlier dimensionality=2						
		LOF	$PROB_{MAX}$	$PROB_{SUM}$	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
SD5	(m=2)	**0.98 ± 0.01**	0.57 ± 0.25	0.57 ± 0.25	**0.97 ± 0.01**	**0.96 ± 0.02**	**0.96 ± 0.03**	**0.96 ± 0.03**
	(m=5)	0.72 ± 0.04	0.63 ± 0.17	0.63 ± 0.17	0.26 ± 0.16	0.13 ± 0.19	**0.94 ± 0.04**	**0.93 ± 0.04**
	(m=10)	0.64 ± 0.16	0.68 ± 0.16	0.68 ± 0.16	0.16 ± 0.15	0.13 ± 0.15	**0.94 ± 0.04**	**0.95 ± 0.03**
	(m=30)	0.38 ± 0.11	0.68 ± 0.18	0.68 ± 0.18	0.06 ± 0.13	0.05 ± 0.11	**0.96 ± 0.04**	**0.95 ± 0.03**
SD6	(m=2)	0.83 ± 0.08	0.42 ± 0.17	0.49 ± 0.22	0.84 ± 0.05	0.80 ± 0.11	**0.89 ± 0.03**	**0.92 ± 0.04**
	(m=5)	0.06 ± 0.06	0.39 ± 0.16	0.53 ± 0.22	0.01 ± 0.01	0.01 ± 0.01	**0.88 ± 0.10**	**0.89 ± 0.09**
	(m=10)	0.06 ± 0.13	0.29 ± 0.12	0.43 ± 0.12	0.02 ± 0.04	0.02 ± 0.04	**0.94 ± 0.06**	**0.95 ± 0.05**
	(m=30)	0.00 ± 0.00	0.15 ± 0.21	0.28 ± 0.29	0.00 ± 0.00	0.00 ± 0.00	**0.84 ± 0.08**	**0.82 ± 0.07**

Table 4.9: Average precision-alert rate in [0.00, 0.01] ($APAR_{[0.00,0.01]}$). Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$).

performance than other tested methods in most experiments. This confirms the validity of ROS-MDP in addressing the MCOD problem.

The results also reveal that, although PROB produces relatively consistent results (compared to ROS-M and LOF), there are significant performance gaps between ROS-MDP and PROB (Table 4.9). Given that ROS-MDP and PROB run on essentially identical linear discriminative projection (logistic regression), this signifies that our ROS-MDP framework can effectively deal with conditional outliers, especially on data that form nonlinear decision boundaries. Note that this capability comes from the base unconditional outlier detection method (LOF). In other words, ROS-MDP can recover the discriminability that the linear projection may lose, by locally examining data as the base LOF method constitutes.

Compared to ROS-MDP and PROB, LOF and ROS-M appear to suffer severely from increasing input dimensionality. More specifically, LOF performs reasonably well when $d = 2$; this suggests that, when the input dimensionality is low, unconditional outlier detection methods still can identify conditional outliers. However, when $d \geq 5$, the performance of LOF

AUPRC		Outlier dimensionality=1						
		LOF	PROB_{MAX}	PROB_{SUM}	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
SD5	(m=2)	**0.62 ± 0.09**	0.24 ± 0.07	0.33 ± 0.14	0.41 ± 0.12	0.29 ± 0.09	**0.60 ± 0.12**	**0.60 ± 0.13**
	(m=5)	0.27 ± 0.16	0.38 ± 0.16	0.47 ± 0.19	0.08 ± 0.07	0.05 ± 0.05	**0.69 ± 0.09**	**0.69 ± 0.10**
	(m=10)	0.10 ± 0.09	0.31 ± 0.19	**0.38 ± 0.21**	0.03 ± 0.03	0.02 ± 0.01	**0.63 ± 0.22**	**0.63 ± 0.22**
	(m=30)	0.03 ± 0.01	0.38 ± 0.16	**0.45 ± 0.15**	0.02 ± 0.01	0.02 ± 0.01	**0.60 ± 0.15**	**0.58 ± 0.16**
SD6	(m=2)	**0.63 ± 0.10**	0.16 ± 0.09	0.25 ± 0.15	0.43 ± 0.06	0.39 ± 0.09	**0.59 ± 0.09**	**0.62 ± 0.09**
	(m=5)	0.12 ± 0.03	0.07 ± 0.04	0.14 ± 0.11	0.03 ± 0.02	0.02 ± 0.02	**0.62 ± 0.11**	**0.61 ± 0.10**
	(m=10)	0.04 ± 0.02	0.06 ± 0.06	0.09 ± 0.09	0.02 ± 0.02	0.02 ± 0.02	**0.51 ± 0.16**	**0.50 ± 0.16**
	(m=30)	0.01 ± 0.00	0.03 ± 0.02	0.11 ± 0.10	0.01 ± 0.00	0.01 ± 0.00	**0.36 ± 0.10**	**0.35 ± 0.10**

AUPRC		Outlier dimensionality=2						
		LOF	PROB_{MAX}	PROB_{SUM}	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
SD5	(m=2)	**0.82 ± 0.05**	0.30 ± 0.21	0.30 ± 0.21	**0.81 ± 0.03**	**0.80 ± 0.03**	**0.74 ± 0.09**	**0.75 ± 0.09**
	(m=5)	0.48 ± 0.07	0.31 ± 0.21	0.31 ± 0.21	0.12 ± 0.05	0.07 ± 0.04	**0.71 ± 0.12**	**0.70 ± 0.12**
	(m=10)	0.36 ± 0.16	0.38 ± 0.14	0.38 ± 0.14	0.06 ± 0.04	0.05 ± 0.04	**0.72 ± 0.07**	**0.74 ± 0.06**
	(m=30)	0.11 ± 0.04	0.35 ± 0.21	0.36 ± 0.21	0.02 ± 0.02	0.02 ± 0.01	**0.76 ± 0.11**	**0.75 ± 0.11**
SD6	(m=2)	0.49 ± 0.11	0.13 ± 0.10	0.19 ± 0.14	0.53 ± 0.06	0.49 ± 0.09	**0.70 ± 0.09**	**0.73 ± 0.10**
	(m=5)	0.10 ± 0.05	0.12 ± 0.10	0.21 ± 0.17	0.04 ± 0.02	0.04 ± 0.02	**0.68 ± 0.14**	**0.69 ± 0.14**
	(m=10)	0.03 ± 0.02	0.08 ± 0.03	0.15 ± 0.08	0.02 ± 0.02	0.02 ± 0.01	**0.74 ± 0.12**	**0.74 ± 0.11**
	(m=30)	0.02 ± 0.00	0.07 ± 0.05	0.14 ± 0.14	0.01 ± 0.00	0.01 ± 0.00	**0.59 ± 0.12**	**0.58 ± 0.11**

Table 4.10: Area under the precision-recall curve. Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$).

significantly degrades because, first of all, finding unusual input-output associations in the higher-dimensional joint space becomes nontrivial; second, LOF does not explicitly handle irrelevant attributes in data. ROS-M shows an even worse performance degradation with high-dimensional input, which concurs with our previous analysis that ROS-M would not show a reasonable performance without a complementary regularization mechanism (Section 4.3.2).

Lastly, the choice of the multivariate combine function yields interesting patterns in results. That is, PROB_{SUM} always outperforms PROB_{MAX}; whereas $\text{ROS-M}_{\text{MAX}}$ often outperforms $\text{ROS-M}_{\text{SUM}}$. This may suggest that there is a proper choice of combine function for each method. On the other hand, ROS-MDP is less affected by the combine function, which could be an advantage as an outlier detection method. We will continue our discussion on this in the experiments on real-world data.

(2) *Area Under the Precision-Recall Curve* Table 4.10 presents the results in terms

Dataset	N / m / d	Domain	Value Description	
			Input	Output
Mediamill	43,907 / 120 / 101	Video	Video frames	Concepts
Yahoo-business	11,214 / 21,924 / 30	Text	News articles	Topics
Yahoo-arts	7,484 / 23,146 / 26	Text	News articles	Topics
Bibtex	7,395 / 1,836 / 159	Text	Paper metadata	Topics
Enron	1,702 / 1,001 / 53	Text	Emails	Properties
Birds	645 / 276 / 19	Sound	Bird songs	Species
Cal500	502 / 68 / 174	Music	Waveforms	Annotations
Yeast	2,417 / 103 / 14	Biology	Genes	Functionalities
Rcv1sub1-top10	6,000 / 8,394 / 10	Text	News articles	Topics
Rcv1sub3-top10	6,000 / 8,328 / 10	Text	News articles	Topics

Table 4.11: Dataset characteristics (N: number of instances, m: input dimensionality, d: output dimensionality).

of AUPRC. Again, the numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set.

In general, the results reported in AUPRC show similar patterns as in PAR/APAR. ROS-MDP reports overall the best AUPRC, followed by PROB, LOF, and ROS-M. This indicates that the ROS-MDP scores achieve a good balance between precision and recall. All in all, the results in AUPRC reaffirm our observations and conclusions from PAR/APAR.

4.3.2.3.2 Public Datasets We use *ten* public datasets with multi-dimensional input and output in our experiments.[6] These are collected from various application domains, including semantic video/image annotation (*Mediamill*), text categorization (*Yahoo* and *Rcv1* datasets, *Bibtex*, and *Enron*), biology (*Yeast*), and sound/music recognition (*Birds* and *Cal500*). Table 4.11 summarizes the characteristics of the datasets, such as dataset size, data domain, and short descriptions of the input and output variables.

Simulated Outliers Each experiment run contains 1.0% of conditional outliers. We use *outlier dimensionality* $= \{5.0, 10.0, 50.0, 100.0\}\%$.

[6]Datasets are available at http://mulan.sourceforge.net [Tsoumakas et al., 2010].

$APAR_{[0.00,0.01]}$	Outlier dimensionality = 5 %						
	LOF	$PROB_{MAX}$	$PROB_{SUM}$	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.02 ± 0.04	**0.13 ± 0.13**	0.07 ± 0.02	0.02 ± 0.04	0.02 ± 0.04	**0.13 ± 0.08**	**0.14 ± 0.05**
Yahoo-business	0.02 ± 0.05	0.03 ± 0.05	0.00 ± 0.00	0.03 ± 0.05	0.02 ± 0.03	**0.47 ± 0.10**	**0.42 ± 0.09**
Yahoo-arts	0.00 ± 0.00	**0.13 ± 0.11**	**0.06 ± 0.08**	0.01 ± 0.02	0.00 ± 0.00	**0.10 ± 0.09**	**0.04 ± 0.07**
Bibtex	0.00 ± 0.00	**0.11 ± 0.15**	**0.09 ± 0.16**	0.00 ± 0.00	0.00 ± 0.00	**0.22 ± 0.17**	**0.25 ± 0.25**
Enron	0.00 ± 0.00	**0.10 ± 0.11**	**0.17 ± 0.16**	0.00 ± 0.00	0.00 ± 0.00	**0.11 ± 0.05**	**0.15 ± 0.13**
Birds	0.04 ± 0.06	**0.43 ± 0.26**	0.16 ± 0.13	0.00 ± 0.00	0.01 ± 0.01	0.06 ± 0.06	0.15 ± 0.11
Cal500	0.00 ± 0.00	0.30 ± 0.16	**0.44 ± 0.17**	0.00 ± 0.00	0.00 ± 0.00	**0.49 ± 0.14**	**0.49 ± 0.14**
Yeast	-	-	-	-	-	-	-
Rcv1sub1-top10	-	-	-	-	-	-	-
Rcv1sub3-top10	-	-	-	-	-	-	-
$APAR_{[0.00,0.01]}$	Outlier dimensionality = 10 %						
	LOF	$PROB_{MAX}$	$PROB_{SUM}$	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.02 ± 0.04	0.23 ± 0.15	0.15 ± 0.06	0.02 ± 0.04	0.02 ± 0.04	**0.49 ± 0.07**	**0.59 ± 0.11**
Yahoo-business	0.00 ± 0.00	0.01 ± 0.03	0.01 ± 0.01	0.03 ± 0.05	0.02 ± 0.05	**0.49 ± 0.13**	**0.42 ± 0.12**
Yahoo-arts	0.00 ± 0.00	**0.13 ± 0.12**	**0.07 ± 0.10**	0.01 ± 0.02	0.00 ± 0.00	**0.12 ± 0.11**	**0.06 ± 0.04**
Bibtex	0.00 ± 0.00	**0.13 ± 0.19**	**0.10 ± 0.18**	0.00 ± 0.00	0.00 ± 0.00	**0.24 ± 0.20**	**0.31 ± 0.25**
Enron	0.00 ± 0.00	**0.15 ± 0.20**	**0.28 ± 0.24**	0.00 ± 0.00	0.00 ± 0.00	**0.13 ± 0.11**	**0.08 ± 0.13**
Birds	0.16 ± 0.12	**0.38 ± 0.23**	**0.61 ± 0.19**	0.00 ± 0.00	0.00 ± 0.00	0.09 ± 0.07	0.14 ± 0.15
Cal500	0.00 ± 0.00	0.47 ± 0.05	**0.75 ± 0.13**	0.00 ± 0.00	0.00 ± 0.00	**0.79 ± 0.09**	**0.80 ± 0.11**
Yeast	0.20 ± 0.07	**0.54 ± 0.11**	**0.55 ± 0.10**	0.02 ± 0.04	0.01 ± 0.01	**0.49 ± 0.07**	**0.46 ± 0.05**
Rcv1sub1-top10	0.00 ± 0.00	**0.41 ± 0.17**	0.28 ± 0.22	0.00 ± 0.00	0.00 ± 0.00	**0.52 ± 0.17**	**0.53 ± 0.16**
Rcv1sub3-top10	0.00 ± 0.00	0.58 ± 0.10	0.45 ± 0.13	0.00 ± 0.00	0.04 ± 0.06	**0.74 ± 0.10**	**0.74 ± 0.10**
$APAR_{[0.00,0.01]}$	Outlier dimensionality = 20 %						
	LOF	$PROB_{MAX}$	$PROB_{SUM}$	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.03 ± 0.04	0.47 ± 0.09	0.53 ± 0.07	0.02 ± 0.04	0.02 ± 0.04	**0.81 ± 0.07**	**0.85 ± 0.07**
Yahoo-business	0.01 ± 0.01	0.08 ± 0.08	0.04 ± 0.09	0.03 ± 0.05	0.01 ± 0.03	**0.71 ± 0.17**	0.51 ± 0.17
Yahoo-arts	0.00 ± 0.00	**0.20 ± 0.16**	**0.14 ± 0.11**	0.01 ± 0.02	0.00 ± 0.00	**0.13 ± 0.12**	**0.07 ± 0.07**
Bibtex	0.00 ± 0.00	0.21 ± 0.20	**0.37 ± 0.38**	0.00 ± 0.00	0.00 ± 0.00	**0.60 ± 0.08**	**0.73 ± 0.14**
Enron	0.00 ± 0.00	**0.23 ± 0.13**	**0.46 ± 0.24**	0.00 ± 0.00	0.00 ± 0.00	0.08 ± 0.05	0.00 ± 0.00
Birds	0.17 ± 0.19	0.40 ± 0.12	**0.74 ± 0.22**	0.01 ± 0.02	0.00 ± 0.00	0.07 ± 0.07	0.14 ± 0.12
Cal500	0.01 ± 0.01	0.32 ± 0.25	**0.93 ± 0.06**	0.00 ± 0.00	0.00 ± 0.00	**0.92 ± 0.06**	**0.90 ± 0.08**
Yeast	0.27 ± 0.05	**0.55 ± 0.03**	**0.58 ± 0.05**	0.02 ± 0.04	0.00 ± 0.00	0.38 ± 0.06	0.34 ± 0.03
Rcv1sub1-top10	0.00 ± 0.00	0.51 ± 0.19	0.47 ± 0.17	0.00 ± 0.00	0.00 ± 0.00	**0.73 ± 0.10**	**0.71 ± 0.09**
Rcv1sub3-top10	0.00 ± 0.00	**0.74 ± 0.09**	**0.71 ± 0.08**	0.00 ± 0.00	0.03 ± 0.06	**0.74 ± 0.09**	**0.76 ± 0.07**
$APAR_{[0.00,0.01]}$	Outlier dimensionality = 50 %						
	LOF	$PROB_{MAX}$	$PROB_{SUM}$	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.05 ± 0.05	0.67 ± 0.13	**1.00 ± 0.00**	0.02 ± 0.04	0.02 ± 0.04	**0.95 ± 0.04**	**0.99 ± 0.01**
Yahoo-business	0.01 ± 0.02	0.10 ± 0.06	0.30 ± 0.09	0.03 ± 0.05	0.00 ± 0.00	**0.78 ± 0.06**	0.49 ± 0.04
Yahoo-arts	0.00 ± 0.00	**0.46 ± 0.15**	**0.47 ± 0.10**	0.00 ± 0.01	0.00 ± 0.00	**0.40 ± 0.10**	0.28 ± 0.09
Bibtex	0.01 ± 0.03	0.15 ± 0.13	**0.73 ± 0.22**	0.00 ± 0.00	0.00 ± 0.00	0.81 ± 0.06	**0.91 ± 0.05**
Enron	0.00 ± 0.00	**0.22 ± 0.14**	**0.33 ± 0.12**	0.00 ± 0.00	0.00 ± 0.00	**0.33 ± 0.22**	0.00 ± 0.00
Birds	0.23 ± 0.18	0.39 ± 0.16	**0.99 ± 0.01**	0.01 ± 0.01	0.00 ± 0.00	0.22 ± 0.24	0.20 ± 0.09
Cal500	0.03 ± 0.04	0.22 ± 0.21	**0.95 ± 0.04**	0.00 ± 0.00	0.00 ± 0.00	**0.91 ± 0.06**	**0.86 ± 0.10**
Yeast	0.31 ± 0.08	**0.59 ± 0.02**	**0.62 ± 0.03**	0.02 ± 0.06	0.00 ± 0.00	0.26 ± 0.03	0.26 ± 0.04
Rcv1sub1-top10	0.00 ± 0.00	0.63 ± 0.20	**0.78 ± 0.11**	0.01 ± 0.01	0.00 ± 0.00	**0.87 ± 0.07**	**0.88 ± 0.05**
Rcv1sub3-top10	0.00 ± 0.00	0.78 ± 0.09	**0.89 ± 0.08**	0.00 ± 0.00	0.02 ± 0.03	0.71 ± 0.07	0.79 ± 0.05

Table 4.12: Average precision-alert rate in [0.00, 0.01] ($APAR_{[0.00,0.01]}$). Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Dashes (-) indicate the sets that we cannot create due to low output dimensionality.

AUPRC	Outlier dimensionality = 5 %						
	LOF	PROB_{MAX}	PROB_{SUM}	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.02 ± 0.01	**0.04 ± 0.03**	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	**0.07 ± 0.02**	**0.06 ± 0.02**
Yahoo-business	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.00	0.01 ± 0.01	0.01 ± 0.00	**0.20 ± 0.06**	**0.15 ± 0.06**
Yahoo-arts	0.01 ± 0.00	**0.03 ± 0.02**	**0.03 ± 0.01**	0.01 ± 0.00	0.01 ± 0.00	**0.04 ± 0.01**	**0.03 ± 0.01**
Bibtex	0.01 ± 0.01	**0.10 ± 0.13**	**0.10 ± 0.17**	0.01 ± 0.01	0.01 ± 0.00	**0.09 ± 0.08**	**0.12 ± 0.16**
Enron	0.01 ± 0.00	0.07 ± 0.07	0.12 ± 0.12	0.01 ± 0.00	0.01 ± 0.00	0.06 ± 0.04	0.07 ± 0.07
Birds	**0.03 ± 0.01**	**0.23 ± 0.21**	**0.14 ± 0.06**	0.02 ± 0.01	0.02 ± 0.01	**0.06 ± 0.03**	**0.07 ± 0.04**
Cal500	0.02 ± 0.00	0.16 ± 0.09	0.18 ± 0.05	0.01 ± 0.00	0.01 ± 0.00	**0.29 ± 0.07**	**0.27 ± 0.09**
Yeast	-	-	-	-	-	-	-
Rcv1sub1-top10	-	-	-	-	-	-	-
Rcv1sub3-top10	-	-	-	-	-	-	-

AUPRC	Outlier dimensionality = 10 %						
	LOF	PROB_{MAX}	PROB_{SUM}	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.03 ± 0.01	0.11 ± 0.06	0.08 ± 0.03	0.01 ± 0.00	0.01 ± 0.00	**0.31 ± 0.08**	**0.34 ± 0.10**
Yahoo-business	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	**0.22 ± 0.07**	**0.17 ± 0.08**
Yahoo-arts	0.01 ± 0.00	**0.05 ± 0.04**	**0.05 ± 0.02**	0.01 ± 0.00	0.01 ± 0.00	**0.06 ± 0.04**	**0.06 ± 0.03**
Bibtex	0.01 ± 0.01	0.11 ± 0.16	0.10 ± 0.18	0.01 ± 0.01	0.01 ± 0.00	0.10 ± 0.09	0.15 ± 0.17
Enron	0.01 ± 0.00	**0.10 ± 0.10**	**0.18 ± 0.17**	0.01 ± 0.00	0.01 ± 0.00	**0.07 ± 0.04**	**0.06 ± 0.05**
Birds	0.06 ± 0.03	**0.22 ± 0.20**	**0.38 ± 0.18**	0.02 ± 0.01	0.02 ± 0.01	0.08 ± 0.04	0.08 ± 0.07
Cal500	0.04 ± 0.01	0.19 ± 0.03	**0.50 ± 0.15**	0.01 ± 0.00	0.01 ± 0.00	**0.60 ± 0.13**	**0.61 ± 0.14**
Yeast	0.05 ± 0.02	**0.34 ± 0.09**	**0.33 ± 0.08**	0.01 ± 0.00	0.01 ± 0.00	0.25 ± 0.05	0.23 ± 0.04
Rcv1sub1-top10	0.01 ± 0.00	0.14 ± 0.10	0.10 ± 0.08	0.01 ± 0.00	0.01 ± 0.00	**0.27 ± 0.11**	**0.27 ± 0.10**
Rcv1sub3-top10	0.01 ± 0.00	0.26 ± 0.09	0.19 ± 0.07	0.01 ± 0.00	0.01 ± 0.01	**0.46 ± 0.14**	**0.46 ± 0.13**

AUPRC	Outlier dimensionality = 20 %						
	LOF	PROB_{MAX}	PROB_{SUM}	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.06 ± 0.01	0.25 ± 0.06	0.43 ± 0.06	0.01 ± 0.00	0.01 ± 0.00	**0.66 ± 0.09**	**0.72 ± 0.05**
Yahoo-business	0.01 ± 0.00	0.03 ± 0.01	0.02 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	**0.48 ± 0.16**	**0.38 ± 0.13**
Yahoo-arts	0.01 ± 0.00	**0.06 ± 0.04**	**0.06 ± 0.03**	0.01 ± 0.01	0.01 ± 0.00	**0.07 ± 0.02**	**0.06 ± 0.02**
Bibtex	0.01 ± 0.01	0.19 ± 0.15	**0.30 ± 0.32**	0.01 ± 0.01	0.01 ± 0.00	**0.36 ± 0.09**	**0.47 ± 0.18**
Enron	0.01 ± 0.00	**0.11 ± 0.09**	**0.28 ± 0.26**	0.01 ± 0.00	0.01 ± 0.00	**0.05 ± 0.02**	0.03 ± 0.00
Birds	0.09 ± 0.06	0.22 ± 0.12	**0.55 ± 0.21**	0.02 ± 0.01	0.02 ± 0.01	0.09 ± 0.03	0.10 ± 0.06
Cal500	0.07 ± 0.01	0.15 ± 0.11	**0.78 ± 0.11**	0.01 ± 0.00	0.01 ± 0.00	**0.77 ± 0.09**	**0.75 ± 0.10**
Yeast	0.07 ± 0.02	**0.32 ± 0.04**	**0.34 ± 0.05**	0.01 ± 0.00	0.01 ± 0.00	0.21 ± 0.02	0.16 ± 0.03
Rcv1sub1-top10	0.01 ± 0.00	0.21 ± 0.13	0.19 ± 0.12	0.01 ± 0.00	0.01 ± 0.00	**0.46 ± 0.09**	**0.45 ± 0.08**
Rcv1sub3-top10	0.01 ± 0.00	**0.45 ± 0.10**	**0.43 ± 0.09**	0.01 ± 0.00	0.01 ± 0.01	**0.50 ± 0.09**	**0.49 ± 0.07**

AUPRC	Outlier dimensionality = 50 %						
	LOF	PROB_{MAX}	PROB_{SUM}	ROS-M_{MAX}	ROS-M_{SUM}	ROS-MDP_{MAX}	ROS-MDP_{SUM}
Mediamill	0.12 ± 0.02	0.49 ± 0.12	**0.95 ± 0.00**	0.01 ± 0.00	0.01 ± 0.00	0.90 ± 0.05	**0.94 ± 0.01**
Yahoo-business	0.01 ± 0.00	0.05 ± 0.01	0.11 ± 0.04	0.01 ± 0.00	0.01 ± 0.00	**0.63 ± 0.06**	**0.51 ± 0.08**
Yahoo-arts	0.01 ± 0.00	**0.22 ± 0.10**	**0.26 ± 0.09**	0.01 ± 0.00	0.01 ± 0.00	**0.22 ± 0.08**	**0.19 ± 0.06**
Bibtex	0.02 ± 0.01	0.11 ± 0.06	**0.64 ± 0.20**	0.01 ± 0.01	0.01 ± 0.00	**0.61 ± 0.07**	**0.75 ± 0.10**
Enron	0.01 ± 0.00	**0.09 ± 0.05**	**0.17 ± 0.10**	0.01 ± 0.00	0.01 ± 0.00	**0.17 ± 0.10**	0.03 ± 0.01
Birds	0.06 ± 0.05	0.24 ± 0.11	**0.90 ± 0.01**	0.02 ± 0.01	0.02 ± 0.00	0.21 ± 0.09	0.20 ± 0.04
Cal500	0.13 ± 0.02	0.11 ± 0.08	**0.82 ± 0.04**	0.01 ± 0.00	0.01 ± 0.00	0.73 ± 0.06	0.66 ± 0.10
Yeast	0.09 ± 0.05	**0.35 ± 0.03**	**0.39 ± 0.04**	0.01 ± 0.01	0.01 ± 0.00	0.14 ± 0.02	0.15 ± 0.01
Rcv1sub1-top10	0.01 ± 0.00	0.34 ± 0.18	0.53 ± 0.13	0.01 ± 0.00	0.01 ± 0.00	**0.72 ± 0.09**	**0.75 ± 0.09**
Rcv1sub3-top10	0.01 ± 0.00	**0.56 ± 0.15**	**0.70 ± 0.14**	0.01 ± 0.00	0.01 ± 0.00	0.54 ± 0.09	**0.61 ± 0.11**

Table 4.13: Area under the precision-recall curve. Numbers shown in bold indicate the best results on each experiment set (by paired t-test at $\alpha = 0.05$). Dashes (-) indicate the sets that we cannot create due to low output dimensionality.

Results Figures 4.11-4.20 and Tables 4.12 and 4.13 report the experiment results on the public datasets. The results are averages over *five* repetitions of outlier simulations.

(1) *Precision-Alert Rate* Figures 4.11-4.20 show the precision of the tested methods at different alert rates (precision-alert rate (PAR) curves) ranging between 0.00 and 0.04. The vertical dashed lines at alert rate = 0.01 indicate where the alert rate coincides with the ratio of simulated outliers. Notice that, for each dataset, the PAR curves are shown in two groups according to the type of the multivariate combine function (*i.e.*, MAX or SUM) used by the MCOD methods. Within each group, we list the results for different *outlier dimensionality* from 5.0 to 50.0% (for *Yeast*, *Rcv1sub1-top10*, and *Rcv1sub3-top10*, *outlier dimensionality* is given from 5.0 to 50.0%). Table 4.12 presents the *average PAR* (APAR) in $[0.00, 0.01]$ range. The numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set.

Overall, ROS-MDP exhibits the best performance. In terms of APAR over alert rate $[0.00, 0.01]$, ROS-MDP$_{\text{MAX}}$ achieves statistically superior performance on 28 experiments (numbers shown in boldface in Table 4.12); ROS-MDP$_{\text{SUM}}$ does so on 25 experiments. The PAR curves also suggest that ROS-MDP$_{\text{MAX}}$ and ROS-MDP$_{\text{SUM}}$ can produce precisely controllable outlier scores. For example, on *Mediamill*, *Yahoo-business*, *Bibtex*, *Cal500*, *Rcv1sub1-top10*, and *Rcv1sub3-top10*, ROS-MDP$_{\text{MAX}}$ and ROS-MDP$_{\text{SUM}}$ result in excellent PAR curves (Figures 4.11-4.20).

PROB also performs very competently. PROB$_{\text{SUM}}$ reports statistically superior performance on 26 experiments; and PROB$_{\text{MAX}}$ does so on 18 experiments. The PAR curves of the methods show a desirable pattern as well. For example, on *Yahoo-arts* (when *Outlier dimensionality* ≥ 20), *Enron*, *Birds*, *Cal500* (PROB$_{\text{SUM}}$), *Yeast*, and *Rcv1sub3-top10*, PROB$_{\text{MAX}}$ and PROB$_{\text{SUM}}$ show preferable results (Figures 4.11-4.20).

On the other hand, the results of LOF, ROS-M$_{\text{MAX}}$, and ROS-M$_{\text{SUM}}$ apparently do not produce competitive results. Our conjecture on why LOF do not perform well is accounted in the previous discussion on experiments with synthetic data. As expected (Section 4.3.2), ROS-M$_{\text{MAX}}$, and ROS-M$_{\text{SUM}}$ do not produce any meaningful results on the real-world datasets.

The results with respect to the choice of combine function and varying *outlier dimensionality* also reveal an interesting performance pattern. The performance of ROS-MDP$_{\text{MAX}}$ and ROS-MDP$_{\text{SUM}}$ appear to be correlated; when ROS-MDP$_{\text{MAX}}$ performs well, ROS-MDP$_{\text{SUM}}$ also performs well, or *vice versa*. Their APAR are often within statistically equivalent. On the other hand, the performance of PROB$_{\text{MAX}}$ and PROB$_{\text{SUM}}$ show contrasts. When the *outlier dimensionality* is low (5%), PROB$_{\text{MAX}}$ often outperforms PROB$_{\text{SUM}}$. As the *outlier dimensionality* increases, however, PROB$_{\text{SUM}}$ more often outperforms PROB$_{\text{MAX}}$. Note that this agrees with our conjecture that, when the outlier dimensionality is low, the max function should be preferred; when the outlier dimensionality is high, otherwise.

The results also suggests that certain datasets prefer certain methodology. For example, on *Birds*, the PROB methods perform extremely well compared to ROS-MDP. On *Yahoo-business*, the same is observed for ROS-MDP.

(2) *Area Under the Precision-Recall Curve*

Table 4.13 presents the results in terms of AUPRC. Again, the numbers shown in boldface indicate the best results (by paired t-test at $\alpha = 0.05$) on each experiment set. Overall, ROS-MDP and PROB exhibits the best performance. ROS-MDP$_{\text{MAX}}$ achieves statistically superior performance on 26 experiments (numbers shown in boldface in Table 4.13); ROS-MDP$_{\text{SUM}}$ does so on 26 experiments. Similarly, PROB$_{\text{SUM}}$ reports statistically superior performance on 23 experiments; PROB$_{\text{MAX}}$ does so on 16 experiments. On the other hand, the results of LOF, ROS-M$_{\text{MAX}}$, and ROS-M$_{\text{SUM}}$ apparently do not produce competitive results.

4.3.3 Discussion

In this section, we developed and explored solutions for multivariate conditional outlier detection (MCOD).

We started our investigation by proposing the probabilistic approach to multivariate conditional outlier detection that relies on a model of the conditional joint probability $P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$. In this approach, data instances that correspond to low probabilities for this model are considered to be outliers. To build the model of the conditional joint

probability, we used the chain decomposition idea from multi-label classification (MLC) to decompose the model to multiple components, one for each dimension of the output space. The chain model decomposition comes with one important property from the viewpoint of outlier detection and outlier scoring: the multivariate conditional outlier score can be computed using a collection of univariate conditional outlier scores defined upon the decomposed components of the chain model. This property can be generalized and extended to a wide range of new multivariate conditional outlier scores that are defined in terms of the univariate conditional outlier scores.

Following the new multivariate conditional outlier score decomposition schema, we proposed and studied two new modifications of the probabilistic scores. First, we replaced the model components in the chain decomposition of $P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$ with univariate conditional components based on circular output dependences. Second, we proposed the idea of modifying the probabilistic score with reliability weights where less reliable model components contribute less to the overall outlier score.

Next, we explored the application of the new multivariate conditional outlier score decomposition schema to a more general class of conditional outlier scoring approaches that are not necessarily probabilistic. We revisited the ratio-based outlier scoring methods, developed for univariate conditional outlier detection (Section 4.2.2), and showed how to incorporate the ratio-based outlier scores to support multivariate conditional outlier detection.

Through the experiments on the synthetic and public datasets with simulated outliers, we provided empirical evidences that support our proposed MCOD methods. More specifically, the experiments with the probabilistic model-based approach (Section 4.3.1.5) showed that our methods can effectively identify multivariate conditional outliers, which could not be found by analysis in the joint space. The results also indicated that our reliability weights can further improve the MCOD performance and produce more consistent outlier scores. To summarize, by exploiting the probabilistic chain decomposition and modifying the framework using individual model reliabilities, we successfully extended the model-based COD approach to properly handle the MCOD problem, which has not previously been addressed.

From the experiments with the ratio-based approach (Section 4.3.2.3), we confirmed the validity of the ROS framework (Equation (4.10)). The results of our framework tested in

combination with the LOF score (our choice out of many unconditional scores) demonstrated its superior MCOD performance, especially when data are high-dimensional and have a non-linear discriminative boundary. To conclude, the application of the decomposition schema, inspired by the probabilistic MCOD approach, let us develop a new ratio-based MCOD framework that can potentially work with any unconditional outlier score. Our solutions may be particularly useful for the COD problems where we cannot easily obtain reliable probabilistic data models. One important recapitulation of the approach is that it connects the unconditional and conditional outlier detection methodologies that previously have not much benefited from each other. We expect that our approach opens new opportunities in the advancement of conditional outlier detection by adopting and testing with different types of unconditional outlier scores in the framework.

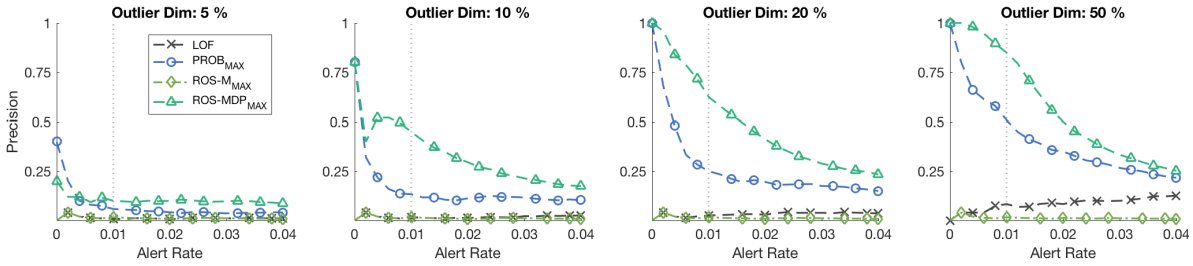
4.4 SUMMARY

We studied the conditional outlier detection (COD) problem, a special type of the outlier detection problem where data instances are associated with a set of binary responses. We introduced two approaches, the probabilistic and ratio-based outlier scoring approaches, by focusing on two types of the COD problem – univariate and multivariate conditional outlier detection (UCOD and MCOD).

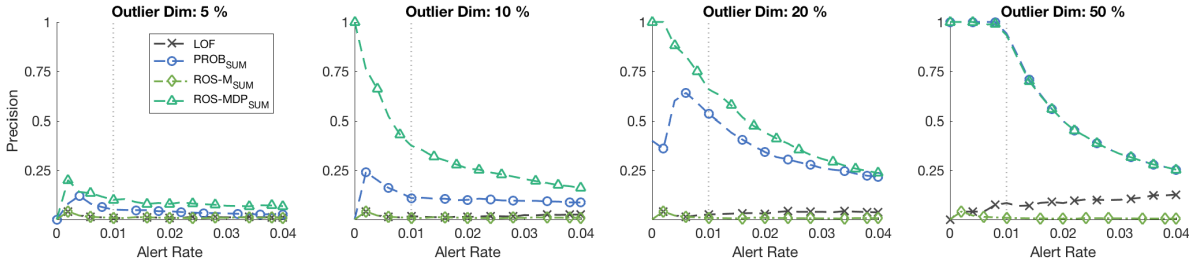
First, we presented the probabilistic COD approach relying on a model of the conditional joint probability $P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$. We reviewed existing solutions to the UCOD problem and set the basic framework that learns a conditional model from data and examines instances by probability estimation, such that data instances corresponding to low probabilities for this model are considered to be outliers. We further developed this framework to tackle the MCOD problem using the chain decomposition idea. That is, to build the model of the conditional joint probability, we decompose the model into multiple components, one for each dimension of the output space. This, in turn, let us compute a multivariate conditional outlier score using a collection of univariate conditional outlier scores defined upon the decomposed components of the chain model.

Second, we proposed and studied the ratio-based outlier scoring (ROS) approach that uses unconditional outlier detection methods and their scores to calculate the conditional score. We defined the ROS score for the UCOD problem by comparing (via ratio) two unconditional outlier scores: one score calculated against data instances with the same observed output value; and another calculated against instances with the opposite output value. We then showed how to incorporate the ROS methods to support MCOD. This new COD approach offers a couple of important advantages. First, it allows us to utilize a wide variety of unconditional outlier scores. Also, it lets us effectively avoid cases where instances with rare $\mathbf{x}$ (but properly associated with $\mathbf{y}$) undesirably receive a high conditional outlier score.

We provided experimental results on synthetic and public datasets with simulated outliers that support the effectiveness of our methods in addressing UCOD and MCOD problems.

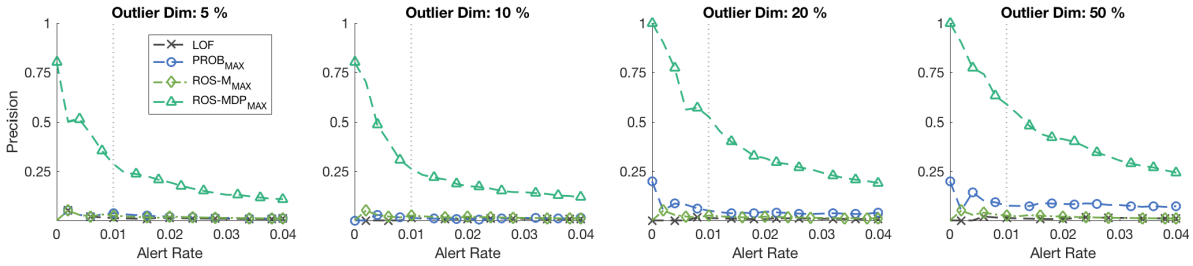


(a) Combine function: *MAX*

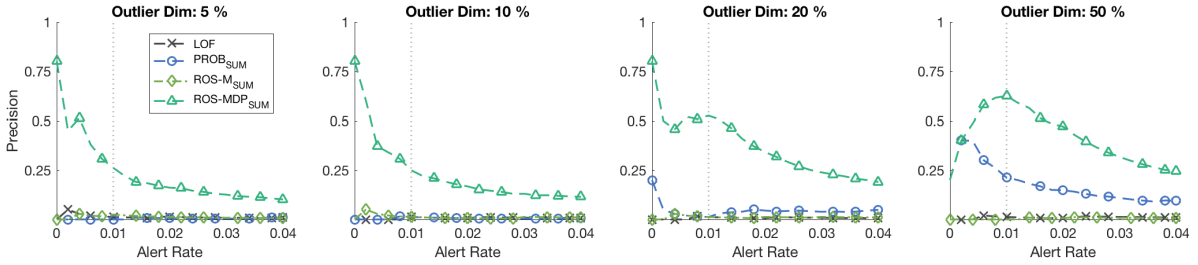


(b) Combine function: *SUM*

Figure 4.11: Precision-alert rate (PAR) on Mediamill (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

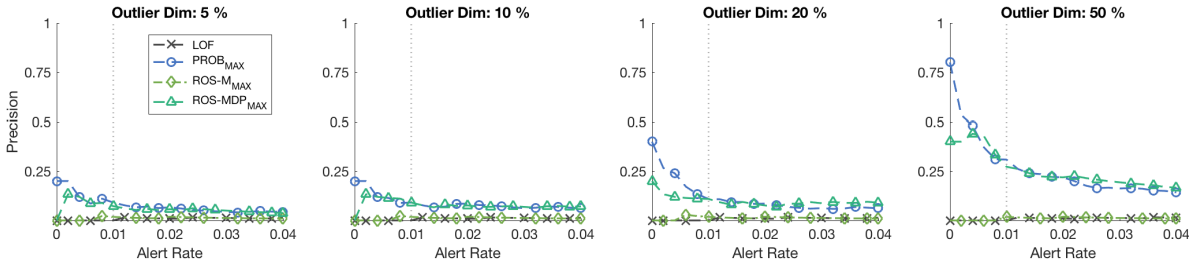


(a) Combine function: *MAX*

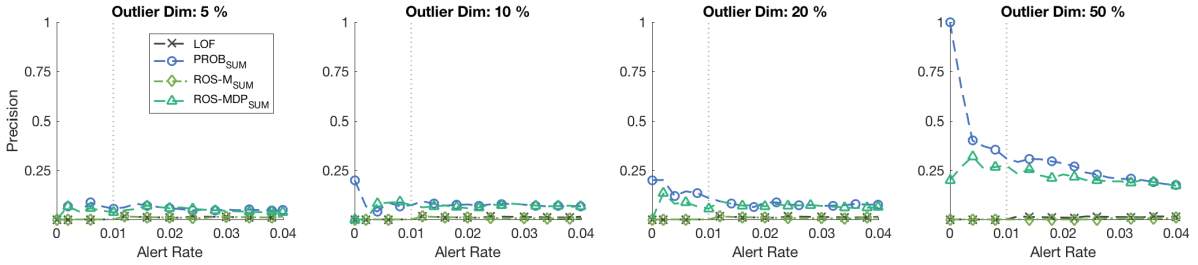


(b) Combine function: *SUM*

Figure 4.12: Precision-alert rate (PAR) on Yahoo-business (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

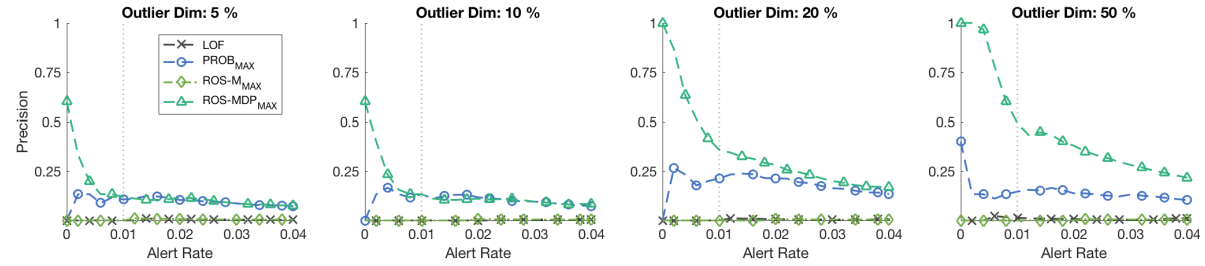


(a) Combine function: *MAX*

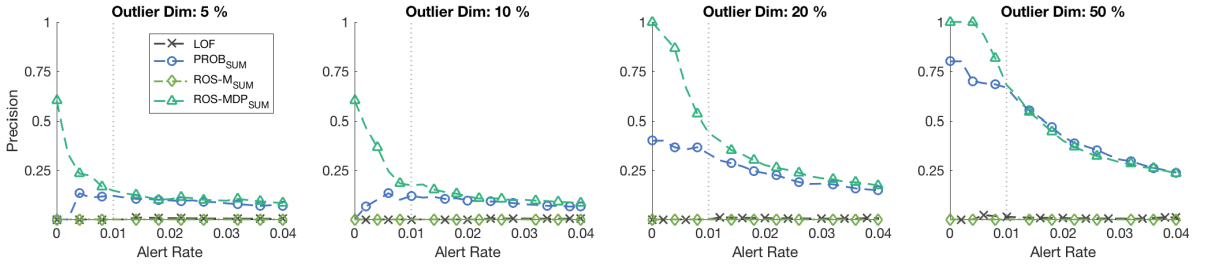


(b) Combine function: *SUM*

Figure 4.13: Precision-alert rate (PAR) on Yahoo-arts (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

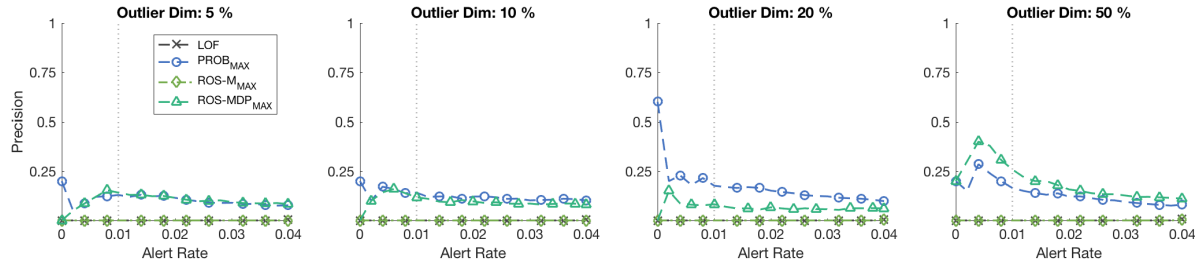


(a) Combine function: *MAX*

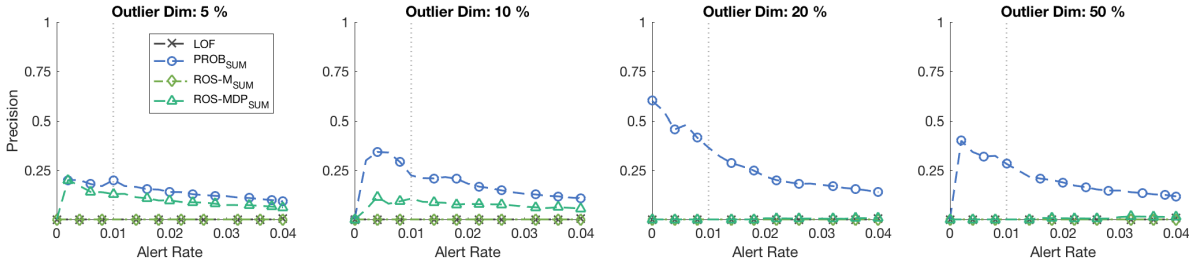


(b) Combine function: *SUM*

Figure 4.14: Precision-alert rate (PAR) on Bibtex (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

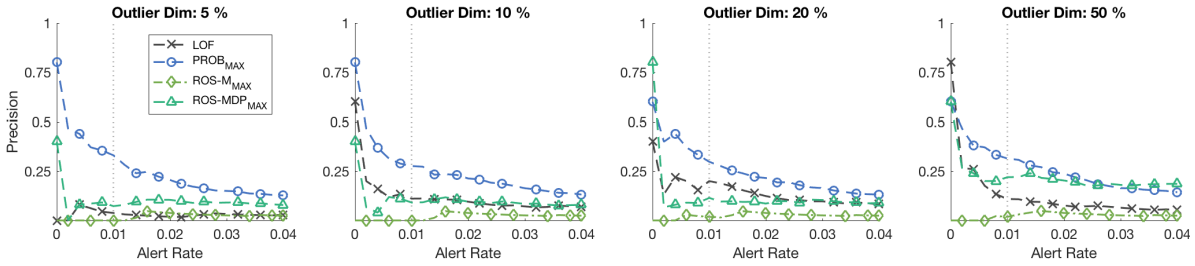


(a) Combine function: *MAX*

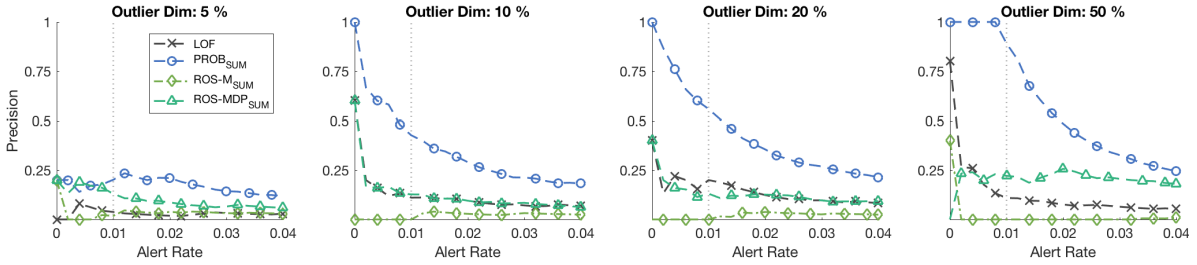


(b) Combine function: *SUM*

Figure 4.15: Precision-alert rate (PAR) on Enron (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

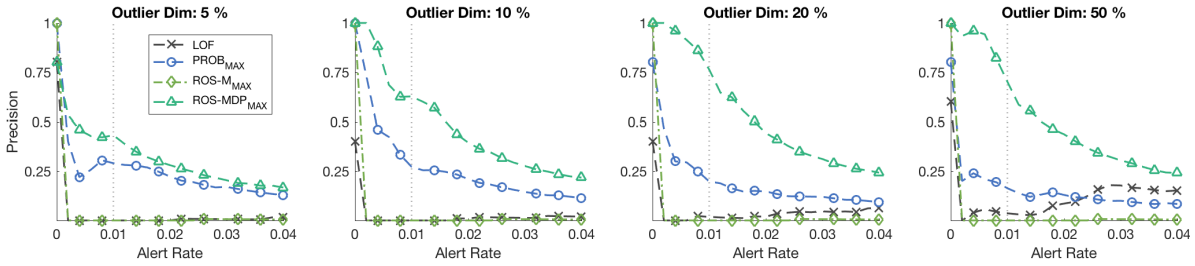


(a) Combine function: *MAX*

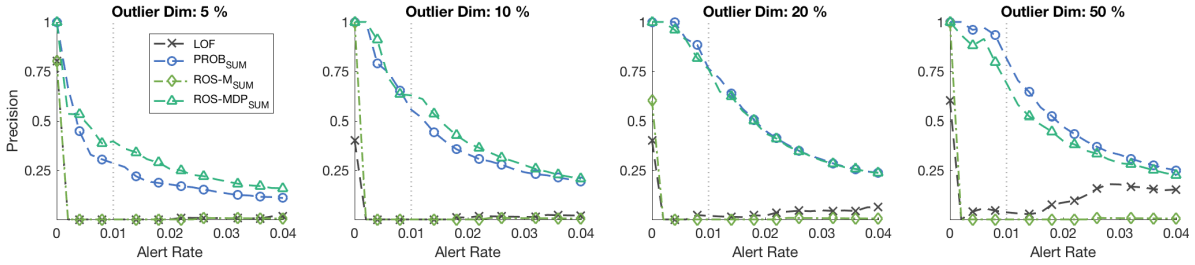


(b) Combine function: *SUM*

Figure 4.16: Precision-alert rate (PAR) on Birds (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

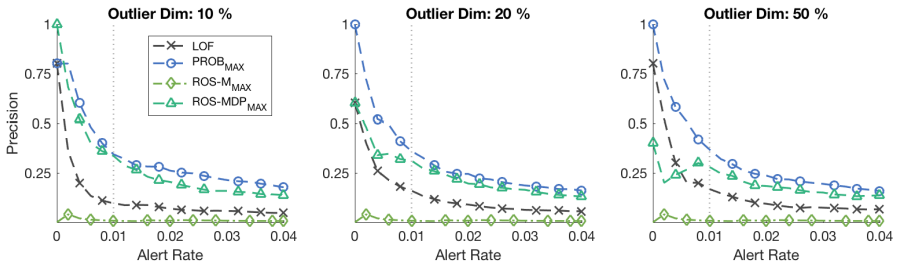


(a) Combine function: *MAX*

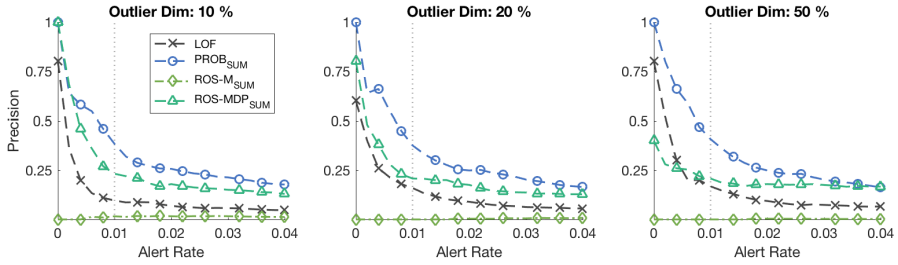


(b) Combine function: *SUM*

Figure 4.17: Precision-alert rate (PAR) on Cal500 (outlier dimensionality = {5.0, 10.0, 20.0, 50.0}%).

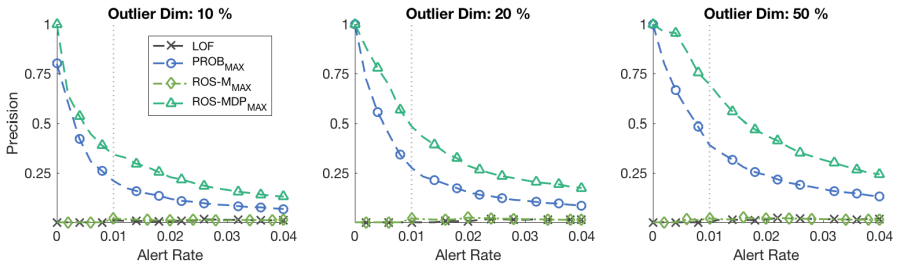


(a) Combine function: *MAX*

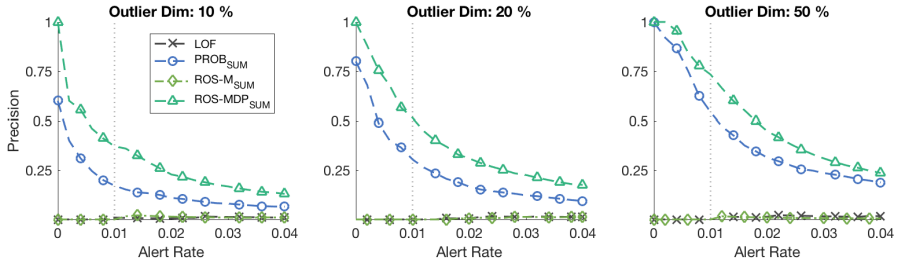


(b) Combine function: *SUM*

Figure 4.18: Precision-alert rate (PAR) on Yeast (outlier dimensionality = {10.0, 20.0, 50.0}%).

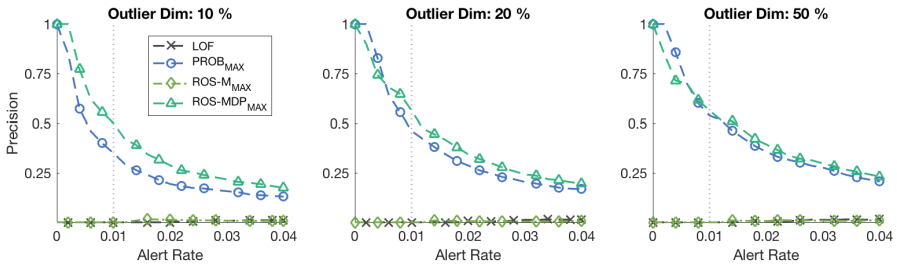


(a) Combine function: *MAX*

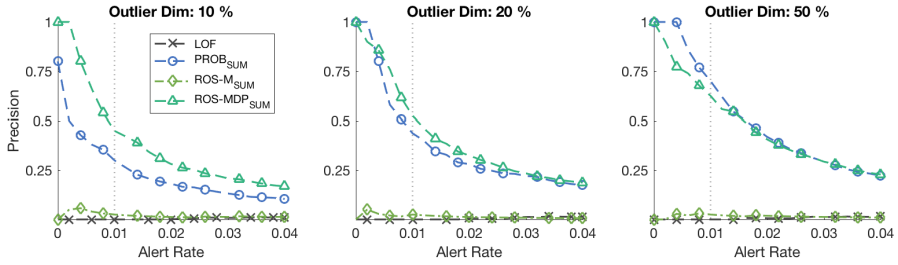


(b) Combine function: *SUM*

Figure 4.19: Precision-alert rate (PAR) on Rcv1sub1-top10 (outlier dimensionality = {10.0, 20.0, 50.0}%).



(a) Combine function: *MAX*



(b) Combine function: *SUM*

Figure 4.20: Precision-alert rate (PAR) on Rcv1sub3-top10 (outlier dimensionality = {10.0, 20.0, 50.0}%).

5.0 CONCLUSIONS

In this thesis, we focused on data objects with multivariate binary output and two problems related to them:

1. **Multi-Label Classification (MLC)** that studies modeling and prediction of multivariate output from complex input-output data.
2. **Conditional Outlier Detection (COD)** that is concerned with how to effectively identify contextually unusual output patterns in multivariate conditional data.

5.1 MODELING AND PREDICTION OF MULTIVARIATE RESPONSES

In Chapter 3, we have considered the multi-label classification (MLC) problem.

5.1.1 Contributions

- We presented a tree-structured probabilistic model that represents the posterior distribution of multivariate output. We also developed supporting algorithms for structure and parameter learning, and a MAP (maximum a posteriori) prediction.
- We studied a mixture model of multiple tree-structured Bayesian networks. We developed algorithms to learn multiple tree structures and their parameters from data and to make a MAP prediction using the trained mixture model.
- We presented a generalized representation of the multivariate posterior distribution that includes a number of previous relevant data models as special cases, such as binary

relevance [Boutell et al., 2004, Clare and King, 2001], classifier chains [Read et al., 2009], and conditional tree-structured Bayesian networks [Batal et al., 2013].

- We extended and applied the Mixtures-of-Experts [Jacobs et al., 1991] framework to represent the conditional joint distribution of multi-dimensional output using our generalized multivariate posterior representation as base models.

5.1.2 Open Questions

- While we have successfully addressed the MLC problem using the structured learning and prediction approach, we tested our models and methods only with a linear probabilistic base model (*i.e.*, logistic regression). However, in many practical problems, the underlying decision boundary is nonlinear or discontinuous. In such cases, our choice of the base model could fail or limit the model capacity. Considering nonlinear probabilistic base models (*e.g.*, kernel SVMs [Shawe-Taylor and Cristianini, 2004] with a post-hoc calibration [Platt, 1999, DeGroot and Fienberg, 1983, Pakdaman, 2017]) would define an interesting extension from our solutions.
- Class imbalance [He and Garcia, 2009] is one of the commonly encountered issues when dealing with classification problems. When class imbalance is present, the obtained (learned from data) classification model could deteriorate both in terms of predictive accuracy and the quality of probability estimates. In this regard, class imbalance could be a critical issue in addressing the MLC problem, especially when a structured learning and prediction approach is used as in our work. Conducting a comprehensive study on the effect of class imbalance to the quality of the MLC models and performance would induce important findings.
- Our prediction algorithm for the mixture models is based on the simulated annealing approach, which is a stochastic approximation to find the global optimum (the maximum a posteriori (MAP) prediction). Although the approach let us achieve outstanding MAP prediction results, a proper non-approximating prediction algorithm would be much preferred. Methods such as dual decomposition [Sontag, 2010] would be a promising candidate.

- An investigation of a large-scale MLC problem that has high N and d (*i.e.*, a large number of instances with a high-dimensional output space) is another intriguing research direction. In such an extreme case, our structure learning algorithms may suffer, since their time complexity is bounded by both N and d^2 (see Sections 3.2.2.1, 3.3.4.1, and 3.4.4.1). What kind of approximations could address such a massive MLC problem within a reasonable amount of time and space? How can we discover and utilize the conditional relations among the output variables in such solutions? To this end, the solutions would suggest more effective approaches to practical problems that require scalable data models and prediction, such as web-scale data analyses [Agrawal et al., 2013, Yu et al., 2014, Bhatia et al., 2015].

- Considering a similar predictive modeling problem with multi-dimensional *continuous* output variables – which is *multi-target regression* (MTR) [Borchani et al., 2015] – would be a very interesting and practical investigation. Could our structured prediction and ensemble approaches apply to the problem with the different type of output? What kind of modification should be taken to resolve the regression counterpart? Furthermore, we may consider an even more complicated problem where the output space is defined by a mixture of discrete and continuous attributes (*i.e.*, multivariate conditional data modeling with mixed types of output variables) [De Leon and Chough, 2013, Dine et al., 2009, Choi, 2012]. Would our proposed structured modeling and prediction algorithms still work with different types of output variables and produce acceptable performance? Could our ensemble approaches still improve the predictive accuracy of the base multi-dimensional methods? Success in this regard would deliver a useful set of tools and theories for complex data analysis, such as medical/clinical data analysis [Miglioretti, 2003, Dine et al., 2009, Saha et al., 2017].

5.2 CONDITIONAL OUTLIER DETECTION

In Chapter 4, we have explored the conditional outlier detection (COD) problem.

5.2.1 Contributions

- We started our investigation by considering conditional outlier detection with one dimensional output space [Hauskrecht et al., 2007]. We have identified a large gap and disconnect in the development of conditional and unconditional outlier methods. Motivated by this fact, we proposed a new ratio-based conditional outlier (ROS) score that can be derived from any unconditional outlier score. To cope with a high-dimensional input data, we proposed a variant of the ROS method that applies discriminative dimensionality reduction techniques prior to calculating the ratio-based score.
- We introduced and defined multivariate conditional outlier detection (MCOD) problem in which outliers are assumed to occur in multi-dimensional binary output (response) space, conditioned on their input (context).
- We presented a probabilistic framework for the multivariate conditional outlier detection (MCOD) problem that finds data instances that fall in the regions of low conditional joint probability. Inspired by the multi-label classification (MLC) models, our framework works by decomposing the model using the chain rule and by using a collection of discriminative probabilistic models to represent each output dimension. We showed that under this model the probabilistic multivariate conditional outlier score decomposes to the sum of probabilistic univariate condition outlier scores, one univariate score per one ouptut dimension.
- We use the result on the decomposition of the probabilistic outlier scores to define a more general family of decomposable multivariate conditional scores and extended it to handle:
  - models with circular instead of chain output dependences;
  - models permitting outlier score weighting; and
  - models with univariate Ratio of Outlier Scores (ROS) and its variants.

5.2.2 Open Issues

- To build a probabilistic data model for MCOD, we relaxed the chain rule and applied the circular-chain heuristic to represent the underlying output dependency structure. At this

time we do not have any empirical evidence that this heuristic either improves or hurts the outlier detection performance. However, a theoretical justification of the circular relaxation or its performance guarantees remain an interesting open question.

- Similarly, we have developed heuristically-motivated weighting schemas for the probabilistic model-based MCOD approach. Although our proposed approaches have shown a favorable performance in our empirical analysis, a theoretical study on the bounds that the weights guarantee could be followed. The new findings will let us assure the optimality of the proposed schemas or come up with the optimal set of weights for the approach.
- We have developed the ROS approach to bridge the gap in between unconditional and conditional outlier methods. However, in the experimental phase we tested ROS approach by considering only one possible unconditional outlier score, namely Local Outlier Factors (LOF) [Breunig et al., 2000]. Plugging in other types of unconditional outlier scores to the ROS framework and testing their performance would lead to better and more refined insights on the advantages and potential shortcomings of ROS.
- Throughout our evaluation studies, we only considered and used a simple discriminative function based on the logistic regression model to represent probabilistic dependences for calculating the probabilistic outlier score. It would be interesting to test and explore alternative nonlinear models and their benefits for the problem. Similarly, simple logistic regression models were used to implement ROS projection methods. Applying and testing different dimensionality reduction methods in the ROS framework would let us further improve the method.
- In our experiments, we built our data models from all data, without excluding any of the instances. However, we might obtain more precise data models and, hence, further improve outlier detection performance, if we trained the models on data without suspected outliers [Aggarwal, 2017]. Interesting questions would be: what if we train the data models from an "outlier-free" dataset? What if we instead applied a robust model building approach to minimize the effect of outliers on the data model? Although the questions may raise an unrealistic scenario, answering them would push us to contemplate the correctness and capability of the method on a fundamental level.

- Similar to one of the open issues of MLC, considering MCOD on data with an output space that is defined by continuous attributes or, most generally, a mixture of discrete and continuous attributes would be an interesting and useful research direction.
- Another open question is related to missing output values: what is the best decision (outlier calls) when only a subset of output values are observed? How do we compute the outlier scores based on such limited observations?
- In the thesis, we have been concerned only with data in which the division of attributes between inputs and outputs is pre-specified. However, if no such information is available, how do we apply our proposed approach to perform MCOD? More generally, how can we determine whether to apply conditional or unconditional outlier detection approach when the input-output division is not a priori known? The results would suggest a generalized data analytic procedure for (both conditional and unconditional) outlier detection.

BIBLIOGRAPHY

[Aggarwal, 2017] Aggarwal, C. C. (2017). *Outlier Analysis*. Springer, second edition.

[Aggarwal et al., 2001] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, pages 420–434, London, UK, UK. Springer-Verlag.

[Aggarwal and Yu, 2001] Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data. *SIGMOD Rec.*, 30:37–46.

[Agrawal et al., 2013] Agrawal, R., Gupta, A., Prabhu, Y., and Varma, M. (2013). Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, pages 13–24. ACM.

[Amer et al., 2013] Amer, M., Goldstein, M., and Abdennadher, S. (2013). Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*.

[Angiulli and Pizzuti, 2002] Angiulli, F. and Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, pages 15–26, London, UK, UK. Springer-Verlag.

[Antonucci et al., 2013] Antonucci, A., Corani, G., Mauá, D. D., and Gabaglio, S. (2013). An ensemble of bayesian networks for multilabel classification. In *IJCAI*, pages 1220–1225.

[Arning et al., 1996] Arning, A., Agrawal, R., and Raghavan, P. (1996). A linear method for deviation detection in large databases. In *KDD*, pages 164–169.

[Bakir et al., 2007] Bakir, G. H., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., and Vishwanathan, S. V. N. (2007). *Predicting Structured Data (Neural Information Processing)*. The MIT Press.

[Barbara et al., 2001] Barbara, D., Wu, N., and Jajodia, S. (2001). Detecting novel network intrusions using bayes estimators. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–17. SIAM.

[Batal et al., 2013] Batal, I., Hong, C., and Hauskrecht, M. (2013). An efficient probabilistic framework for multi-dimensional classification. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, pages 2417–2422. ACM.

[Bella et al., 2009] Bella, A., Ferri, C., Hernández-Orallo, J., and Ramírez-Quintana, M. J. (2009). Similarity-binning averaging: a generalisation of binning calibration. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 341–349. Springer.

[Berger, 1985] Berger, J. (1985). *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer, New York, NY, second edition.

[Bhatia et al., 2015] Bhatia, K., Jain, H., Kar, P., Varma, M., and Jain, P. (2015). Sparse local embeddings for extreme multi-label classification. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 730–738. Curran Associates, Inc.

[Bielza et al., 2011] Bielza, C., Li, G., and Larrañaga, P. (2011). Multi-dimensional classification with bayesian networks. *Int'l Journal of Approximate Reasoning*, 52(6):705–727.

[Bishop and Lasserre, 2007] Bishop, C. and Lasserre, J. (2007). Generative or discriminative? getting the best of both worlds. *BAYESIAN STATISTICS*, 8:3–24.

[Borchani et al., 2015] Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233.

[Boutell et al., 2004] Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757 – 1771.

[Bradley and Guestrin, 2010] Bradley, J. K. and Guestrin, C. (2010). Learning tree conditional random fields. In *International Conference on Machine Learning (ICML 2010)*, Haifa, Israel.

[Breunig et al., 2000] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM.

[Brier, 1950] Brier, G. W. (1950). Verification of Forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

[Byers and Raftery, 1998] Byers, S. and Raftery, A. E. (1998). Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584.

[Cao et al., 2010] Cao, K.-A. L., Meugnier, E., and McLachlan, G. J. (2010). Integrative mixture of experts to combine clinical factors and gene markers. *Bioinformatics*, 26(9):1192–1198.

[Cetin and Karl, 2001] Cetin, M. and Karl, W. C. (2001). Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization. *Image Processing, IEEE Transactions on*, 10(4):623–631.

[Cheng and Hüllermeier, 2009] Cheng, W. and Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225.

[Choi, 2012] Choi, J. (2012). *Prediction in the joint modeling of mixed types of multivariate longitudinal outcomes and a time-to-event outcome*. PhD thesis, University of Pittsburgh.

[Chung, 1997] Chung, F. R. (1997). *Spectral graph theory*. Number 92. American Mathematical Soc.

[Clare and King, 2001] Clare, A. and King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *In: Lecture Notes in Computer Science*, pages 42–53. Springer.

[Cohen, 1995] Cohen, W. W. (1995). Fast effective rule induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann.

[Cooper and Herskovits, 1992] Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347.

[Das and Schneider, 2007] Das, K. and Schneider, J. (2007). Detecting anomalous records in categorical datasets. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 220–229, New York, NY, USA. ACM.

[Davis and Goadrich, 2006] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

[De Leon and Chough, 2013] De Leon, A. R. and Chough, K. C. (2013). *Analysis of mixed data: methods & applications*. CRC Press.

[De Stefano et al., 2000] De Stefano, C., Sansone, C., and Vento, M. (2000). To reject or not to reject: That is the question-an answer in case of neural classifiers. *Trans. Sys. Man Cyber Part C*, 30(1):84–94.

[DeGroot and Fienberg, 1983] DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *The Statistician: Journal of the Institute of Statisticians*, 32:12–22.

[Dembczynski et al., 2010] Dembczynski, K., Cheng, W., and Hüllermeier, E. (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 279–286. Omnipress.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38.

[Dietterich and Bakiri, 1995] Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.*, 2(1):263–286.

[Dine et al., 2009] Dine, A., Larocque, D., and Bellavance, F. (2009). Multivariate trees for mixed outcomes. *Computational Statistics & Data Analysis*, 53(11):3795–3804.

[Dundar et al., 2007] Dundar, M., Krishnapuram, B., Bi, J., and Rao, R. B. (2007). Learning classifiers when the training data is not iid. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 756–761, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Ebrahimpour et al., 2009] Ebrahimpour, R., Moradian, M. R., Esmkhani, A., and Jafarlou, F. M. (2009). Recognition of persian handwritten digits using characterization loci and mixture of experts. *JDCTA*, 3(3):42–46.

[Edmonds, 1967] Edmonds, J. (1967). Optimum branchings. *Research of the National Bureau of Standards*, 71B:233–240.

[Elisseeff and Weston, 2001] Elisseeff, A. and Weston, J. (2001). A kernel method for multilabelled classification. In *NIPS*, pages 681–687.

[Eskin et al., 2002] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer.

[Estabrooks and Japkowicz, 2001] Estabrooks, A. and Japkowicz, N. (2001). A mixture-of-experts framework for text classification. In *Proceedings of the 2001 Workshop on Computational Natural Language Learning - Volume 7*, ConLL '01, pages 9:1–9:8, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Fan et al., 2001] Fan, W., Miller, M., Stolfo, S. J., Lee, W., and Chan, P. K. (2001). Using artificial anomalies to detect unknown and known network intrusions. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 123–130, Washington, DC, USA. IEEE Computer Society.

[Fawcett and Provost, 1997] Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316.

[Garcia-Teodoro et al., 2009] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28.

[Ghamrawi and McCallum, 2005] Ghamrawi, N. and McCallum, A. (2005). Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 195–200. ACM.

[Godbole and Sarawagi, 2004] Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *PAKDD'04*, pages 22–30.

[Gormley and Murphy, 2011] Gormley, I. C. and Murphy, T. B. (2011). *Mixture of Experts Modelling with Social Science Applications*, pages 101–121. John Wiley & Sons, Ltd.

[Guttormsson et al., 1999] Guttormsson, S. E., Marks, R., El-Sharkawi, M., and Kerszenbaum, I. (1999). Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion*, 14(1):16–22.

[Hauskrecht et al., 2016] Hauskrecht, M., Batal, I., Hong, C., Nguyen, Q., Cooper, G. F., Visweswaran, S., and Clermont, G. (2016). Outlier-based detection of unusual patient-management actions: An {ICU} study. *Journal of Biomedical Informatics*, 64:211 – 221.

[Hauskrecht et al., 2013] Hauskrecht, M., Batal, I., Valko, M., Visweswaran, S., Cooper, G. F., and Clermont, G. (2013). Outlier detection for patient monitoring and alerting. *Journal of Biomedical Informatics*, 46(1):47–55.

[Hauskrecht et al., 2010] Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaram, S., and Cooper, G. (2010). Conditional outlier detection for clinical alerting. *Annual American Medical Informatics Association Symposium*.

[Hauskrecht et al., 2007] Hauskrecht, M., Valko, M., Kveton, B., Visweswaram, S., and Cooper, G. (2007). Evidence-based anomaly detection. In *Annual American Medical Informatics Association Symposium*, pages 319–324.

[Hawkins, 1980] Hawkins, D. (1980). *Identification of Outliers*. Monographs on applied probability and statistics. Chapman and Hall.

[Hawkins et al., 2002] Hawkins, S., He, H., Williams, G. J., and Baxter, R. A. (2002). Outlier detection using replicator neural networks. In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, DaWaK 2000, pages 170–180, London, UK, UK. Springer-Verlag.

[He and Garcia, 2009] He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.

[Hinneburg et al., 2000] Hinneburg, A., Aggarwal, C. C., and Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 506–515, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Hodge and Austin, 2004] Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126.

[Hong et al., 2014] Hong, C., Batal, I., and Hauskrecht, M. (2014). A mixtures-of-trees framework for multi-label classification. In *Proceedings of the 23nd ACM International Conference on Information and Knowledge Management*, CIKM '14. ACM.

[Hong et al., 2015] Hong, C., Batal, I., and Hauskrecht, M. (2015). A generalized mixture framework for multi-label classification. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM.

[Hong and Hauskrecht, 2015] Hong, C. and Hauskrecht, M. (2015). Multivariate conditional anomaly detection and its clinical application. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4239–4240. AAAI Press.

[Hong and Hauskrecht, 2016] Hong, C. and Hauskrecht, M. (2016). Multivariate conditional outlier detection and its clinical application. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 4216–4217.

[Hsu et al., 2009] Hsu, D., Kakade, S., Langford, J., and Zhang, T. (2009). Multi-label prediction via compressed sensing. In *NIPS*, pages 772–780.

[Hubert and Debruyne, 2010] Hubert, M. and Debruyne, M. (2010). Minimum covariance determinant. *Wiley interdisciplinary reviews: Computational statistics*, 2(1):36–43.

[Hyvärinen et al., 2004] Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent component analysis*, volume 46. John Wiley & Sons.

[Ioffe and Forsyth, 2001a] Ioffe, S. and Forsyth, D. (2001a). Human tracking with mixtures of trees. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 690–695 vol.1.

[Ioffe and Forsyth, 2001b] Ioffe, S. and Forsyth, D. A. (2001b). Mixtures of trees for object recognition. In *CVPR (2)*, pages 180–185. IEEE Computer Society.

[Jacobs and Jordan, 1993] Jacobs, R. A. and Jordan, M. I. (1993). Learning piecewise control strategies in a modular neural network architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2):337–345.

[Jacobs et al., 1991] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87.

[Janakiram et al., 2006] Janakiram, D., Reddy, V., and Kumar, A. P. (2006). Outlier detection in wireless sensor networks using bayesian belief networks. In *Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on*, pages 1–6. IEEE.

[Johnson et al., 1998] Johnson, T., Kwok, I., and Ng, R. T. (1998). Fast computation of 2-dimensional depth contours. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31, 1998*, pages 224–228. AAAI Press.

[Jolliffe, 1986] Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer.

[Kazawa et al., 2005] Kazawa, H., Izumitani, T., Taira, H., and Maeda, E. (2005). Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems 17*, pages 649–656. MIT Press.

[Keller et al., 2012] Keller, F., Muller, E., and Bohm, K. (2012). Hics: High contrast subspaces for density-based outlier ranking. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1037–1048. IEEE.

[Kim et al., 2010] Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., Speck, J. A., and Turnbull, D. (2010). State of the art report: Music emotion recognition: A state of the art review. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 255–266, Utrecht, The Netherlands. http://ismir2010.ismir.net/proceedings/ismir2010-45.pdf.

[Kirshner and Smyth, 2007] Kirshner, S. and Smyth, P. (2007). Infinite mixtures of trees. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 417–423, New York, NY, USA. ACM.

[Knorr and Ng, 1997] Knorr, E. M. and Ng, R. T. (1997). A unified approach for mining outliers. In *Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research*, CASCON '97, pages 11–. IBM Press.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

[Kriegel et al., 2010] Kriegel, H.-P., Kröger, P., and Zimek, A. (2010). Outlier detection techniques. In *Tutorial at the 2010 SIAM International Conference on Data Mining*.

[Kriegel et al., 2008] Kriegel, H.-P., S hubert, M., and Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 444–452, New York, NY, USA. ACM.

[Kumar et al., 2012] Kumar, A., Vembu, S., Menon, A. K., and Elkan, C. (2012). Learning and inference in probabilistic classifier chains with beam search. In *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer-Verlag.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01.

[Lauritzen and Spiegelhalter, 1988] Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224.

[Lazarevic and Kumar, 2005] Lazarevic, A. and Kumar, V. (2005). Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 157–166. ACM.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lee et al., 1997] Lee, W., Stolfo, S. J., and Chan, P. K. (1997). Learning patterns from unix process execution traces for intrusion detection. In *Proceedings of the AAAI Workshop on AI Methods in Fraud and Risk Management*.

[Lewis et al., 2004] Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

[Liu and Nocedal, 1989] Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528.

[Liu et al., 2004] Liu, H., Shah, S., and Jiang, W. (2004). On-line outlier detection and data cleaning. *Computers & Chemical Engineering*, 28(9):1635–1647.

[Lu, 2006] Lu, Z. (2006). A regularized minimum cross-entropy algorithm on mixtures of experts for time series prediction and curve detection. *Pattern Recogn. Lett.*, 27(9):947–955.

[MacKay, 2003] MacKay, D. J. C. (2003). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

[Markou and Singh, 2003] Markou, M. and Singh, S. (2003). Novelty detection: A review—part 1: Statistical approaches. *Signal Process.*, 83(12):2481–2497.

[Meilă and Jordan, 2000] Meilă, M. and Jordan, M. I. (2000). Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48.

[Miglioretti, 2003] Miglioretti, D. L. (2003). Latent transition regression for mixed outcomes. *Biometrics*, 59(3):710–720.

[Moon, 1996] Moon, T. (1996). The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60.

[Mossavat et al., 2010] Mossavat, S. I., Amft, O., De Vries, B., Petkov, P., and Kleijn, W. B. (2010). A bayesian hierarchical mixture of experts approach to estimate speech quality. In *2010 2nd International Workshop on Quality of Multimedia Experience*, volume QoMEX 2010 - Proceedings, pages 200–205.

[Ng, 2004] Ng, A. Y. (2004). Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM.

[Ng and Jordan, 2002] Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848.

[Nowlan, 1991] Nowlan, S. J. (1991). *Soft Competitive Adaptation: Neural Network Learning Algorithms Based on Fitting Statistical Mixtures*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA. UMI Order No. GAX91-26958.

[Nowozin et al., 2014] Nowozin, S., Gehler, P. V., Jancsary, J., and Lampert, C. H. (2014). *Advanced Structured Prediction*. The MIT Press.

[Odin and Addison, 2000] Odin, T. and Addison, D. (2000). Novelty detection using neural network technology. In *the COMADEN Conference*, Houston, TX, USA.

[Pakdaman et al., 2014] Pakdaman, M., Batal, I., Liu, Z., Hong, C., and Hauskrecht, M. (2014). An optimization-based framework to learn conditional random fields for multi-label classification. In *SDM*. SIAM.

[Pakdaman, 2017] Pakdaman, M. N. (2017). Obtaining accurate probabilities using classifier calibration.

[Papadimitriou et al., 2003] Papadimitriou, S., Kitagawa, H., Gibbons, P. B., and Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326. IEEE.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Pham and Pagh, 2012] Pham, N. and Pagh, R. (2012). A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 877–885, New York, NY, USA. ACM.

[Pimentel et al., 2014] Pimentel, M., Clifton, D., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99:215 – 249.

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

[Qi et al., 2007a] Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., Mei, T., and Zhang, H.-J. (2007a). Correlative multi-label video annotation. In *Proceedings of the 15th international conference on Multimedia*, pages 17–26. ACM.

[Qi et al., 2007b] Qi, Y., Klein-Seetharaman, J., and Bar-Joseph, Z. (2007b). A mixture of feature experts approach for protein-protein interaction prediction. *BMC bioinformatics*, 8(Suppl 10):S6.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106.

[Raiffa, 1997] Raiffa, H. (1997). *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Mcgraw-Hill.

[Read et al., 2009] Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier chains for multi-label classification. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer-Verlag.

[Roth, 2005] Roth, V. (2005). Outlier detection with one-class kernel fisher discriminants. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1169–1176. MIT Press.

[Roth, 2006] Roth, V. (2006). Kernel fisher discriminants for outlier detection. *Neural computation*, 18(4):942–960.

[Rousseeuw, 1984] Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):pp. 871–880.

[Rousseeuw and Driessen, 1999] Rousseeuw, P. J. and Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

[Rousseeuw and Hubert, 2011] Rousseeuw, P. J. and Hubert, M. (2011). Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):73–79.

[Rousseeuw and Leroy, 1987] Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York, NY, USA.

[Rousseeuw and Zomeren, 1990] Rousseeuw, P. J. and Zomeren, B. C. v. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411):pp. 633–639.

[Ruts and Rousseeuw, 1996] Ruts, I. and Rousseeuw, P. J. (1996). Computing depth contours of bivariate point clouds. *Computational Statistics & Data Analysis*, 23:153–168.

[Saha et al., 2017] Saha, B., Gupta, S., Phung, D., and Venkatesh, S. (2017). A framework for mixed-type multi-outcome prediction with applications in healthcare. *IEEE journal of biomedical and health informatics*.

[Schapire and Singer, 2000] Schapire, R. E. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2):135–168.

[Schölkopf et al., 1999] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (1999). Support vector method for novelty detection. *NIPS*, 12:582–588.

[Shahaf and Guestrin, 2009] Shahaf, D. and Guestrin, C. (2009). Learning thin junction trees via graph cuts. In *AISTATS*, volume 5 of *JMLR Proceedings*, pages 113–120. JMLR.org.

[Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

[Song et al., 2007] Song, X., Wu, M., Jermaine, C., and Ranka, S. (2007). Conditional anomaly detection. *IEEE Trans. on Knowl. and Data Eng.*, 19(5):631–645.

[Sontag, 2010] Sontag, D. (2010). *Approximate Inference in Graphical Models using LP Relaxations*. PhD thesis, Massachusetts Institute of Technology.

[Tai and Lin, 2010] Tai, F. and Lin, H.-T. (2010). Multi-label classification with principle label space transformation. In *the 2nd International Workshop on Multi-Label Learning*.

[Tan et al., 2002] Tan, K., Killourhy, K., and Maxion, R. (2002). Undermining an anomaly-based intrusion detection system using common exploits. In *Recent Advances in Intrusion Detection*, Lecture Notes in Computer Science. Springer Berlin Heidelberg.

[Tang et al., 2002] Tang, J., Chen, Z., Fu, A. W.-C., and Cheung, D. W.-L. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '02, pages 535–548, London, UK, UK. Springer-Verlag.

[Tarjan, 1977] Tarjan, R. E. (1977). Finding Optimum Branchings. *Networks*, 7:22–35.

[Tax and Duin, 2004] Tax, D. M. and Duin, R. P. (2004). Support vector data description. *Machine learning*, 54(1):45–66.

[Tipping, 2001] Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244.

[Titterington et al., 1985] Titterington, D., Smith, A., and Makov, U. (1985). *Statistical analysis of finite mixture distributions*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley.

[Trohidis et al., 2011] Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. (2011). Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011(1):1–9.

[Trohidis et al., 2008] Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. (2008). Multi-label classification of music into emotions. In *ISMIR*, pages 325–330.

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi label classification: An overview. *International Journal of Data Warehouse and Mining*, 3(3):1–13.

[Tsoumakas et al., 2010] Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer US.

[Tukey, 1961] Tukey, J. W. (1961). Curves as parameters, and touch estimation. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 681–694, Berkeley, Calif. University of California Press.

[Ulusoy and Bishop, 2006] Ulusoy, I. and Bishop, C. M. (2006). Comparison of generative and discriminative techniques for object detection and classification. In *Toward Category-Level Object Recognition*, pages 173–195. Springer.

[Valdes and Skinner, 2000] Valdes, A. and Skinner, K. (2000). Adaptive, model-based monitoring for cyber attack detection. In *International Workshop on Recent Advances in Intrusion Detection*, pages 80–93. Springer.

[Valko et al., 2008] Valko, M., Cooper, G. F., Seybert, A., Visweswaran, S., Saul, M., and Hauskrecht, M. (2008). Conditional anomaly detection methods for patient-management alert systems. In *Workshop on Machine Learning in Health Care Applications in The 25th International Conference on Machine Learning*.

[Valko and Hauskrecht, 2008] Valko, M. and Hauskrecht, M. (2008). Distance metric learning for conditional anomaly detection. In *Twenty-First International Florida Artificial Intelligence Research Society Conference*. AAAI Press.

[Valko et al., 2011a] Valko, M., Kveton, B., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2011a). Conditional anomaly detection with soft harmonic functions. In *Proceedings of the 2011 IEEE International Conference on Data Mining*.

[Valko et al., 2011b] Valko, M., Valizadegan, H., Cooper, B. K. G. F., and Hauskrecht, M. (2011b). Conditional anomaly detection using soft harmonic functions: An application to

clinical alerting. In *The 28th International Conference on Machine Learning Workshop on Machine Learning for Global Challenges*.

[van der Gaag and de Waal, 2006] van der Gaag, L. C. and de Waal, P. R. (2006). Multi-dimensional bayesian network classifiers. In *Probabilistic Graphical Models*, pages 107–114.

[Šingliar and Hauskrecht, 2007] Šingliar, T. and Hauskrecht, M. (2007). Modeling highway traffic volumes. In *Proceedings of the 18th European Conference on Machine Learning*, ECML '07, pages 732–739. Springer-Verlag.

[Wang, 2010] Wang, S. (2010). A comprehensive survey of data mining-based accounting-fraud detection research. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 1, pages 50–53.

[Weber et al., 1998] Weber, R., Schek, H.-J., and Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Weigend et al., 1995] Weigend, A. S., Mangeas, M., and Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, 6:373–399.

[Weigend and Shi, 2000] Weigend, A. S. and Shi, S. (2000). Predicting daily probability distributions of S&P500 returns. *Journal of Forecasting*, 19(4).

[Williams et al., 2002] Williams, G., Baxter, R., He, H., Hawkins, S., and Gu, L. (2002). A comparative study of rnn for outlier detection in data mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 709–, Washington, DC, USA. IEEE Computer Society.

[Willmott and Matsuura, 2005] Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82.

[Wong et al., 2003] Wong, W.-K., Moore, A., Cooper, G., and Wagner, M. (2003). Bayesian network anomaly pattern detection for disease outbreaks. In Fawcett, T. and Mishra, N., editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815, Menlo Park, California. AAAI Press.

[Yu et al., 2014] Yu, H.-F., Jain, P., Kar, P., and Dhillon, I. (2014). Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning*, pages 593–601.

[Yuan et al., 2004] Yuan, C., Lu, T.-C., and Druzdzel, M. J. (2004). Annealed map. In *UAI*, pages 628–635. AUAI Press.

[Yuksel et al., 2012] Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Trans. Neural Netw. Learning Syst.*, 23(8):1177–1193.

[Zaragoza et al., 2011] Zaragoza, J. H., Sucar, L. E., Morales, E. F., Bielza, C., and Larrañaga, P. (2011). Bayesian chain classifiers for multidimensional classification. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Three*, IJCAI'11, pages 2192–2197. AAAI Press.

[Zhang and Zhou, 2013] Zhang, M. and Zhou, Z. (2013). A review on multi-label learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, PP(99):1.

[Zhang and Zhang, 2010] Zhang, M.-L. and Zhang, K. (2010). Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 999–1008. ACM.

[Zhang and Zhou, 2006] Zhang, M.-L. and Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

[Zhang and Zhou, 2007] Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recogn.*, 40(7):2038–2048.

[Zhang and Schneider, 2011] Zhang, Y. and Schneider, J. (2011). Multi-label output codes using canonical correlation analysis. In *AISTATS 2011*.

[Zhang and Schneider, 2012] Zhang, Y. and Schneider, J. (2012). Maximum margin output coding. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1575–1582.

[Zhou et al., 2004] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.