A NOVEL ALGORITHM FOR MULTI-LABEL CLASSIFICATION

A THESIS SUBMITTED TO
SAVITRIBAI PHULE PUNE UNIVERSITY

FOR THE AWARD OF DEGREE OF
DOCTOR OF PHILOSOPHY (PH.D.)
(COMPUTER ENGINEERING)
IN THE FACULTY OF
SCIENCE AND TECHNOLOGY

SUBMITTED BY

Mrs. Vaishali Santosh Tidake

UNDER THE GUIDANCE OF

Dr. Shirish Shrikrishna Sane

RESEARCH CENTRE

DEPARTMENT OF COMPUTER ENGINEERING
MATOSHRI COLLEGE OF ENGINEERING AND RESEARCH CENTER
NASHIK, INDIA

AUGUST 2020

A NOVEL ALGORITHM FOR MULTI-LABEL CLASSIFICATION

A THESIS SUBMITTED TO

SAVITRIBAI PHULE PUNE UNIVERSITY

FOR THE AWARD OF DEGREE OF

DOCTOR OF PHILOSOPHY (PH.D.)

(COMPUTER ENGINEERING)

IN THE FACULTY OF

SCIENCE AND TECHNOLOGY

SUBMITTED BY

Mrs. Vaishali Santosh Tidake

UNDER THE GUIDANCE OF

Dr. Shirish Shrikrishna Sane

RESEARCH CENTRE

Matoshri Education Society's

DEPARTMENT OF COMPUTER ENGINEERING
MATOSHRI COLLEGE OF ENGINEERING AND RESEARCH CENTER NASHIK, INDIA

AUGUST 2020

I dedicate this thesis to My Beloved Parents, Loving Husband and Cheerful Children ...!

MATOSHRI COLLEGE OF ENGINEERING AND RESEARCH CENTER, NASHIK

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that, the work incorporated in the thesis, **"A Novel Algorithm for Multi-label Classification"** is submitted by **Mrs. Vaishali Santosh Tidake** for the **Doctor of Philosophy (Ph.D.)** in **Computer Engineering, Savitribai Phule Pune University**, has been carried out by the candidate at **Department of Computer Engineering, Matoshri College of Engineering and Research Centre, Eklahare, Nashik** during the period from **August 2014 to August 2020** under the guidance of **Prof. Dr. Shirish S. Sane.**

Prof.(Dr.) Shirish S. Sane
Research Guide

Prof.(Dr.) Varsha H. Patil
Head of Department
Computer Engineering

Prof.(Dr.) Gajanan K. Kharate
Principal

CERTIFICATE OF GUIDE

This is to certify that, the work incorporated in the thesis **"A Novel Algorithm for Multi-label Classification"** submitted by **Mrs. Vaishali Santosh Tidake** was carried out by the candidate for the Doctor of Philosophy (Ph.D.) degree at **Department of Computer Engineering, Matoshri College of Engineering and Research Center, Eklahre, Nashik** during the period from **August 2014 to August 2020** under my direct supervision and guidance.

Date: 24/08/2020
Place: Nashik

Prof. Dr. Shirish S. Sane
Guide

FACULTY OF SCIENCE AND TECHNOLOGY, SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE - 411007

CERTIFICATE

This is to certify that, the viva-voce of **Mrs. Vaishali Santosh Tidake** (Ph.D. candidate in Computer Engineering) was conducted in the thesis entitled **"A Novel Algorithm for Multi-label Classification"**, at **Savitribai Phule Pune University**, Pune on / / 2020

Guide
Prof. Dr. Shirish S. Sane

External Referee
Dr.

Chairman
Dr.

Place: SPPU, Pune
Date:

DECLARATION BY THE CANDIDATE

I hereby declare that the thesis entitled **"A Novel Algorithm for Multi-label Classification"** submitted by me for the degree of **Doctor of Philosophy** is the record of work carried out by me during the period from **AUGUST 2014** to **AUGUST 2020** under the guidance of **Dr. Shirish S. Sane** (Research Guide) and has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of Higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

Date: 24/08/2020
Place: Nashik

Vaishali Santosh Tidake
(Eligibility No.: 12015232876)

ABSTRACT

Multi-label classification has gained significant importance due to its wide range of applications in the recent past and thereby attracted researchers too. In this kind of classification, a classifier model is trained, and once trained, it assigns a set of one or more predefined labels for a given unknown sample. It is carried out using either a data transformation approach or algorithm adaptation approach or a hybrid approach. The data transformation approach utilizes traditional classifier algorithms by transforming the data and may, therefore, lose correlations amongst labels and generally provide lesser prediction accuracy. On the other hand, the algorithm adaptation approach alters the classification algorithms rather than the data and thus provides better prediction accuracy as compared to data transformation. The third approach combines existing methods.

k-nearest neighbors (kNN) is one of the popular choices for algorithm adaptation based multi-label classification. The kNN-based multi-label classification method uses information extracted from neighbors of multi-label instances to perform classification. However, existing kNN based approaches reported in the literature have explored only feature similarity while searching the neighbors in multi-label data. ML-kNN is one such existing algorithm that provides better predictive accuracy compared to all other existing algorithms.

An instance in multi-label data is associated with a set of labels. Thus, label correlation may play a crucial role in the classification process. Therefore, a newer method may be designed that will incorporate not only the feature similarity but also the label dissimilarity while determining the neighbors.

The thesis presents research work that proposes a novel kNN based algorithm called Multi-Label Classification using Feature similarities and Label Dissimilarities (MLFLD). It is based on the computation of feature similarity and labels dissimilarity. The proposed algorithm assigns weights to the neighbors. The weight of a particular neighbor of an instance is either incremented or decremented based on the features and labels of the neighbor and the example under consideration. The computed weight is considered during the selection of neighbors.

Experiments are carried out to test and compare the performance of the proposed algorithm with the existing ones. Performance testing is carried out using i) cross-validation on five benchmark datasets, ii) using the train-test method on thirteen smaller benchmark datasets, and iii) two large benchmark datasets using in all ten standard performance

measures. From the performance analysis, it is seen that the proposed method outperforms existing data transformation based and algorithm adaptation based algorithms, including ML-kNN.

Algorithm MLFLD, although, outperforms existing approaches, it is observed that it is unsuccessful in predicting any relevant labels for a few instances and thus resulted in a Not a Number (NaN) value for a few performance measures. Further, this work presents an extended version of algorithm MLFLD, called MLFLD-MAXP. This algorithm overcomes the issue of NaN and thus also enhances the classification performance.

Algorithms MLFLD and MLFD-MAXP, when tested with cross-validation, show significant performance improvements. It is observed that both algorithms are sensitive to outlier data as like existing algorithms such as ML-kNN.

Generally, the Euclidean distance measure is used for the computation of feature similarity. Both algorithms are tested to observe the effect of different distance measures for not only feature similarity but also label dissimilarity. It is noted that with cross-validation using the algorithm MLFLD-MAXP, Manhattan and Jaccard triplet performed better in terms of average rank obtained over ten metrics, whereas, for train-test, MLFLD-MAXP, Euclidean and Hamming triplet is found to be better. Also it is noticed that the use of Manhattan distance needed the least amount of computation time while Minkowski needed maximum computation time. The computation time needed in the case of Euclidean distance is moderate as expected.

The use of multi-label feature and/or instance selection algorithms for prep-processing is found to be beneficial, as in the case of single-label classification. Use of either instance selection using sampling or combined multi-label instance and feature selection provides significant performance enhancements with lesser training time.

MLFLD and MLFLD-MAXP presented in this thesis thus may be potential candidates for performing effective multi-label classification. Further investigations are needed to validate the performance of the proposed algorithm using datasets with both numeric and categorical features.

ACKNOWLEDGEMENT

Ph.D. is an opportunity and experience once in a lifetime. It is tough at times and may feel like a time without end, but it teaches you a lot, and I am really happy that I have a chance to complete it. It would not have happened without all those people who directly and indirectly helped me along the way.

I express my deep sense of gratitude to my research guide **Prof. Dr. Shirish S. Sane**, Vice Principal, Professor and Head, Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik for extending his immense help and cooperation throughout my research work. His deep rooted knowledge, keen observations, insightful comments, encouragement and inspiration motivated me at all the time of research and writing of this thesis. His active and valuable guidance helped me in every step of the research work. He has given me sufficient freedom to explore research challenges of my choice and guided me when I felt lost. Without his insights, encouragements, and endless patience, this research would not have been completed.

I am also thankful to **Prof. Dr. Varsha H. Patil**, Chairman, BoS Computer Engg., SPPU, Pune and Vice Principal, Professor and Head, Department of Computer Engineering, Matoshri College of Engineering and Research Center, Nashik for valuable reviews on my research work. Thanks to her for continuous support and motivation.

I also express my special thanks to **Prof. Dr. G. K. Kharate**, Principal, Matoshri College of Engineering and Research Center, Nashik for providing valuable reviews as well as infrastructure and research facilities needed to carry out this research work. Without his precious support it would not be possible to conduct this research.

I express my deep gratitude to **Hon. Smt. Nilimatai Vasantrao Pawar**, Sarchitnis, MVP Samaj, Nashik. I am thankful to **Dr. N. S. Patil**, Principal and **Dr. V. S. Pawar**, Head, Dept. of Computer Engineering and all staff of MVPS's Karmaveer Adv. Baburao Ganpatrao Thakare College of Engineering, Nashik for supporting me during tenure of my research work.

I deeply acknowledge **Prof. N. M. Shahane**, **Dr. S. M. Kamlapur**, **Prof. N. L. Bhale**, **Dr. S. A. Bhavsar** and **Dr. Mubin Tamboli** for their help during my research work.

I would also like to thank Research Promotion Scheme (RPS), BCUD, Savitribai Phule Pune University for providing necessary resources in a research project. Many thanks to all officials of SPPU, specially **Mrs. Aparna Chavan**, for their cooperation.

I dedicate this Ph.D. research work to my beloved Parents **Mr. Bhaskarrao R. Kathe** and **Mrs. Pushpa B. Kathe**. They always encouraged me for everything. I would like to thank my Father in law **Mr. Govind M. Tidake** and Mother in law **Mrs. Radhabai G. Tidake**, my brothers **Prashant** and **Ganesh** and all my family members and friends for continuous support.

Last but not the least, I am heartily thankful to my Husband **Santosh** and my lovely children **Aditya** and **Anushree** for their constant encouragement and moral support, their love, inspiration, patience at all times, and to fill my life with joy and happiness.

Vaishali Santosh Tidake

Table of Contents

List of Figures

List of Tables

Abbreviations

BR	Binary Relevance
CC	Classifier Chain
CLR	Calibrated Label Ranking
CV	Cross-validation
DT	Decision Tree
ECC	Ensemble of Classifier Chain
Ex-F1	Example-based F1
kNN	k-Nearest Neighbours
LP	Label Powerset
MAP	Maximum a posteriori
Macro-F1	Macro-averaged F1
Micro-F1	Micro-averaged F1
MIML	Multi-instance multi-label learning
MISL	Multi-instance single-label learning
ML	Multi-label
MLC	Multi-label Classification
MLDB	Multi-label Dataset
ML-DT	Multi-label Decision Tree
MLFLD	Multi-label Classification using Feature Similarities and Label Dissimilarities
MLFLD-MAXP	MLFLD with MAXimum Probability
MLFS	Multi-label Feature Selection
MLFSIS	Multi-label Feature Selection and Instance Selection
MLIS	Multi-label Instance Selection
MLkNN	Multi-label k Nearest Neighbour

MLNB	Multi-label Naive Bayes
PPT	Pruned Problem Transformation
RPC	Ranking by Pairwise Comparison
RAkEL	Random k Labelset
SimGIC	Similarity of Graphical Information Content
SimIC	Similarity of Information Content
SIML	Single-instance multi-label learning
SISL	Single-instance single-label learning
SL	Single label
SLC	Single-label Classification
SVM	Support Vector Machine
TC	Text Categorization
TrTe	Train-test splits

Chapter 1

Introduction

1.1 Preamble

Nowadays, the multi-label framework is used in a variety of domains like object recognition, scene annotation, object detection from videos, video annotation, and human attribute recognition [1]. In the field of product design, images of products are annotated by designers with multiple labels. If products are associated with description documents, then text categorization (TC) is used to categorize these images according to their description. Thus multi-label classification is also useful in such domains.

In any organization, a lot of data is generated in day to day work. Rapid growth in the area of information technology has also steered an increase in data over the last few decades. When data increases, it becomes difficult to access the desired information. So it is necessary to categorize data for proper organization and quick access. The data can be classified by either unsupervised learning or supervised learning, through clustering of unlabeled or classification of labelled data.

Classification is very commonly used task in mining. It is referred to as supervised learning as it involves a train set having known instances that are used to train the model and then tested on other known instances. A train set is a set of records. A record is expressed by a set of attributes, and it is associated with the class which represents a category of that record. The trained models are used for the classification of unknown instances. Many scenarios in day to day life reflect the application of supervised learning

[1]-[11]. For example, an image may represent a beach or a forest. A video can express a desert or a mountain. Forts have an important place in history as well as architecture.

A lot of data from different domains is already available in the form of datasets. Experts from the particular field have associated records from these domains with appropriate classes. M. L. Zhang et al. [12] have used an image dataset. These images are divided into groups according to the type which is manually assigned by relevant experts. Xin Chen et al. [10] collected problems faced by students doing engineering. They used twitter for data collection and used a specific hashtag to filter the required data. This task of manually associating labels to data is automated by classification. Classification is the task that helps to design models which can assign labels to unseen data by using knowledge gained from already labelled data.

In conventional classification, a record is associated with only one class. But in many real-world scenarios observed nowadays, a record cannot be categorized to only one label. It better reflects the situation if it is associated with one or more labels. The later scenario is termed as multi-label classification (MLC). It is a process which correlates a set of predefined categories to an unseen entity according to its characteristics.

A set of photographs can be grouped according to objects in them such as people, traffic, road, hotels, restaurants, forest, trees and much more. When relevant images are grouped, then they can be labelled with a suitable class. For example, images in the urban area can be categorized as buildings, roads, grounds. Images related to the road can be further categorized to reflect traffic scenario. It helps to percept further information by labelling traffic images to indicate whether traffic is dense or sparse.

Several videos are available on Youtube (www.youtube.com) that are tagged by multiple labels.

Just now our Government has done an announcement of the scheme for yellow ration cardholders to avail medical facilities under Ayushman Bharat Scheme. News for such announcement is related to more than one categories, the government as well as health.

Blood reports of a patient are useful to diagnose diseases. It may reveal general symptoms of one or more diseases. It helps doctors to suggest specialized pathological tests further if required.

Because of today's changing lifestyle, shopping malls are preferred by many people for purchasing where each product may be assigned more than one category. These categories help to decide which products should be kept together to increase sale. Online shopping has changed the scenario of the market. However, since many options of shopping are available to people on one click of a smartphone, it has become challenging to attract customers. Thanks to digital marketing, that keeps an eye to understand shopping habits or likings of the visitor. This collected data is utilized to categorize customers having similar taste so that launching of a new relevant product can be broadcasted to such group together.

In the text categorization (TC), text documents are categorized according to the contents of documents. Newsletters can be classified according to news involved in them. Sometimes a story may be related to multiple categories. Thus it is again an example of multi-label classification. Gmail (http://www.gmail.com) allows its users to attach various labels to an email. For instance, if person X is working in some organization, then his/her emails can be categorized as personal or official. An official email can be further classified as department-level or institute-level. As per manual of the National Board of Accreditation for Engineering Tier II, a record of students placed, doing higher studies and working as an entrepreneur is required in Criterion 4 as well as Criterion 7. BBC (http://www.bbc.com) also assigns multiple labels for a news article. News published by BBC that "First cookies baked in space oven by astronauts" is associated with space as well as environment. As announced by Indian Space Research Organization (ISRO) on 23^{rd} January 2020, "Vyom will be the first robot who will work as an astronaut for a space mission without human. It will help to monitor how the human system will behave in the environment to control life support system". This news is associated with space, environment and artificial intelligence. BBC has also labelled this news as World, Asia and India.

Thus multi-label classification has gained significant importance and application in the recent past and thereby attracted researchers too.

1.2 Taxonomy of classification

Multi-label data can be classified in various ways; namely, label-based, level-based and based on learning framework.

1.2.1 Label-based Taxonomy

Let us consider a news story to be classified. Suppose it is to be checked whether this news story belongs to health class or not. Such type of classification is called single-label classification (SLC) as there is only one category health. Suppose there are two classes, namely health and politics, and if it is to be checked whether news story under consideration belongs to either health or politics. Such type of classification is called as binary classification as it involves two categories. Let there be three classes, namely health, sports and politics. Again it is to be checked whether news story belongs to any of these classes. This type of classification is called a multiclass classification that involves more than two categories. Now sometimes it is observed that contents of a news story may be associated with more than one categories. That is, it may belong to either health category, or health as well as sports categories, or it may belong to all the three categories. Such type of classification is called as multi-label classification (MLC) [3] [2] [5] [7] [8].

Taxonomy of classification is given below:

- Single label classification
  - Every input instance is associated with only one output label.
- Binary classification
  - Label space consists of only two labels.
- Multiclass classification
  - Label space consists of more than two labels.
- Multi-label classification
  - An input instance is associated with a set of labels.

1.2.2 Taxonomy based on learning frameworks

According to the number of instances and labels associated with each other, four types of learning frameworks exist [21] as shown in Figure 1.1.

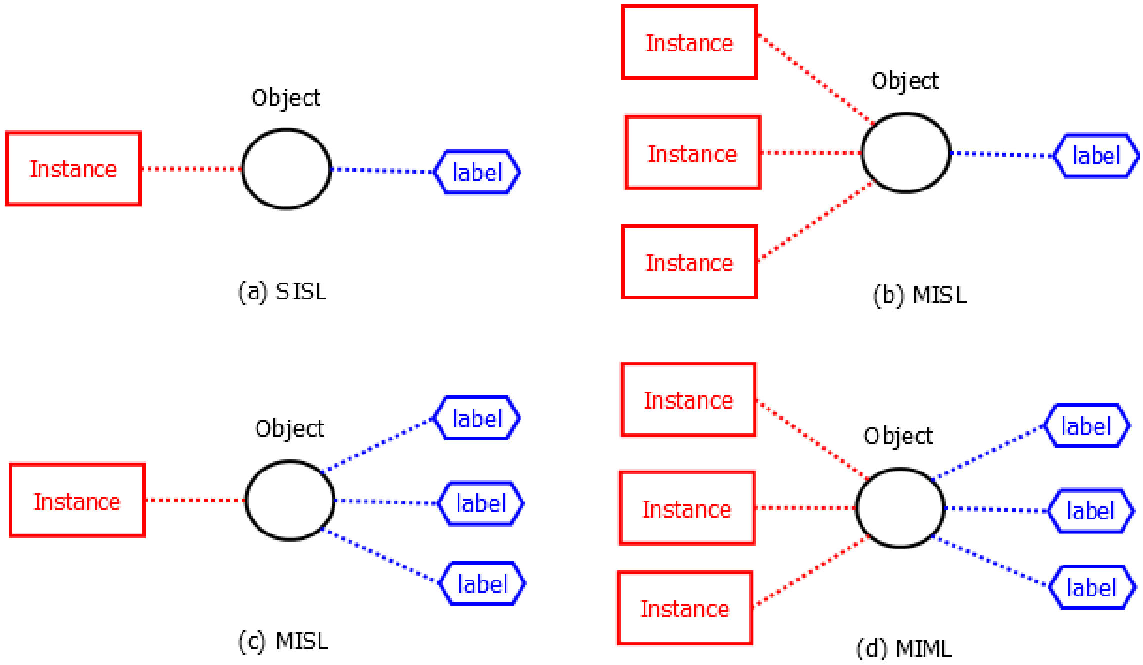


FIGURE 1.1: Learning Frameworks

1. Single-label learning (SLL or SISL)

   Single-instance single-label learning (SISL) is nothing but traditional supervised learning. It consists of one instance associated with one label, as shown in Figure 1.1(a). It can be described as

   $$f_{SLL} : X \longrightarrow L$$

   It assumes that every document represents only one semantic concept. For ex, a news story may represent either sports or education category.

2. Multi-instance learning (MIL or MISL)

   It is termed as multi-instance single-label learning (MISL). It associates many instances with a single label, as shown in Figure 1.1(b). It can be described as

   $$f_{MIL} : 2^X \longrightarrow L$$

3. Multi-label learning (MLL or SIML)

   It is termed as single-instance multi-label learning. It consists of one instance associated with a set of labels, as shown in Figure 1.1(c). It can be described as

   $$f_{MLL} : X \longrightarrow 2^L$$

It is a fact that some documents in the real world may represent more than one semantic concepts. For ex, a news story may represent both sports as well as education categories.

4. Multi-instance multi-label learning (MIML)
   MIML consists of many instances associated with a set of labels, as shown in Figure 1.1(d). It can be described as

$$f_{MIML} : 2^X \longrightarrow 2^L$$

1.2.3 Level-based Taxonomy

All the discussion held up to now considers all the labels at the same level. But there are some scenarios in the real world that are better described using hierarchies, termed as Hierarchical Multi-label Classification (HMC) [9] [38]. In HMC, all the labels in the label set are organized as a hierarchy. There exist a parent-child relationship between labels. Let an example is associated with label A. Then it is associated with all labels that appear as a parent of label A in the hierarchy. Gjorgji Madjarov et al. [77] apply clustering to the flat structure of all labels in the label set. Then this information is used by the HMC method.

1.3 Approaches for Multi-label classification

This research deals with the multi-label classification that involves three approaches [2] [4]:

- Transformation
- Algorithm adaptation
- Hybrid approach

The first approach, termed as *transformation*, alters multi-label data so that traditional classifiers can operate on such data having features and only one label at a time. But during this process of alteration of data, some information is often lost like dependency and correlation of labels. The second approach, termed as *adaptation*, modifies traditional algorithms of classifiers to tackle the multi-label data. Binary relevance (BR), label powerset (LP), classifier chains (CC) are a few methods that follow the first approach. MLkNN, ML-DT, MLNB, BPMLL, BRkNN are some methods following the second approach. A *hybrid* approach can be considered as a third one. It is a combination of multiple methods. RAkEL is a method that follows this approach.

This research deals with an *algorithm adaptation* approach.

1.4 Related concepts

There are many concepts related to MLC, like label correlation, label ranking. There exist different learning frameworks. Sometimes labels need to be arranged in a hierarchy, and that leads to hierarchical multi-label classification [95]. In this section, these concepts are described in brief.

1.4.1 Label correlation

Examples in multi-label (ML) datasets are associated with a set of labels. These label sets appear in the dataset in different combinations. If there are A, B, C and D labels in the dataset, then labels A and B appear together less number of times as compared to labels A and C. It is possible that labels A and D never occur together. This co-occurrence of labels may affect the performance of ML classifier. As seen in the text categorization paragraph discussed earlier, two space-related news are mentioned. Both are related to space and environment. Thus the possibility of space and environment categories appearing together is more than that of space and sports. When the size of the label set increases, the time complexity required to perform MLC is also affected.

1.4.2 Label ranking

Some MLC methods predict a set of relevant labels after classification. Some MLC methods may predict the relevance of each label with that instance in the form of probability. It may further be used to rank labels according to their significance with that instance. These probabilities, when split using a threshold, perform classification and when ordered, present ranking.

1.5 Multi-label classification: The current state of the art

Various approaches, like transformation and adaptation are used by many researchers to perform multi-label classification. The k nearest neighbors (kNN) is a very popular single-label classifier which follows a lazy approach. Many researchers have used statistics obtained from the k nearest neighbors of multi-label instances for classification. But as per the survey done so far, it is observed that all the approaches use the feature similarity to find the k nearest neighbors. But for the multi-label instances, the label dissimilarity also plays an important role, and hence it should also be considered. Proposed work is an attempt to study the effect of label dissimilarity while performing MLC.

1.6 Research Statement and Objectives

For the proposed research work, research statement and objectives are as follows.

1.6.1 Research Statement of the Proposed Research

To design and develop a novel algorithm for multi-label classification.

1.6.2 Objectives of the Proposed Research

The goal of the proposed research work is to develop an algorithm for preprocessing and/or multi-label classifier. Objectives of proposed research work are:

1. To study and analyze various aspects of multi-label learning.
2. To review various techniques proposed and implemented by various researchers and to identify the potential research gaps.
3. To design and develop a novel algorithm for multi-label classification.
4. To implement and test the proposed algorithm using available standard datasets.
5. To compare and to analyze the performance of proposed algorithm with the existing algorithms, and to validate the results.

1.6.3 Hypothesis

The hypothesis for the proposed research work is as follows:

- Selection of k nearest neighbors affects the performance of multi-label classifier using algorithm adaptation approach.

1.7 Contribution

Based on the survey carried out and state-of-the-art available in the area of MLC, this work proposes two algorithms, namely MLFLD and MLFLD-MAXP.

The proposed algorithm, namely Multi-Label Classification using Feature Similarities and Label Dissimilarities (MLFLD), takes into account features as well as labels to find neighbors. It assigns weights to the neighbors. When the features of two instances are similar, then the weight of that neighbor increases. But when the labels of two instances are dissimilar, then the weight of that neighbor decreases.

Proposed algorithm MLFLD with MAXimum Probability (MLFLD-MAXP) is an extension of MLFLD that behaves similar to MLFLD. However, it handles those instances where MLFLD does not assign any label for an instance under consideration.

As per our knowledge, no other work has used dissimilarity of labels so far to weigh neighbors to perform MLC using an adaptation approach.

kNN based classification makes use of distance metric. This work evaluates the effect of using three distance metrics to measure feature similarity is observed for both the proposed algorithms with three different distance measures to compute label dissimilarity. New label dissimilarity measure SimIC is also introduced in this work.

Experiments to study the effect of the feature and/or instance selection on multi-label data is also carried out. As per the literature survey carried out, no other work has performed instance selection for multi-label data.

One copyright was obtained, and four papers were published based on this work. One paper is accepted, and publication is in process. Details are given in Publication chapter at the end of thesis.

1.8 Assumptions

Multi-label classification (MLC) is a task of assigning a set of predefined labels to an unseen object according to its characteristics. This work aims to design a novel algorithm for MLC. It follows the assumptions mentioned below.

- Multi-label (ML) data is available in the form of instances. Each instance has several features and a set of labels, called class labels.
- Each instance consists of a set of features that are associated with a set of predefined labels.
- The number of training examples may be smaller or larger.
- The number of labels is comparatively much smaller than the number of attributes in some datasets. Whereas in some datasets, the number of labels is equal to or larger than the number of features.
- Datasets consist of only numeric features.
- Class labels in all the datasets are binary.
- Class labels are at the same level in the hierarchy.

These assumptions are followed by a multi-label classifier that uses knowledge obtained from labelled data to predict labels for unlabeled data.

1.9 Thesis Organization

The thesis organization is shown in Figure 1.2. Chapter 2 gives the introduction of Multi-Label Classification (MLC) along with the necessary notations used throughout the thesis. It also describes performance metrics used for the evaluation of multi-label learning along with datasets and tools used for the same. Chapter 3 describes some related work done by various researchers to perform MLC. Description of the proposed algorithms MLFLD and MLFLD-MAXP is given in chapter 4. Chapter 5 contains details of the experimental setup and datasets used. Aspects of the experimentation performed using the proposed

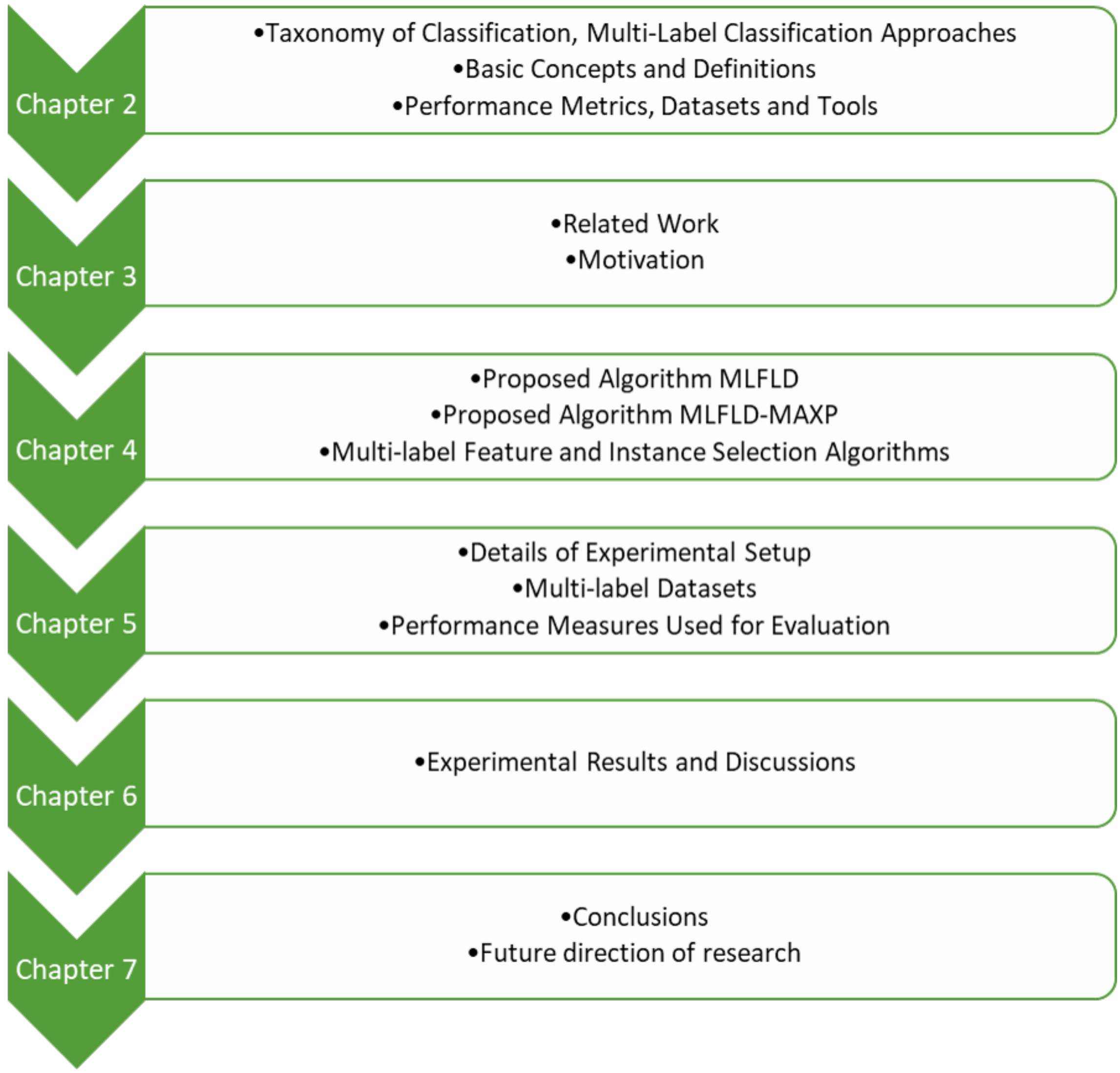


Figure 1.2: Structure of Thesis

algorithms along with a comparison of the performance is covered in chapter 6. Chapter 7 gives the concluding remarks about the work and recognizes some future directions in the related area.

Chapter 2

Multi-label Classification

Rapid growth in the area of¬ information technology has generated a lot of digital data. Classification of this data is essential to get specific information whenever required. Earlier, classification was used only for text categorization (TC). Later on, it is used in various fields like annotation of images, audio and video, biology and advertising [3]-[6].

2.1 Introduction

Multi-label Classification (MLC) is an act of allotting a set of predefined labels to an unseen entity by observing its characteristics. For example,

- A news story may represent both sports as well as education categories.
- A patient's data may represent the possibility of one or more diseases.
- An image may be annotated by sunset, sky and sea.
- A drug compound may be useful for the treatment of multiple diseases.

Classification is the most popular supervised data analysis approach, and machine learning is widely used for it from many decades [36]. MLC also follows a supervised learning approach [21]. It has been used in various applications, as listed in Table 2.1. Some of them are text categorization, image classification, graph classification, bioinformatics, functional genomics, emotion recognition, scene classification, semantic indexing of articles, mining

Table 2.1: Reported applications of multi-label learning

Application	Reported in
Text Categorization (TC)	[3], [2], [7], [13], [16], [44]
Image Classification	[4], [9], [16]
Graph Classification	[5]
Bioinformatics	[6], [16], [67]
Functional Genomics	[7]
Emotion Recognition	[8], [16]
Scene Classification	[9], [72]
Semantic Indexing of Biomedical Articles	[3]
Understand Students Learning Experiences	[10]
Parallel Tasks	[11]
Multimedia Annotation	[14], [16]

social media, parallel tasks, multimedia annotation and many more [3]-[13]. In last two decades, several research papers, books and PhD theses have been published about MLC [3]-[79], and various survey papers [15]-[21] are also available for the same.

Section 2.2 shows the taxonomy of classification and comparison of conventional and multi-label classification. Taxonomy of MLC, its basic approaches and methods that follow these approaches is described in sections 2.3 and 2.4. Section 2.5 shows another taxonomy of MLC according to dependency. Sections 2.6-2.8 talk about performance metrics, datasets and tools respectively.

2.2 Taxonomy of Classification

Classification is a process of assigning a class to an unseen object based on its features. It is a supervised learning approach. In general, classification task can be categorized according to a total number of labels in label space and number of labels that can be associated with an instance. Accordingly, the taxonomy of classification is given below:

- Single label classification
  - Every input instance is associated with only one output label.
- Binary classification
  - Label space consists of only two labels.
  - Ex. a news story may represent either sports or education category.

- Multiclass classification
  - Label space consists of more than two labels.
  - Ex. a news story may represent one of sports, politics or education categories.
- Multi-label classification
  - An input instance is associated with a set of labels.
  - Ex. a news story may represent both sports as well as education categories.

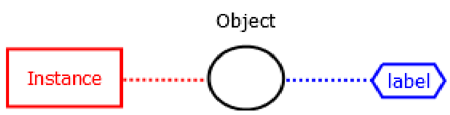
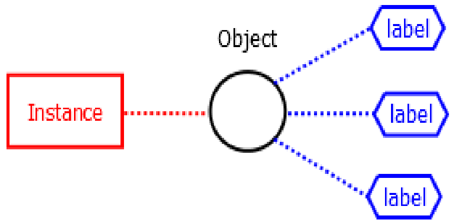
Assigning a single category to each input example is termed as single label (SL) classification or just classification. According to the total count of categories involved, SL classification can be either BSL or MSL. BSL (binary single-label) classification when the label space has only two categories. MSL (multiclass single label) classification if the label space includes more than two categories. Suppose students are asked about the topics of their interest among Cloud Computing (A), Big Data (B) and IoT (C). Then students' may reply as follows: some students like only A, some like only B, and some like only C. There are some students who like A and B, or B and C, or A and C, or all the three subjects. This scenario represents multi-label data and is handled by multi-label classification (MLC) [5]. Different algorithms are available to handle SL problems. But various applications need MLC such as TC, the discovery of the drug, tag recommendation, prediction of gene function [3]-[11] etc. Hence it is gaining the position of an upcoming research field in the area of machine learning.

As this thesis focuses on multi-label classification, its comparison with traditional classification is presented here in Table 2.2 [21] [77].

2.3 Taxonomy of Multi-Label Classification (MLC)

MLC is classified by researchers differently. In 2007, Grigorios T. and Ioannis K. [15] categorized existing MLC techniques into transformation and adaptation, as shown in Table 2.3(a). Their hierarchy is shown in Figure 2.1 [15]-[21]. As the name indicates, transformation involves the conversion of data from multiple labels to a single label (SL) followed by single-label classification (SLC). The adaptation category involves modification of basic single-label algorithm to process multiple label data directly. In 2009, Grigorios

TABLE 2.2: Single-label versus Multi-label Classification

Sr. No.	**Single-label Classification (SLC)**	**Multi-label Classification (MLC)**
1	One instance is associated with one label.	One instance is associated with a set of labels.
2	Also called as Single-instance single-label learning (SISL).	Also called as Single-instance multi-label learning (SIML).
3	$f_{SLC} : X \rightarrow L$	$f_{MLC} : X \rightarrow 2^L$
4	Every object represents only one semantic concept.	An object represents one or more semantic concepts.
5	Ex. a news story represents either sports or education category.	Ex. a news story represents both sports as well as education categories.
6		

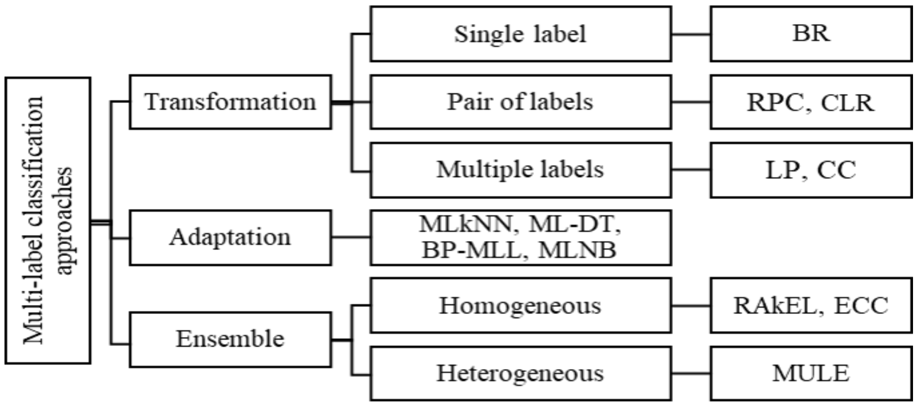


FIGURE 2.1: Taxonomy of multi-label classification approaches

T. et al. [16] further categorized transformation depending on several labels considered at a time. These methods use a single label, a pair of labels, or multiple labels at a time. These three methods are termed as first, second and high-order strategy, respectively by M. L. Zhang et al. [20]. Some researchers followed one more approach, namely ensemble methods. These methods combine several MLC methods in different ways [16][19].

In 2009, Andre et al. [17] categorized MLC methods based on the dependency of the algorithm. They formed two categories, namely an algorithm independent method and an algorithm dependent method, as shown in Figure 2.2 [15]-[21]. The reported literature, according to this taxonomy shown in Figure 2.2 is listed in Table 2.4.

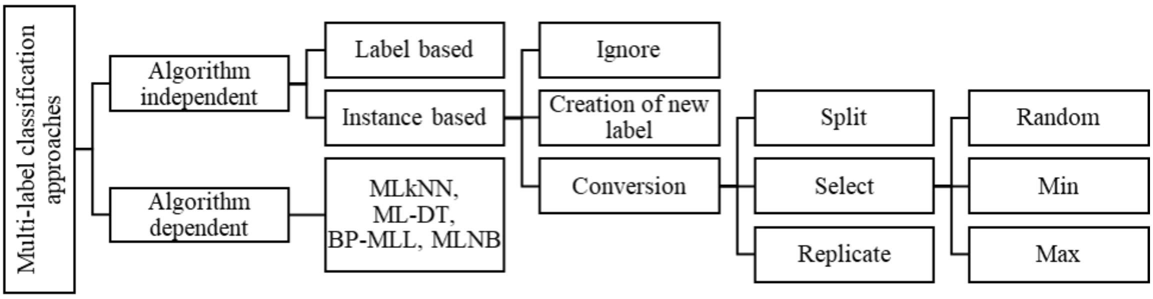


FIGURE 2.2: Taxonomy of multi-label classification methods according to dependency

TABLE 2.3: Classification of reported algorithms based on the approach

MLC approach	Reported in
Transformation	[6], [8], [10], [11], [15]-[23], [29], [30], [32]-[35], [37], [44]-[43] [49], [56], [60], [61], [63], [64], [71], [80]
Adaptation	[3], [7], [15]-[23], [12], [26], [29], [32]-[35], [37], [42]-[43] [56], [60], [63], [64], [71]
Ensemble	[3], [16], [19], [20], [29], [31], [35], [64]

TABLE 2.4: Classification of reported methods based on dependency

Multi-label classification approach	Reported in
Algorithm independent methods	[6], [15]-[17], [20], [68]
Algorithm dependent methods	[6], [15]-[17], [20], [68]

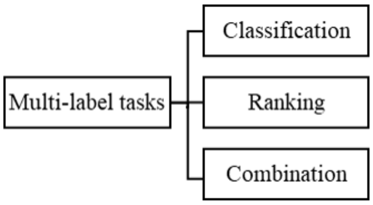


FIGURE 2.3: Taxonomy of Multi-label tasks

2.3.1 Taxonomy of Multi-label tasks according to output

MLC can be categorized according to tasks performed during learning, as shown in Figure 2.3. These tasks are, namely, classification and ranking [18] [19]. In the classification, labels are divided into two groups, namely relevant and irrelevant. In contrast, in the ranking, a sequence of all the labels is generated in the order of their relevance. One more task can be considered that combines the functionality of both ranking and classification [20]. It partitions as well as ranks the labels. According to the learning task, a suitable metric can be used for evaluation, as discussed in section 2.5. These tasks are elaborated in brief as follows.

Let $L = \{L_1, L_2, L_3, L_4, L_5\}$ be a set of disjoint class labels.

1. Classification
   - It partitions the label set L into two sets: a set of relevant (positive) labels and a set of irrelevant (negative) labels.
   - It outputs a set of positive labels P. Then negative labels can be obtained by set difference L–P.
   - For ex, positive labels $L = \{L_3, L_4, L_5\}$ and negative labels $L = \{L_1, L_2\}$.
2. Ranking
   - It produces an order of all the labels in L.
   - It is expected that the ranking of positive labels should be higher than that of negative labels.
   - For ex, $rank(L_5) > rank(L_3) > rank(L_4) > rank(L_2) > rank(L_1)$
3. Combination of classification and ranking
   - It outputs the ranking of positive labels.
   - For ex, $rank(L_5) > rank(L_3) > rank(L_4)$

Gjorgji Madjarov et al. [19] have evaluated twelve ML algorithms using eleven ML datasets and observed performance of sixteen metrics. The efficiency of algorithms is also analyzed. Authors checked statistical significance with Nemenyi and Friedman tests.

2.4 The state-of-the-art Multi-label (ML) methods

According to taxonomy given in Figure 2.1, and Figure 2.2, the state-of-the-art ML methods are discussed in brief in this section.

2.4.1 Transformation

As the name indicates, transformation involves transferring the data to change its multi-labelled nature to single-label so that it can be dealt with SLC. These methods can be classified further according to the number of labels considered by a classifier. These methods use a single label, a pair of labels, or multiple labels at a time. Accordingly, they are termed as first, second and high-order strategy respectively by M. L. Zhang et al. [7] [16] [20] [21] [12] [28] [53] [57].

In this section, some of the methods used for transformation approach are explained in brief.

2.4.1.1 Single-label approaches

Methods which follow a single-label approach for transformation consider only one label at a time. BR and Ignore/Select are the methods which support this approach.

Ignore/Select: These methods either remove an instance with multiple labels or select one label and associate it with that instance, respectively. These methods are referred to as ranking via single-label learning in the literature [16]. They are explained with an example as follows.

- Ignore
  - As the name indicates, it merely ignores all multi-label examples.
  - For ex., instance 2 with two labels is removed from the dataset (Figure 2.4).
  - Cons: Lot of information is lost.

Instance Number	Labels
1	L1
2	**L1, L2**
3	L2
4	L1

Instance Number	Labels
1	L1
3	L2
4	L1

FIGURE 2.4: Example of Ranking via Single-Label Learning (Ignore)

- Select
  - For instance, having two or more labels, it selects and associates only one label to that instance and rejects the remaining labels.

- Criteria for selection of label can be a minimum occurrence of a label (Min), the maximum occurrence of a label (Max) and random occurrence of a label (Random) in the whole dataset.
- For ex., in *Min Select*, instance two is associated with L_3 as it occurs twice, and L_4 occurs thrice. Similarly, in *Max Select*, instance two is associated with L_4, which occurs maximum times, and in *random select* L_4 is selected randomly (Figure 2.5).
- Cons: Information loss

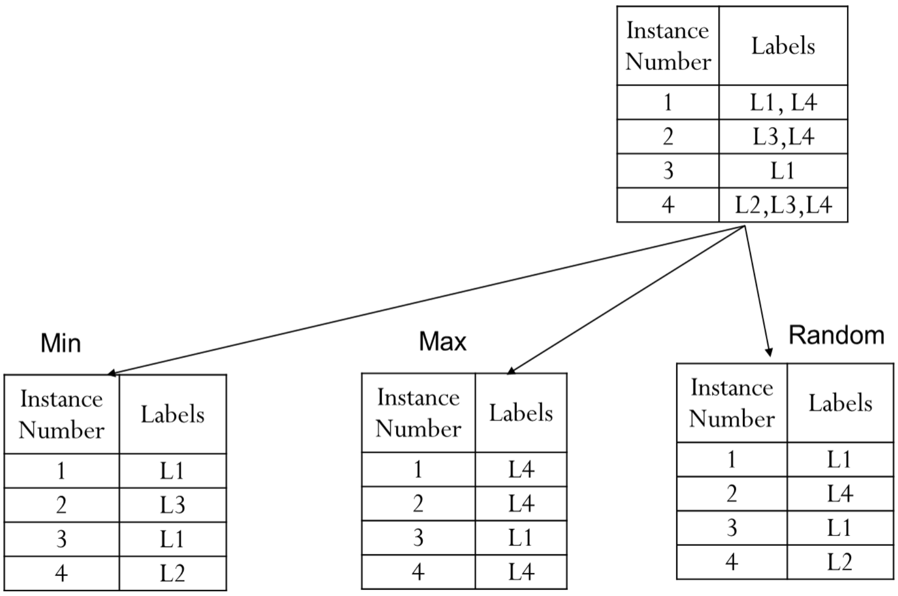


FIGURE 2.5: Example of Ranking via Single-Label Learning (Select)

Copy and Copy-Weight (Entropy):

- Both ignore and select methods face a problem of information loss.
- Hence copy method is used to replace each example (x_i, y_i) with $|y_i|$ examples.
- For ex., instance two is associated with two labels. It is replicated twice, once with L_3 and other with L_4.
- In copy-weight, $1/|yi|$ weight is also assigned to all replicated instances.
- For ex., all replicas of instance two are assigned weight 0.5 (Figure 2.6).
- Pros: No information loss

Instance Number	Labels
1	L1, L4
2	L3,L4
3	L1
4	L2,L3,L4

Instance Number	Labels	Weights
1x	L1	0.5
1y	L4	0.5
2x	L3	0.5
2y	L4	0.5
3	L1	1.0
4x	L2	0.33
4y	L3	0.33
4z	L4	0.33

FIGURE 2.6: Example of Ranking via Single-Label Learning (Copy-weight)

- Drawback: Increased number of examples/instances

Binary Relevance (BR): Consider there are three labels $C_x, C_y and C_z$, respectively. Then BR designs three separate classifiers where each classifier handles these three labels independently (Figure 2.7).

Instance Number	Labels
1	L1
2	L1,L2
3	L2
4	L1

(a) Multi-label table with L={L1, L2}

Instance Number	L1
1	T
2	T
3	F
4	T

(b) Classifier output for label L1

Instance Number	L2
1	F
2	T
3	T
4	F

(c) Classifier output for label L2

FIGURE 2.7: Example of Binary relevance

As many traditional methods are available to handle individual label, anyone method can be picked. Finally, for the classification of new data, the results of all the three classifiers for three labels are considered. The cons of the technique is that relation among different labels is simply ignored [19]-[23]. But it has many useful features also. As it treats each label independently, the classifier model can be easily updated dynamically if the label set is appended with a new label and scales linearly with the number of labels. Also, it is beneficial to handle active data. The classifier model can run in multiple parallel classifiers for different labels. Due to so many features and ease of design, BR is very popular and widely used.

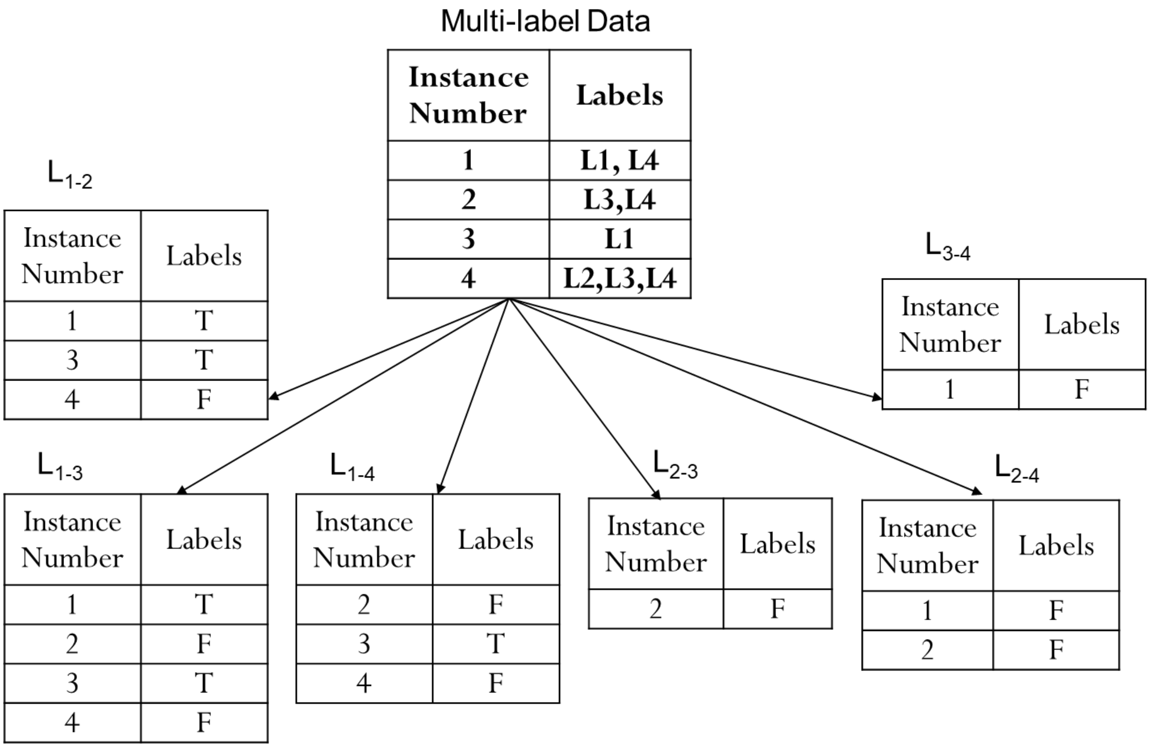


FIGURE 2.8: Example of RPC

2.4.1.2 Pair of labels

RPC and CLR are the ML classifiers that consider a pair of labels together instead of a label. These two methods follow a transformation approach.

Ranking by Pairwise Comparison (RPC): As the name suggests, RPC considers a pair of labels at a time. If there are m classes in the data, then $(m \times (m-1))/2$ pairs of classes can be formed. A separate classifier for each pair is constructed in RPC [16][78]. Each C_{p-q} classifier considers instances having either class C_p or C_q. All the instances having neither C_p nor C_q classes are ignored. The instance associated with C_p or C_q is marked as 1 or 0, respectively. Then classes are ranked as per votes received from all C_{pq} pair models [18] [22]. For ex., Figure 2.8 shows actual data having four multi-label instances related to four labels $L_{1...4}$. For four labels, six combinations viz. $L_{1-2}, L_{1-3}, L_{1-4}, L_{2-3}, L_{2-4}$ and L_{3-4} exist as shown in Figure 2.8. Suppose new instance is classified by these six models and they give votes shown in Table 2.5. Counts of these votes are used to rank labels, as shown in Table 2.6. For ex., L_2 having maximum votes is listed first, indicating most relevant label for instance, under consideration.

TABLE 2.5: Votes of RPC models for a new instance

Model	L_{1-2}	L_{1-3}	L_{1-4}	L_{2-3}	L_{2-4}	L_{3-4}
Votes	L_2	L_1	L_1	L_2	L_2	L_4

TABLE 2.6: Total votes and rank of labels by RPC

Labels	L_1	L_2	L_3	L_4
Total votes	2	3	0	1
Rank of labels	Rank 2	Rank 1	Rank 4	Rank 3

Calibrated Label Ranking (CLR): From Table 2.6, it can be observed that RPC generates the ranking of all labels.

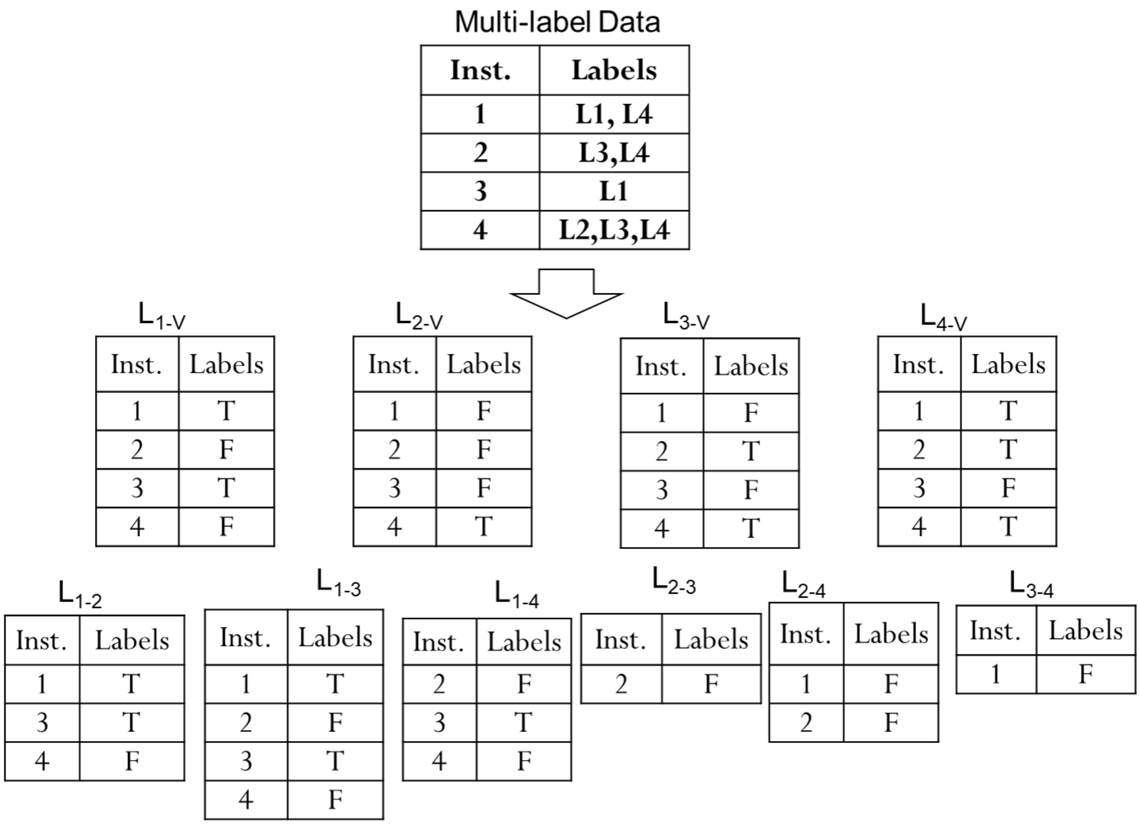


FIGURE 2.9: Example of CLR

Relevant and irrelevant labels are not distinguished separately. This drawback is overcome in the CLR method. It adds a virtual (imaginary) label [78] to the existing label set of size m in the original data. Rest of the operations are same as RPC (Figure 2.9). As a result, the ranking of $(m+1)$ labels is obtained where an imaginary label separates relevant labels from irrelevant labels [18]-[22]. Table 2.7 shows votes received from $((m+1) \times m)/2$ models for unseen instance. Table 2.8 shows relevant labels L_2, L_1 having rank higher than that of virtual label L_v and irrelevant labels L_4, L_3 having rank lower than that of L_v.

Table 2.7: Votes of CLR models for a new instance

Model	L_{1-2}	L_{1-3}	L_{1-4}	L_{2-3}	L_{2-4}	L_{3-4}	L_{1-V}	L_{2-V}	L_{3-V}	L_{4-V}
Votes	L_2	L_1	L_1	L_2	L_2	L_4	L_1	L_2	L_V	L_V

Table 2.8: Total votes and rank of labels by CLR

Labels	L_1	L_2	L_3	L_4	L_v
Total votes	3	4	0	1	2
Rank of labels	Rank 2	Rank 1	Rank 5	Rank 4	Rank 3

2.4.1.3 Multiple labels

A multi-label instance is associated with a set of labels in most of the cases. If all the labels or its subset is used to build a classification model, then the relationship between labels is utilized, and better performance can be achieved. LP, RAkEL, CC and ECC are based on this concept. RAkEL, CC and ECC use a subset of labels whereas LP uses a set of all the labels of each instance.

Label Powerset (LP): As mentioned in section 2.4.1, creating a new label is also one method to handle instances associated with more than one label. LP [18] [19] [21]-[23] uses same approach. Every distinct combination of labels associated with instances is treated as a new class. Now, this data represents multiclass data which can be handled by conventional classifiers. Thus relationship among labels is considered by processing multiple labels simultaneously, and this handles the disadvantage of BR. Sometimes many combinations of labels are present in the original data. It generates many classes. The problem occurs when few classes are associated with comparatively less number of instances. Accuracy is hampered when some classes possess very few instances among others. For the unseen data, the model predicts the most probable set of labels. Again the issue with this method is that it can predict only label sets existing in the original data. Multi-label data in Figure 2.10 is related to four labels. Hence each instance in transformed data is related with a set of four labels $\{L_1, L_2, L_3, L_4\}$ where each $L_i \epsilon \{0, 1\}$. For ex, instance one is associated with L_1 and L_4 represented by 1 and L_2 and L_3 represented by 0, thus forming label set 1001.

PPT (Pruned Problem Transformation): As seen in LP, some label sets may possess very few instances among others. It hampers accuracy. Such a problem is overcome in PPT [18] [29]. All those instances are removed that have label sets occurring in the dataset number of times less than a threshold. Such instances are replaced by instances

Instance Number	Labels
1	L1, L4
2	L3,L4
3	L1
4	L2,L3,L4

$\Rightarrow$

Instance Number	Labels
1	1001
2	0011
3	1000
4	0111

FIGURE 2.10: Example of Label Powerset (LP)

having disjoint subsets of that label set. Again it is checked whether newly added instances with disjoint label sets occur several times higher than a threshold, then they are considered otherwise discarded.

Sr. No.	Labels	Occurrence Count
1	L1, L4	8
2	L3, L4	7
3	L2	16
4	L1,L3,L4	2

$\Rightarrow$

Sr. No.	Labels	Occurrence Count
1	L1, L4	8
2	L2	16
3	L1	2
4	L3, L4	9

FIGURE 2.11: Example of PPT

For ex, let us consider threshold t_1 is 3. As shown in Figure 2.11, a label set $\{L_1, L_3, L_4\}$ occurs less than t_1 times. Hence all such instances are replaced by instances having subsets $\{L_1\}$ and $\{L_3, L_4\}$ respectively. As occurrence count of $\{L_1\}$ is still less than t_1, all such instances are discarded. Occurrence count of $\{L_3, L_4\}$ is more than t_1. Hence all such instances are considered. Problem with the pruning is that crucial information of infrequent label may be lost after pruning.

Random k-Label sets (RAkEL): Instead of considering all the labels of an instance together as in LP, it is possible to consider only a subset of labels at a time. This group of labels is termed as a label set by Tsoumakas G. and Vlahavas I. P. [22]. It helps to reduce the complexity of LP also. The method uses a parameter k that restricts count of labels to be used by one model. It also uses parameter m denoting the number of models to be constructed. As parameters k and m affect the performance, it is crucial to decide their values. Parameter k can take values between one and size of label space. Smaller k is observed to give better performance, whereas parameter m should not be minimal. It can take a value at least twice the number of labels as suggested in the literature [18] [20]

TABLE 2.9: Example of Random k-Label sets: Decisions for a new instance

Model	3-label sets	L_1	L_2	L_3	L_4	L_5	L_6
H_1	$\{L_1, L_2, L_6\}$	1	0	-	-	-	1
H_2	$\{L_2, L_3, L_4\}$	-	1	1	0	-	-
H_3	$\{L_3, L_5, L_6\}$	-	-	0	-	0	1
H_4	$\{L_2, L_4, L_5\}$	-	0	-	0	0	-
H_5	$\{L_1, L_4, L_5\}$	1	-	-	0	1	-
H_6	$\{L_1, L_2, L_3\}$	1	0	1	-	-	-
H_7	$\{L_1, L_4, L_6\}$	0	-	-	1	-	0
Average votes	-	3/4	1/4	2/3	1/4	1/3	2/3
Final prediction	-	1	0	1	0	0	1

[21] [22]. Label sets used by m models also affects performance. For an unseen instance, each label is predicted by averaging results obtained from m models. It should be noted that a non-existing label set may be predicted for unseen data. G. Tsoumakas et al. [22] have implemented two variants of RAkEL, one with disjoint label sets and the other with overlapping labels.

Table 2.9 shows a snapshot of predictions for a new instance by seven models using three label sets. Consider model H_1. As it uses label set $\{L_1, L_2, L_6\}$, it will vote for L_1, L_2, L_6 labels only. For new instance, final prediction for label L_1 is 1 as an average of votes is 3/4 that is above 50 percent.

Classifier Chain (CC): As discussed earlier, BR designs three independent classifiers for three labels L_x, L_y and L_z. This separate consideration of labels simplifies the task at the cost of losing label relationships. Read J et al. [22] proposed CC that handles this issue by considering three labels independently but in a particular sequence (Figure 2.12). For ex, sequence considered is L_z, L_y and L_x. So first L_z is predicted by considering all features. Next L_y is predicted considering all features and predicted L_z. Then L_x is predicted considering all features and predicted L_y. Thus relationship between labels is taken into account by each classifier. The chain of labels can be permuted in multiple ways, and that is a very crucial part in CC as it directly affects its accuracy. It also dictates the inability of parallelizing the process [20] [21] [23]. Read J. et al. has introduced performance measure log loss in [23] that uses certainty of prediction. Jesse Read et al. [39] also proposed probabilistic classifier chains (PCC). It uses Naïve Bayes to yield probabilistic output.

Inst.	Features F	Labels
1	F1	L1
2	F2	L1,L2
3	F3	L2
4	F4	L1

(a) Multi-label data with L={L1, L2}

Inst.	Input	Output
	F	Prediction of L1
1	F1	T
2	F2	T
3	F3	F
4	F4	T

(b) Classifier output for label L1

Inst.	Input		Output
	F1	Prediction of L1	Prediction of L2
1	F1	T	F
2	F2	T	T
3	F3	F	T
4	F4	T	F

(c) Classifier output for label L2

FIGURE 2.12: Example of CC

Ensemble of Classifier Chain (ECC): Performance of CC is very much dependent on the chain of labels used. There can be many permutations of labels. Finding the best chain is quite tricky. Jesse Read et al. [23] handled this issue resulting in an ensemble of multiple CC models, each one using a different chain of labels. It yields better accuracy than CC. Another advantage of ECC is that it never predicts an empty label set due to various chains.

2.4.2 Adaptation

Adaptation involves an amendment of the existing single-label algorithm to handle multiple labels directly. Many researchers have amended current methods to manage multi-label data, and still, research is going on in this field. M. L. Zhang et al. [20] described the task as “fit an algorithm to the data”. These methods have amended conventional classifiers like decision tree (DT), support vector machine (SVM), Naïve Bayes (NB), neural network (NN) and k nearest neighbours (kNN) to use multiple label data directly without conversion [15]-[21].

This section describes some methods used for algorithm adaptation in brief.

Multi-Label k Nearest Neighbors (ML-kNN): M. L. Zhang et al. [12] proposed this algorithm that is designed by adapting conventional kNN. For ex., let k neighbors for instance X are computed. Then neighbors of an instance X belonging to each label C_m of X is counted. Also, neighbors of an instance X belonging to each label C_m not belonging to X are counted. Next likelihood probability is computed using these counts. Prior probabilities are also obtained from the training set by counting instances having label C_m and not having label C_m respectively. Next labels of a new instance are obtained using

Maximum a posteriori that is based on Bayes theorem [13] [10]. The posterior probability for each label C_m is then computed for an unseen instance. Experimentation is performed on three datasets. MLkNN [18]-[22] [12] has proven to be the state-of-the-art algorithm, although it has one limitation of not considering a label relationship.

Backpropagation Multi-label neural network (BP-MLL): M. L. Zhang et al. [7] proposed an algorithm that is derived from a conventional neural network. This algorithm is modified to deal with multiple labels. It uses backpropagation. It aims to design an error function that generates the rank of relevant labels higher than that of irrelevant labels for each multi-label instance X_m. Each instance X_m contributes to computing the error. It determines the output of the neural network for each relevant label of X_m and that for an irrelevant label of X_m. Difference between these two values is used further. Thus multi-label data is used to compute errors, and the information is fed back such that errors are minimized. Performance is evaluated using Yeast dataset. Ensemble of BP-MLL is suggested by authors for performance improvement [18] [19] [22] 78.

ML-C4.5: A. Clare and R. King [26] developed a new multi-label algorithm based on the C4.5 algorithm for decision tree [27]. The reason was, they found that the phenotype data of yeast is multi-label. Some genes may belong to multiple functional classes. Hence multi-label rather than multiclass classification fits here properly. For this purpose, the authors introduced ML-IG technique. The technique calculates entropy for each class. The probabilities of a class are calculated from the information of instances belonging and not belonging to each class. This information decides the attribute to be used for partitioning the dataset at each node. Most important thing to note is that a set of labels rather than a single label is assigned to the leaf node of the tree. The resulting tree is also useful to generate rules for classes which are easy to understand [17] [19] [20] [22] [26] [27]. As the yeast dataset is small, authors have used the bootstrap method [79] for sampling so that more samples of data can be created and consequently, more rule sets can be generated.

MLNB: Zhang M. L. et al. [28] presented a basic version of modified Naïve Bayes MLNB. It estimates posterior probability from prior and conditional probabilities of each class. Authors also presented two extensions of MLNB, one using principal component analysis (PCA) and the other using genetic algorithm (GA). Both extensions perform feature selection before MLNB. Finally, PCA followed by GA is also used for selecting features. In GA, fitness is assessed by averaging hamming and ranking loss. Ten-fold cross-validation is used for experimentation on twelve synthetic datasets and two real-world datasets. One drawback of the algorithm is that while estimating probability, the relationship between labels is not used.

Rank-SVM: It is an adaptation of a maximum margin strategy [17] [19] [20]. It is designed using a set of C linear classifiers for C labels which are optimized to minimize empirical ranking loss. It uses the following convention: $W = \{(w_m, b_m)|1 \leq m \leq C\}$ where w_m is weight vector and b_m is the bias for m^{th} label. For instance x_m, margin is calculated using relevant and irrelevant labels of instance x_m.

2.4.3 Ensemble

Sometimes applying a classifier once to the data may not perform up to the mark. But if the same classifier is applied to the same data but with parameter variation, then it has been observed to get improved performance when results of each run are combined. This technique has been proven to perform much better and to provide improved accuracy also. RAkEL [19] [22] [71] and ECC [19] [23] follow this methodology. RAkEL and ECC are ensembles of LP and CC respectively. They are described earlier in brief. Yannis Papanikolaou et al. [3] further categorized the ensemble approach as homogeneous or heterogeneous. RAkEL and ECC [19] [30] are termed as homogeneous because they are ensembles of the same base classifiers viz. LP, CC and PPT, respectively. MULE [3] is termed as heterogeneous because it is an ensemble of different base classifiers.

2.5 Multi-label classification according to dependency

MLC methods can also be classified according to dependency, as shown in Figure 2.2. This section describes these classifiers, which are grouped as algorithm independent and algorithm dependent.

2.5.1 Algorithm independent methods

The name itself describes the nature of algorithms that follow this approach. They use traditional classifiers as a base. But traditional classifiers are single-label. So the simplest way is converting data from multi-label to single-label [15]-[18] [22]. That is these methods change data, not an algorithm. These methods are described very nicely by Zhang M. L. and Zhou Z.H. [20] as methods that "fit data to an algorithm". These methods can be grouped as label-based and instance-based. They are described as follows. Assume that a multi-label data under consideration has C labels.

- **Label-based methods:** In these methods, C base classifiers $K_1, K_2 \ldots K_C$ are used. Each K_m classifier is single-label. It considers all instances having label m as relevant and remaining instances as irrelevant. Votes for new instance are obtained from $K_1, K_2 \ldots K_C$ classifiers.
- **Instance-based methods:** These methods again can be categorized as per variations to consider instances and assign label(s) to instances.
  - **Ignore:** The simplest method to handle ML data by SL classifier is to ignore instances that have more than one label and consider only those instances that have one label only. The previous method faces the problem of data loss as it does not use ML instances.
  - **Creation of new label:** This method considers all the instances. But because base classifier is still single-label, this method represents each unique combination of labels as a new label. Thus all instances are preserved, and no data is lost.
  - **Conversion:** Name implies converting data from ML to SL. It can be done using split, select or replicate the approach.

In a ***split approach***, data is split into D_C samples. Let there be only two classes K_m and K_n. Then two splits D_1 and D_2 will be created. The instance I having both classes K_m and K_n will be added in split D_1 as (I, K_m) and D_2 as (I, K_n). But instance I having either K_m or K_n will be added in either D_1 or D_2 only with respective class.

In ***selection approach*** of conversion, class to be associated with an instance is selected. Let there be an instance I that is associated with classes K_m and K_n.

- **Random approach** assigns class K_m or K_n on a random basis to an instance I.
- **Min approach** assigns class K_m to an instance I if it appears minimum time compared to K_n in the data.

- **Max approach** assigns class K_m to an instance I if it appears maximum time compared to K_n in the data.

In the ***replicate approach*** of conversion, an instance I associated with say two labels K_m and K_n is replaced by two replicas of that instance as (I, K_m) and (I, K_n).

There is more data loss in ignore approach. Comparatively less data loss is in case of select method. No data loss is there in replicate and split approaches, but there is an increase in instances.

2.5.2 Algorithm dependent methods

During the conversion from ML to SL, the relationship between labels is lost. Hence some researchers used multi-label data as it is but designed algorithms to handle it. These methods are described very nicely by Zhang M. L. and Zhou Z.H. [20] as methods that "fit an algorithm to data". Traditional algorithms like support vector machine, decision tree, neural network, Naïve Bayes and k nearest neighbors are modified by many researchers to tackle ML data directly [15]-[22] [26]-[8]. Some methods are described in section 3.3.

2.6 Assessment of MLC algorithms

Measures used for evaluating the performance of ML algorithms are different than SL algorithms. They can be assessed based on either calculation or output of learner [16] [19] [20] [22] [71]. ML performance measures can be categorized, as shown in Figure 2.13. Those metrics that assess performance by averaging actual and predicted values of all examples under consideration are termed as example-based metrics. And those metrics that are assessed by averaging performance of all labels, which is calculated from the performance of each individual label are termed as label-based metrics [19] [20] [22].

ML learning algorithm can generate output in three different ways:

- Prediction of ***binary*** values one for each label in the label set indicating whether a particular instance is associated with that label

- ***Ranking*** of all labels as per their relevance to a particular instance

- Predicting the ***probability*** value for each label in the label space

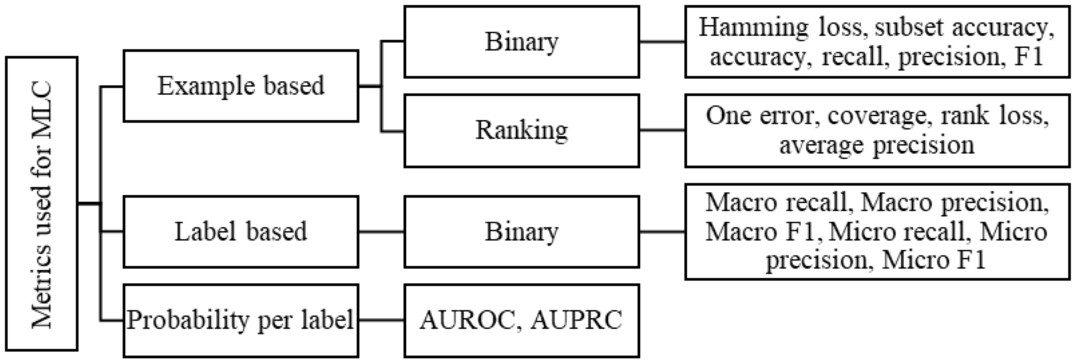


Figure 2.13: Taxonomy of performance metrics

2.6.1 Notations

Before proceeding, let us define basic terms to be used for ML tasks.

- Let us denote an ML dataset under consideration by E and label space by S.
- Let (x_m, AL_m) denote m^{th} instance of dataset E, where x_m is a record having f features, $m = 1 \ldots |E|$ and AL_m is a subset of S.
- Let g_c be a task of ML classification. Then the objective of $g_c(x_m)$ is to find PL_m that is a prediction of labels for an instance x_m.
- Let g_r be a task of ML ranking. Then the objective of $g_r(x_m)$ is to find a ranking of labels for an instance x_m.

As stated above, AL_i and PL_i denote a set of actual labels of instance x_i and a set of predicted labels by $g_c(.)$ for the same. ML classifiers are assessed using various metrics that are listed below.

2.6.2 Example-based measures

Performance measures that compute data from individual instances and then make an average of data obtained are termed as example-based measures. They can be grouped as binary and ranking.

- **Binary measures**

  Example-based measures that predict whether an instance is associated with a particular label or not are termed as binary measures. They are described here.

Hamming loss: It counts the number of times actual labels of an instance do not match predicted labels [25].

$$HL(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{|V(PL_i \Theta AL_i|)}{|S|} \tag{2.1}$$

where Θ denotes symmetric difference. $V(.) = 0$ if all predicted labels PL_i are the same as AL_i for an instance i, else it is 1. $HL(g_c) = 0$ means all instances are correctly classified. Smaller $HL(g_c)$ indicates better performance.

Subset Accuracy: It finds average from the exact match of the instance-wise actual label set and corresponding predicted label set for all the instances [14-20].

$$SA(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|}V(PL_i = AL_i) \tag{2.2}$$

where $V(.) = 1$ if AL_i and PL_i of instance I match, else $V(.) = 0$.

Recall, Precision, F-Measure and accuracy [31]:

$$Rc(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{|PL_i \bigcap AL_i|}{|AL_i|} \tag{2.3}$$

$$Pr(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{|PL_i \bigcap AL_i|}{|PL_i|} \tag{2.4}$$

$$F1(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{2 \times |PL_i \bigcap AL_i|}{|AL_i| + |PL_i|} \tag{2.5}$$

$$Acc(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{|PL_i \bigcap AL_i|}{|PL_i \bigcup AL_i|} \tag{2.6}$$

- **Ranking measures**

  All the ranking measures are also example-based [19]. They are defined in terms of ranking function, say, $\mu(.)$. Let $\mu(l, i)$ denotes relevance of label l with an instance i. Assume that smaller $\mu(l, i)$ shows the higher significance of l for i.

Ranking loss: computes whether a relevant label is ranked below a particular irrelevant label [25].

$$RL(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{1}{|AL_i|.|\overline{AL_i}|}\{(y_r, y_{ir})|\mu(y_r, x_i) \geq \mu(y_{ir}, x_i)\}| \tag{2.7}$$

Here $\overline{AL_i}$ denotes complement of a set of relevant labels of an instance i. Elements y_r and y_{ir} are members of sets AL_i and $\overline{AL_i}$ respectively. $RL(g_r) = 0$ indicates all relevant labels are ranked above irrelevant labels for all instances. Smaller $RL(g_r)$ is desired for better performance.

Coverage: It observes the list of predicted labels to find a number of steps for inclusion of all relevant labels of each instance and computes average over all the instances. The assumption is that the most relevant label appears at the start of the list. Smaller CG(gr) indicates excellent performance.

$$CG(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|} max_{y_r \in AL_i}\mu(y_r, x_i) - 1 \tag{2.8}$$

Average precision: determines an average value from all relevant labels ranked higher than a particular relevant label. More $AP(gr)$ indicates better performance.

$$AP(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{1}{|AL_i|}\sum_{y_{r1} \in AL_i}\frac{|\{y_{r2} \in AL_i|\mu(y_{r2}, x_i) \leq \mu(y_{r1}, x_i)\}|}{\mu(y_{r1}, x_i)} \tag{2.9}$$

Both y_{r1} and y_{r2} labels are relevant.

One-error: determines the number of times an irrelevant label is predicted with the top rank (measures how many times a predicted label at the top rank is not in the list of relevant labels of an instance). An optimal value for $OE(g_r)$ is zero. Smaller $OE(g_r)$, better the performance [15]-[21].

$$OE(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|} argmin_{y \in S}\mu(y, x_i) \notin AL_i \tag{2.10}$$

$V(.)$ returns 0 in case of false condition, else it returns 1.

2.6.3 Label-based binary measures

Measures that calculate average performance from that of individual labels are termed as label-based measures. They are binary measures, namely macro-averaging and micro-averaging.

- **Macro-averaging and Micro-averaging:** These are binary metrics based on a count of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) [15]-[21]. Macro (Micro) averaging gives equal importance to all the labels (instances). In other words, macro (micro) averaging finds an average across all the labels (example/label pairs). If c is a label, then macro-averaged metric V and micro-averaged metric V are calculated in general as

$$V_{ma} = \frac{1}{|S|}\sum_{c=1}^{|S|} V(TP_c, FP_c, TN_c, FN_c) \tag{2.11}$$

$$V_{mi} = V(\sum_{c=1}^{|S|} TP_c, \sum_{c=1}^{|S|} FP_c, \sum_{c=1}^{|S|} TN_c, \sum_{c=1}^{|S|} FN_c) \tag{2.12}$$

Definitions of macro-averaged and micro-averaged precision, recall, F1 and accuracy are given below from Eq. 2.13 to Eq. 2.20 [1][18].

- **Macro-precision:**

$$MaPr = \frac{1}{|S|}\sum_{c=1}^{|S|} \frac{TP_c}{TP_c + FP_c} \tag{2.13}$$

- **Micro-precision:**

$$MiPr = \frac{\sum_{c=1}^{|S|} TP_c}{\sum_{c=1}^{|S|} TP_c + \sum_{c=1}^{|S|} FP_c} \tag{2.14}$$

- **Macro-recall:**

$$MaRc = \frac{1}{|S|}\sum_{c=1}^{|S|} \frac{TP_c}{TP_c + FN_c} \tag{2.15}$$

- **Micro-recall:**

$$MiRc = \frac{\sum_{c=1}^{|S|} TP_c}{\sum_{c=1}^{|S|} TP_c + \sum_{c=1}^{|S|} FN_c} \tag{2.16}$$

- **Macro-F1:**

$$MaF1 = \frac{1}{|S|} \sum_{c=1}^{|S|} \frac{2 \times TP_c}{2 \times TP_c + FP_c + FN_c} \tag{2.17}$$

- **Micro-F1:**

$$MiF1 = \frac{2 \times \sum_{c=1}^{|S|} TP_c}{2 \times \sum_{c=1}^{|S|} TP_c + \sum_{c=1}^{|S|} FP_c + \sum_{c=1}^{|S|} FN_c} \tag{2.18}$$

MaAcc and MiAcc result in the same values. Macro and micro averaging do not affect accuracy measure.

- **Macro-accuracy:**

$$MaAcc = \frac{1}{|S|} \sum_{c=1}^{|S|} \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \tag{2.19}$$

- **Micro-accuracy:**

$$MiAcc = \frac{\sum_{c=1}^{|S|} TP_c + \sum_{c=1}^{|S|} TN_c}{\sum_{c=1}^{|S|} TP_c + \sum_{c=1}^{|S|} TN_c + \sum_{c=1}^{|S|} FP_c + \sum_{c=1}^{|S|} FN_c} \tag{2.20}$$

2.6.4 Probability per label measures

Two measures AUROC and AUPRC provide a probability for each label. In this section, these two metrics are described in brief.

- **AUROC/AUC (Area Under Receiver Operating Characteristics):** It represents the probability that a randomly chosen relevant sample will be ranked better than a randomly chosen irrelevant sample [23] [38].

$$MaAUC = \frac{1}{|S|} \sum_{c=1}^{|S|} AUC_c = \frac{1}{|S|} \sum_{c=1}^{|S|} \frac{|\{(x_1, x_2) | \mu(y_c, x_1) \leq \mu(y_c, x_2), (x_1, x_2) \in (Z_c \times \overline{Z_c})\}|}{|Z_c|.|\overline{Z_c}|} \tag{2.21}$$

where

$Z_c = \{x_i | y_c \in AL_i, 1 \leq i \leq |E|\}$

$\overline{Z_c} = \{x_i | y_c \notin AL_i, 1 \leq i \leq |E|\}$

Z_c and $\overline{Z_c}$ are sets of test instances belonging and not belonging to label c respectively.

$$MiAUC = \frac{|\{(x_1, x_2, y_1, y_2) | \mu(y_1, x_1) \leq \mu(y_2, x_2), (x_1, y_1) \in Z_i, (x_2, y_2) \in Z_{ir}\}|}{|Z_i|.|Z_{ir}|} \tag{2.22}$$

where

$Z_i = \{(x_i, y) | y \in AL_i, 1 \leq i \leq |E|\}$

$Z_{ir} = \{(x_i, y) | y \notin AL_i, 1 \leq i \leq |E|\}$

Z_i and Z_{ir} are sets of relevant and ¡instance, label¿ pairs respectively.

An optimal value for both MaAUC and MiAUC is 1. Larger MaAUC and MiAUC denote better performance [20].

- **AUPRC (Area Under Precision-Recall Curve):** A precision-recall curve is generally termed as PR curve. It plots the precision of a model as a function of its recall. Let the model predicts the probability that a new instance is positive with a threshold t to obtain the predicted class. This threshold t represents one point in PR space. For plotting a PR curve, threshold t can be varied from 1 to 0. It increases the count of positive instances predicted, thereby increasing the recall and generally decreasing (occasionally increasing) the precision. PR curve shows the predictive behavior of the model. The area between the PR curve and the recall axis is termed as "area under the PR curve (AUPRC)". Optimal value of the AUPRC is 1.

  In multi-label classification, PR curves are plotted for each class where the examples associated with the class as relevant and the remaining examples as irrelevant. Then the performance of all classes is combined using one of the two approaches:

  - *Area Under the Average PR Curve:* It converts a multi-label task into binary tasks to obtain the overall PR curve.
  - *Average Area Under the PR Curves:* It uses the weighted average of the areas under the class-wise PR curves.

Table 2.10: Performance metrics used for assessment of MLC methods

Metric	Reported in
Hamming loss	[6]-[9], [13]-[21], [24]-[25], [28]-[30], [32], [35], [37], [39], [43],[46] [49], [52], [53], [56], [60], [61], [64], [66]-[68], [71], [72], [80]
Ranking loss	[4], [5], [7]-[9], [11], [14], [16], [18]-[21], [12], [25], [28], [39], [46] [52], [53], [61], [72], [80]
One error	[7]-[9], [13], [14], [16]-[21], [12], [25], [28], [46], [52], [53], [72], [80]
Coverage	[7]-[9], [13], [14], [16]-[21], [12], [25], [28], [46], [52], [53], [61], [72]
Average precision	[4], [5], [7]-[9], [11]-[14], [16]-[21], [12], [25], [28], [46], [52], [53] [60], [72], [80]
Accuracy	[6], [10], [15], [16], [18]-[21], [23], [29]-[33], [35], [37], [44]
Accuracy	[49], [56], [60], [61], [67], [68], [71]
Subset accuracy	[6], [11], [16], [18]-[21], [37], [56]
Precision	[6], [10], [11], [15], [16], [18]-[21], [31], [32], [67], [71], [80]
Recall	[6], [10], [11], [15], [16], [18]-[21], [31], [32], [67], [71], [80]
F-measure	[6], [11], [16], [18]-[21], [29]-[31], [37], [42], [43], [47], [56] [60], [64], [71], [80]
ROC	[25], [38], [40], [47], [53], [71]
Macro precision, recall	[19], [20]
Macro F1	[3], [2], [8], [10], [19], [20], [22], [23], [37], [56], [66]
Micro precision, recall	[19], [20], [80]
Micro F1	[3], [2], [8], [10], [19], [20], [22], [24], [37], [56], [63], [66]
Macro Micro AUC	[8], [20]
AUPRC	[23], [38]
Hierarchical loss	[18]
Log loss	[23]
Exact match	[66]

Performance metrics used for assessment of MLC methods by various researchers are listed in Table 2.10. Hamming loss is used as the most common metric used by many researchers.

2.7 Datasets

ML datasets from different domains are provided by MEKA, Mulan and LibSVM [73]-[76]. These datasets show varying performance depending on label statistics. It can be measured by the following parameters.

- **Label Cardinality (LC):** It denotes an average number of labels per instance.
- **Label Density (LD):** It is a ratio of LC to the total number of labels.

- **Label Diversity (LV):** It represents how many sets of different label combinations are available in the data set [16] [20].

Tsoumakas, G. and Katakis, I. [15] introduced LC and LD. Read J. [97] has presented PUNIQ that represents a ratio of LV to the number of examples. One more parameter called PMAX is also introduced by the author that represents the ratio of the count of the most frequent label set to a total number of examples. Low PUNIQ indicates regularity of labels, whereas low PMAX indicates the uniformity of labels. High PUNIQ reveals that many label sets are occurring in the dataset; hence less number of examples is associated with each label set. High PMAX shows that a large number of examples are associated with the most frequent label set, resulting in label skew. Consequently, less number of examples are associated with less frequent label sets.

Different datasets possess a different number of labels. Also, the number of labels to be associated with varies for each instance. It imposes problem while evaluating ML methods and comparing their performance. Above parameters are useful to perform the same.

Datasets with their domains are shown in Table 5.1.

2.8 Tools for implementation of MLC

Different tools provide existing ML methods that can be used by researchers and practitioners for study and to compare with their implementation. Some tools are listed in Table 2.12 [73]-[76] [80]]. MEKA [73]] provides a GUI. It is an open-source library. Mulan [74] provides libraries that can be imported in a Java program. Both tools are built on WEKA [75]. LibSVM [76] is another tool that supports libraries for traditional Support Vector Machines (SVM) which need some changes for ML support. These tools process datasets in either Comma Separated Value (CSV) or Attribute Relation File Format (ARFF). Scikit-multilearn is a library available in Python that is designed to support MLC [80].

To summarize, initially different applications where ML data is used, are listed in this chapter. Taxonomy of classification is discussed, followed by a comparison of SLC and MLC. Then taxonomy of MLC from two different perspectives is discussed, one based on

TABLE 2.11: Datasets used by MLC methods

Dataset	Domain	Reported in
BioASQ	Biology	[3]
OHSUMED	Text	[2], [16], [23], [29], [39]
ImageNet, PASCAL	Multimedia	[4]
NCI, PTC	Biology	[5]
Yeast	Biology	[6], [7], [15], [16], [18], [19], [21]-[23], [12], [28]-[30], [35], [37], [39], [40], [53], [56], [60], [61], [64], [66], [68], [71], [80]
Protein sequences	Biology	[6]
Reuters	Text	[7], [11], [16], [18], [22], [23], [25], [30], [31], [39], [42]-[44], [53], [66], [80]
Scene	Images	[9], [14]-[16], [18], [19], [21]-[23], [12], [28]-[30], [35], [37], [39]-[41], [53], [56], [60], [61], [64], [66], [68], [71], [80]
EUR-Lex	Text	[11], [16], [53]
HiFind	Multimedia	[11], [16]
Web pages	Web	[14], [16], [12], [43], [46], [52]
Genbase	Biology	[15], [18], [21], [56], [66], [68]
Medical	Text	[16], [19], [21], [22], [23], [29], [30], [39], [41], [43], [52], [56], [60], [64], [66], [68]
Mediamill	Multimedia	[16], [18], [19], [21]-[24], [39], [53], [63]
Enron	Web	[16], [19], [21]-[23], [29], [30], [39], [41], [52], [53], [56], [60], [64], [66], [68]
Emotions	Multimedia	[16], [18], [19], [21], [35], [37], [39], [56], [60], [61], [68]
FunCat, GO	Biology	[16]
Delicious	Text	[18], [19], [23], [24], [66]
tmc2007	Text	[18], [19], [22], [23], [39], [53], [71]
Corel5k	Multimedia	[19], [21], [53], [56], [60], [64], [66], [68]
Bibtex	Text	[19], [22], [23], [41], [53], [56], [66], [68]
Bookmarks	Text	[19]
Slashdot	Text	[23], [39], [53]
IMDB	Text	[23], [39]
AP Titles, UseNet data	Text	[25]
CAL500	Multimedia	[41], [53], [56], [60], [68]
Language log	Text	[53]
Image	Multimedia	[39], [53]
Corel16k	Multimedia	[53], [56], [66]
Flags	Multimedia	[68]
Birds	Multimedia	[66]

TABLE 2.12: Tools supporting MLC implementation

Tools	Reported in
MEKA	[19], [21], [73]
Mulan	[6], [18], [19], [21], [37], [49], [52], [55], [56], [60], [63], [66], [68], [74]
WEKA	[19], [23], [29], [30], [55], [56], [60], [63], [75]
LIBSVM	[10], [11], [18], [19], [41], [47], [72], [76]
scikit-multilearn	[80]

the approach and the other based on dependency. Later taxonomy of ML tasks according to output, namely classification, ranking and combination of both is discussed. Next, the state-of-the-art ML methods following transformation approach along with their pros and cons are examined in detail. The state-of-the-art ML methods following adaptation and ensemble approach are discussed in brief. Assessment of MLC differs from that of SLC. How to measure the performance of MLC is elaborated along with taxonomy. Different ML datasets reported in the literature and their domain are listed. Finally, tools supporting the implementation of MLC are listed.

Chapter 3

Literature survey

Multi-label classification (MLC) methods are broadly divided into two groups. Group-I is termed as problem transformation and Group-II as algorithm adaptation. Another way to partition MLC is whether task performed is algorithm independent or algorithm dependent. Other than these significant approaches, researchers have attempted to apply MLC along with label correlation, feature selection, genetic algorithms, use of clustering etc. In this section, these methods of MLC are explained briefly.

3.1 Variations in MLC

Discrimination is the process of giving appropriate treatment to an individual depending on the membership in a specific group. According to Shantanu Godbole and Sunita Sarawagi [31], text classification could be performed using:

- Discriminative techniques: SVM, decision tree, neural network
- Generative techniques: Naïve Bayes, Expectation-Maximization

Consider a set T of documents d. Initially, authors apply discriminative technique SVM on features of documents and call it S_0. Then they use S_0 to augment each document d in T with a supervised set of labels and call it S_1. Then kernel function in linear SVM is expressed as $K_T(d_m, d_n) = (< d_m.d_n >)/(|d_m|.|d_n|)$ and Cos similarity is expressed as $K(d_m, d_n) = f.K_T(d_m, d_n) + (1 - f).K_L(d_m, d_n)$. Here f can be tuned. In this equation, the dot product kernel between terms and label space is used. The authors have given suggestions that negative training instances or confusion matrix are useful to improve the algorithm. Rainbow is used for feature and text processing, and SVMLight is used for all SVM experiments.

Many attempts are made in MLC using association classification. J. Arunadevi et al. [32] use Apriori algorithm along with an evolutionary algorithm for MLC. They propose a MOGA system that works in three phases. It uses problem transformation. In the first phase, chromosomes are represented by a sequence of $x_iy_iz_iw_i$ genes. Here x_iy_i and z_iw_i represent i^{th} attribute and its value respectively. Fitness is checked using comprehensibility, among other parameters. Single-point crossover is applied to the chromosomes selected by proportional strategy. In each chromosome, value at a random position is replaced by any number among 0 to 9. This process, along with ant colony optimization, generates rules. Fitness of these rules is checked in phase two using two parameters, namely predictive accuracy and comprehensibility. Predictive accuracy is computed using several rules satisfying all conditions in the only antecedent and that in both antecedent and consequent. Comprehensibility is calculated using conditions and length of the rule. When learning is complete, rules are merged in the third phase to obtain MLC. J. Arunadevi et al. show the application of MLC for collected shopping preferences of women and how preferences vary according to their statuses like married, working, student and mother.

Ravi Patel et al. [33] converted all the nominal attributes to numeric. For example, let Height be an attribute that takes Short, Medium or Tall values. Then each cell in the dataset with (Height = Short) is replaced by 1, (Height = Medium) is replaced by 2 and (Height = Tall) is replaced by 3. For other attributes, numbers other than 1, 2 and 3 are used. Thus each nominal attribute is replaced by a number. Next FP-growth algorithm is used to generate association rules. As a future, it is possible to process generated rules using genetic algorithm or heuristic search methods to get better rules.

Raed Alazaidah et al. [34] transformed multi-label dataset to single-label. For each instance related to multiple labels, only one label is kept that is the least frequent label in that column. They discovered positive correlations among labels and created rules for all the instances. For example, if labels Bx and By are correlated, then created rule was “if Bx = 1 then By = 1”. Rule-based classification algorithm PART was applied to the rules formed in the previous step. The last step was the prediction.

H Haripriya et al. [35] implemented k-means accompanied by association classification. Initially, k-means was applied for clustering of attributes. Size of label space was used to decide cluster count. Then each cluster S_x was represented by label B_y that had the maximum proportion of instances with label B_y in cluster S_x to a total count of instances in the cluster S_x. Next for each cluster data rules were generated. For a test instance, the rule was constructed from each cluster.

C. Vens et al. [38] proposed three methods of classification based on the decision tree: HMC considering all the superclasses of a node using mean, SC constructing a separate tree for each category and HSC considering the conditional probability of class B with its parent. HMC and HSC were applicable for classification using DAG (Directed Acyclic Graph) while SC was not. Authors used AUPRC (Area Under Precision-Recall Curve) for evaluating prediction performance. The main contribution of authors was the use of class hierarchy that was not studied earlier.

Binary Relevance (BR) designs C independent classifiers if there are C labels in the label set. This separate consideration of labels simplifies the task at the cost of losing label relationships. To handle this issue, Read J et al. [23] proposed Classifier Chain (CC) that considers C labels independently but in a particular sequence. For ex., the sequence used for a label set L_x, L_y, L_z may be L_z, L_y and L_x. That is first L_z is predicted by considering all features. Next L_y is predicted considering all features and predicted L_z. Then L_x is predicted considering all features and predicted L_y.

Thus the relationship between labels is taken into account by each label-wise classifier. The chain of labels can be permuted in multiple ways, and that is a very crucial part in CC as it directly affects its accuracy. It also dictates the inability of parallelizing the process [20] [21] [23]. Read J. et al. has introduced performance measure log loss that uses certainty of prediction. Performance of CC is very much dependent on the chain of labels used. Depending on the size of the label space, there can be many permutations of labels. Finding the best chain is quite tricky. Jesse Read et al. [22] handled this issue resulting in an ensemble of multiple CC models, each one using a different chain of labels. The method is termed as Ensemble of Classifier Chain (ECC). It yields better accuracy than CC. Another advantage of ECC is that it never predicts an empty label set due to various chains.

Classifier chains proposed by Jesse Read et al. [23] uses the greedy algorithmic strategy. It only searches for the most probable label combination. But if all the label combinations are explored for, then definitely the best result is obtained. This approach is used by Probabilistic Classifier Chain (PCC) [39] that computes the conditional probability for every label set based on the product rule of probability. It uses Naïve Bayes (NB) to yield probabilistic output. But its complexity is high at the time of prediction. Authors have used risk minimization model to minimize rank loss, subset 0/1 loss and hamming loss. Ensemble methods ECC [23] and Ensemble of PCC (EPCC) [39] are also used for experimentation. It is observed that the probabilistic versions PCC and EPCC are well-suited/appropriate for all the three measures listed here. Also, EPCC performs the best among all the competing algorithms used by authors getting the benefit of ensembles.

Hypergraph, a generalization of a simple graph, consists of hyperedges. Some researchers have used hypergraph for MLC. Spectral learning feature of hypergraph was used by Liang Sun et al. [40] to investigate the correlation of labels. It was found very helpful for high order relations. Hung-Yi Lo et al. [41] also used hypergraph to capture the relationship between multiple labels and the instances jointly.

Jung-Yi Jiang et al. [42], S. Lee [43] and Rubiya P U et al. [44] all worked on a similar kind of concept. They computed the membership degree called the degree of relevance. Three things namely the membership degree of each term tx in each category By, that of each term tx in each document dz and that of each document dz in each category By were obtained and combined to get final membership degree. All methods [42] [43] [44] performed clustering that helped to reduce features as well as the computational cost of kNN.

3.2 Fuzzy MLC

In MLC, an instance x_i is associated with a set $Z = \{z_1, z_2 \cdots\}$ of labels. Label z_i is set to 1 if z_i is associated with x_i. Otherwise, z_i is 0. Thus set Z is a crisp set. For example, if a committee of experts is appointed to classify particular data, then every expert may have different opinions to classify a particular instance to any class. Some experts may assign a class to the instance completely ($z_i = 1$), some experts may not assign a class to the instance ($z_i = 0$), and some experts may not be sure to assign a class to the instance completely or not at all. This last scenario can be described by assigning a value in between 0 and 1 to z_i (i.e. $0 < z_i < 1$). It is called *fuzzy membership* [81] which reflects the practical scenario better. It is the basic idea behind fuzzy MLC.

Definition: Let Z be a set of disjoint labels. For a set of training documents having (x_j, z_j) pairs, obtain a function $f_z(x)$ to map each instance x_j to a set z_j, for $j = 1 \cdots |E|$, where E denotes a set of training examples and $z_j = \{v|0 \leq v \leq 1\}^{|Z|}$. Here value v represents the degree of membership in each class Z_i. A value near to 1 represents more membership and near to 0 represents less membership.

There are few attempts to use fuzzy set theory for MLC. Z. Younes et al. [82] propose ML classification using FV-kNN algorithm using an adaptation of k-nearest neighbor with the help of fuzzy sets and veristic variables. It uses the context of the veristic variables. Veristic variables can take more than one values, hence are similar to multivalued variables. The traditional kNN algorithm considers all neighbors at the same level. FV-kNN views each neighbor according to its distance. Less distance means more weightage to that neighbor. Then for each instance in training data, its membership in each class is computed. Knowledge obtained from the computation of class membership for instances is represented using veristic statements. Then the knowledge of all the veristic statements is combined, and the set of labels is predicted. Experiments are performed on three datasets, namely emotion, scene and yeast datasets.

Jiang et al. [42] proposed FSKNN that is a Fuzzy Similarity-based approach using kNN. It performs text classification. In Multi-Label k Nearest Neighbor, kNN is modified for handling ML data. But high computation cost for finding neighbors is the main overhead in ML-kNN. To cope up with this issue in FSKNN, first, the clusters of similar documents are formed using the technique of fuzzy similarity measure (FSM) that helps to minimize the search space of neighbors. When the similarity of cluster data and new data computed by FSM is higher than some threshold, then label set for new data is obtained using prior and likelihood information based on MAP rule whose base is the Bayes theorem. Performance of FSKNN algorithm is compared with that of three algorithms. The experimentation is done using the datasets Reuters-21578, RCV1, and 20 Newsgroups for evaluation using micro-averaged F1 and breakeven point (BEP). Experiments indicate that FSKNN outperforms as compared to three competing methods showing improved execution time and precision.

Lee et al. [43] propose ML-FRC algorithm for multi-label data following algorithm adaptation approach. It deals with the overhead of high dimensionality. ML-FRC first represents documents having high dimensionality using vectors having low dimensionality. It involves the conversion of documents having f features into fuzzy relevance vectors of size $|Z|$ where Z is a set of labels. It is achieved using fuzzy relevance measure (FRM). This dimensionality reduction is beneficial to decrease classification time and improve the performance of a classifier. Incrementally these vectors are then added into clusters having similar vectors. Next, the relation between the obtained clusters and classes is searched. Label-wise thresholds are used and the output label set is obtained. Same steps are done during training and testing. The experiments are conducted for comparison of ML-FRC algorithm with ML-KNN, Rank-SVM, BoosTexter and ML-RBF using four datasets, namely Medical, WebKB, RCV1 and YAHOO web pages. Micro-averaged F1 and BEP and hamming loss are used to measure the performance of the classifier. The authors explore the classifier for finding documents which do not belong to any predefined class.

Chen et al. [83] use FHML algorithm for fuzzy hypergraph regularization. It is used for prediction of the subcellular location of multi-location proteins. FHML uses three phases. In the feature layer, the protein database is used where each protein is represented by a vector and then decomposed into latent concepts. A feature graph is also constructed. In the label layer, the label space is decomposed into latent concepts. A graph of labels is also constructed. Fuzzy hypergraphs are used to explore the relation between (i) latent codes and features, and (ii) latent codes and labels. Thus latent layer works as a middle layer in between the label layer and the feature layer. The multi-label learning is used to propagate information among proteins from labelled one to unlabeled one. The experiments are performed on datasets from the Cell-Ploc 2.0 package for multi-location protein, namely, human, eukaryote, plant, gneg, gpos and virus. The authors use metrics such as accuracy and F-Measure that are example-based, whereas precision and recall that are label-based metrics. The authors have reported the benefit of using correlations among features of instances and associations among the classes together.

R. C. Prati [84] has used problem transformation approach for MLL. The author has selected a fuzzy rule-based learning classifier to work as a base classifier. Rule-based classifier algorithms follow either separate-and-conquer family or divide-and-conquer family. Conventional rules associate an instance to particular category completely (represented by 1) or not at all (represented by 0). But this hard decision is not suitable sometimes in few applications. Here fuzzy rules can work better, providing soft decision and gradual changes in the class memberships. Here the author has used FURIA algorithm [85] as a base classifier in the context of multi-label learning problems. FURIA adapts Rule Induction Algorithm along with fuzzy sets. Experimentation is done using four problem transformation methods, in a combination of the eight base-learners, for each of the six datasets, having 32 combinations of the multi-label problem transformation methods and base-classifier. Finally, for each combination, the five different performance measures are calculated using 5-fold cross-validation.

To summarize, various attempts to use fuzzy sets along with MLC is presented in this section. As it is not always possible in some cases to assign each instance to a particular category entirely or not at all, the theory of fuzzy sets can be incorporated. As there are two main techniques of MLC, some researchers use problem transformation with fuzzy sets, and some researchers use algorithm adaptation with fuzzy sets. Researchers have reported an increase in classifier performance by using fuzzy sets.

3.3 Clustering

Clustering is the most popular form of unsupervised data analysis [62]. Many researchers have utilized clustering to reduce computational cost of MLC [24] [35] [40] [41] [42] [43] [44] [50] [53] [59]. Some of them are already discussed briefly in other sections. This section describes a few more attempts for the same.

Nasierding et al. [63] have designed CBMLC algorithm. It works in two phases. In the first phase, k clusters are formed from training instances where the value of k is specified by the user. Labels are not considered during clustering. Next in the second phase, k multi-label classification models are constructed for k clusters independently. For a test instance, its closest cluster is searched, and a model of that cluster is used for classification. Clustering helps to minimize computation time required to train and classify. Experimentation on three datasets is performed to measure micro F1. Two to ten clusters are formed using k-means and expectation maximization. Four state-of-the-art algorithms are used to evaluate performance after clustering is applied. According to Nasierding et al., CBMLC is the first attempt to apply clustering analysis on the dataset before feeding the data to a classifier.

Pranav Gupta and Ashish Anand [64] presented "Multi-Label Classification using Label Clustering" in the 1st Indian Workshop held at IIT Kanpur in 2013 on Machine Learning. The basic idea of replacing less frequent label sets by frequent label sets is taken from Pruned Set (PS). Authors apply k-means clustering on the dataset. After forming clusters of labels, new trained data is constructed such that only those instances which belong to label set C_x are considered for training C_x. Accordingly, the trained data in clusters is modified. Next PS classifier is trained with modified trained data. Performance of three measures on three ML datasets is presented.

Zhilou Yu et al. [65] have proposed a method based on Classifier Chains (CC). In CC, Binary Relevance (BR) is applied one by one for each label. The point where CC differs BR is that CC uses BR in a particular sequence of labels and CC_{x+1} takes input from all features and prediction for label x. The crucial decision for CC is a sequence of labels to be considered. It directly affects predictive accuracy. Zhilou Yu et al. handled this matter by acquiring associations between labels. These associations helped to establish the sequence of labels to be used. Authors employed k-means algorithm repeatedly to extract correlations between labels. It is important to note that in this method, clustering of labels was done, not instances. 5-fold cross-validation was implemented using six regular and twelve large-scale datasets. Clustering helped to reduce the size of datasets to the large extent that revealed in faster execution time.

G.A. Kaminka et al. [66] applied dimension reduction using orthonormalized Partial Least Squares to find the direction of maximum covariance between label space and feature space using SVD. The system produced clusters using k-means and learned meta-labels using Laplacian Eigen map within each cluster. At the end system constructed classifier chains over meta-labels for local model learning.

One challenge in MLC is the scalability of an algorithm concerning dimensions of the label space. Because of more labels, the algorithm has to suffer from the class imbalance problem, computational cost of training and the inefficiency for applications requiring fast response times. Grigorios Tsoumakas et al. [23] designed the algorithm Hierarchy of multi-label classifiers (HOMER) for handling more labels. The first root node is constructed that consists of all the labels. Next clustering with balanced k-means is employed to divide labels into clusters which represent new nodes. Then design a classifier for each cluster to handle labels in that cluster only. If the predicted label is in meta-labels of the child node, then only call classifier of that child node. Advantage of balanced clustering is that the related labels belong to the same cluster, hence the same node of the tree. So the only classifier of that node needs to be invoked, thereby reducing the cost of prediction. Also, each node handles less training instances, thereby improving predictive performance. Note that clustering of labels is done by G. Tsoumakas et al. by partitioning labels into clusters and the tree structure is used for representation [18] [21].

3.4 Natural algorithms

Inspiring from how various things work in nature, evolutionary algorithms are evolved. Neural network in machine learning is inspired by the working of the neuron in our brain. The life of ants inspired ant Colony algorithm used in artificial intelligence. Attempts are made to improve MLC using such natural algorithms. Some of them are listed here.

M. L. Zhang et al. [28] and S. Jungjit et al. [52] have used a genetic algorithm (GA) whereas the later have also used Hill climbing. Ravi Patel et al. [33] have used association classification and evolutionary algorithms, as mentioned in earlier section.

Rosane M. M. Vallim et al. [67] proposed MLOCS in which a genetic algorithm is used to improve association rules. Initially, the problem transformation is done, followed by the application of single label rule mining using association rules. Next, a genetic algorithm is applied to obtain better rules by performing bit change either on the left side of the rule or on the right side of the rule.

As mentioned in section 3.1.3, J. Read et al. [23] has stated that the sequence of labels is essential to get the desired accuracy in the classifier chain (CC). Eduardo Corrêa Gonçalves et al. [68] use CC as a base classifier, and GA is used to find the order in which labels are used in the chain of a classifier.

S. Jungjit et al. [69] use Pearson's correlation coefficient to measure dependency between feature and feature as well as feature and label, and also the mutual information to find the correlation between two labels. An algorithm is implemented using Hill climbing, and a genetic algorithm is applied to characteristics for selection.

3.5 Feature selection and dimensionality reduction in MLC

Many applications in real-life use data with complex structures. Some examples are XML web document, chemical compounds, program flow, etc. Such data cannot be represented with feature vectors accurately. In that case, the graph proves to be a better solution [5]. When vectors are used to describe features, then the feature selection process is somewhat more straightforward because it is assumed that all the features are available initially. It is not possible for graphs because as the size of the graph increases, complexity increases too much. Authors have mentioned the use of label associations for graph classification with feature selection as future scope.

Trohidis, K. and Tsoumakas, G. et al. [8] follow the transformation approach. The general procedure for feature selection by many researchers is as follows: Convert data from multi-label to single-label. Then apply traditional single-label feature selection technique like chi-square and use a max or average method to select best features. In a max process, N number of features are chosen which have maximum chi-square values. In an average technique, the average of all the values for each feature is obtained within all the labels weighted by the prior probability of every label. Then N number of features are selected having maximum values. BR can be applied to these selected features only. The problem with this method is that it considers each label independently. This issue is handled by authors using LP instead of BR. The benefit is that LP implicitly uses label correlations, thereby giving better results when used with chi-square for feature selection. Authors have extracted features of two categories, namely rhythmic and timbre, from music using the Marsyas tool followed by emotion labelling and annotation by music experts.

A. Clare and R. D. King [26] has introduced a feature selection technique ML-IG to handle multiple label data as given in section 3.2. Gao, Sheng et al. [45] have used Singular Value Decomposition (SVD) based Latent Semantic Indexing (LSI) for feature selection. Initially, term-document matrix M is decomposed into a multiplication of three matrices as M=USVT where U, S and V are left singular matrix, a diagonal matrix of singular values and right singular matrix respectively. Also, U and V are column orthonormal. U, S and V matrices are much smaller than M. The advantage is that it dramatically reduces computation requirements.

There are two ways for dimensionality reduction, namely unsupervised and supervised. For example, first can be achieved using Principle Component Analysis and later can be made using Linear Discriminant Analysis. Y. Zhang et al. [46] used a basic idea which tries to identify a feature space of small size to maximize dependency between labels and features. It uses the Hilbert-Schmidt Independence Criterion (HSIC) for measurement of dependence. Initially, the algorithm prepares label kernel matrix L from label space Y. Next eigenvectors are conformed to largest m eigenvalues to get projection P from original features to the reduced features. Authors suggested a variation to use HSIC with gradient descent.

Ji S. et al. [47] used the least-squares loss for the classification to compute the shared structure and solved a generalized eigenvalue problem. M. L. Zhang et al. [28] have implemented a feature selection with multi-label Naïve Bayes (MLNB) algorithm. First, use multi-label dataset Do to apply PCA for feature extraction followed by genetic algorithm for feature selection. If f, C and h(.) denote feature, label and classifier respectively, then $h_f(C) = 1$ if f is selected otherwise $h_f(C) = 0$ if not selected. Form new dataset D_n from selected features. Divide D_n into ten parts and use tenfold cross-validation for evaluation. The author has used the fitness function based upon the average of hamming loss and ranking loss generated by a portion of dataset D_n used in all the ten folds. Next step is to apply MLNB that makes use of prior and posterior probabilities.

G. Doquire and M. Verleysen [49] used Pruned Problem Transformation (PPT) along with mutual information. PPT overcomes the problem in LP that some label sets possess very few instances among others affecting accuracy. In PPT, all instances having label sets that occur in the dataset number of times less than a predefined threshold, are removed. Such examples are replaced by examples having disjoint subsets of that label set. Again it is checked whether newly added instances with disjoint label sets occur several times greater than a threshold, then they are considered otherwise discarded. Assume that occurrence count of label set $\{L_1, L_3, L_4\}$ is less than a threshold. Hence all such examples are replaced by examples having subsets $\{L_1\}$ and $\{L_3, L_4\}$ respectively. Let the occurrence count of $\{L_1\}$ is still less than a threshold. All such instances are discarded. Occurrence count of $\{L_3, L_4\}$ is more than a threshold. Hence all such instances are considered. After using PPT [18] [29] for data conversion, Doquire et al. apply mutual information (MI) for feature selection. The feature selection process follows a greedy approach as it starts with zero features followed by appending the set with feature showing the highest MI with label set. MI measures how much information two features contain about each other. It is important to note that number of neighbors used for MI estimation should be less than the threshold used for pruning in PPT. The method follows a transformation approach only while selecting features. For MLC, all samples from the data are considered. Performance is evaluated in terms of hamming loss and accuracy using three ML datasets.

Li S. et al. [50] used information gain for an ensemble of multi-label feature selection. Initially, the dataset is partitioned into clusters using k-means. Label cardinality introduced in [15] is used to set a count of clusters. Then information gain of every feature x_k for each label C_k is computed and normalized. The normalized value 0 and 1 indicate that particular feature and label are independent or dependent, respectively. Next using normalized values of each feature for all labels, IGS value is calculated, and the procedure is repeated for all the features using all instances in each cluster separately. Aggregate IGS value of each feature is computed as the summation of aggregate IGS value of that feature among all the clusters. Summation of aggregate values of all the features S is used to decide stopping criterion. All features are sorted in descending order of aggregate IGS values. These features are selected one by one until the addition of their aggregate IGS value is less than the threshold set, and only these features are considered. $S \times \delta$ is used to set a threshold. δ belongs to [0, 1]. Authors repeated experiments with δ changed from initial value 0.05, step 0.05 and final value 0.95 and found that δ equal to 0.35 and 0.9 give good results in text and biological domain respectively.

Li L. et al. [51] used the information gain to measure the degree of association between feature f_x and label C_y. A larger value represents better association. It calculates information gain IGS of each feature for the whole label set. These values are normalized, and their average is used to decide threshold μ. Every feature with IGS value less than μ is removed from the list.

Jungjit and Freitas [52] have used Pearson's correlation coefficient and genetic algorithm for implementation. They represented each instance by n bits string. Bit f_x = 1 or 0 denotes whether feature f_x is selected or not respectively. Fitness function is based upon Pearson's linear correlation coefficient. Individuals at each generation are chosen by combining tournament selection operator with elitism generator. Next crossover and mutation are carried out. Feature selection by Hill Climbing (HC) is used for comparison of the results. It should be noted that genetic algorithm selects more features as input features increases. HC has shown better performance in this case.

Zhang, M.L. and Wu, L. [53] have not induced classifier from the original features. They constructed label specific features using k-means clustering. They are for producing a classification model. That is, m features are represented using 2k clusters, k positive and k negative. Thus m-dimensional feature space is reduced to 2k dimensional feature space where m¿¿k (m is much larger than k) in the LIFT algorithm proposed by authors. They designed two variants of the algorithm, one using information gain of all the features and the other using relation between labels and instances.

K. Kira and L. A. Rendell 54 have proposed a feature selection method that is based on a statistical approach instead of a heuristic approach. Relief is one of the feature selection method used for single-label learning. It rewards if two attributes have different feature value for two classes and apply a penalty if two attributes have a different value for the same class. Newton Spolaˆor et al. [55] proposed an algorithm based on Relief to select features in multi-label datasets. The algorithm searches for k neighbors and also uses dissimilarity of instances to find the importance of features.

Newton Spolaˆor et al. 56 determined the contribution of each feature for each label. An average score of each feature within all the labels is computed. Features having average score more than a threshold are chosen.

Lazy approaches are proved beneficial while evaluating methods of feature selection. The reason is that classifiers based on lazy strategies are generally vulnerable to irrelevant features. Three procedures of feature selection are practised by most of the researchers. They are (i) filter - not dependent of the learning algorithm, (ii) wrapper - used along with the learning algorithm and (iii) embedded - in which feature selection is the part of the training process. Measures used to know the importance of features are information gain (IG) [79], Relief, chi-square, Gini index, rough set, etc. When a dataset has three labels L_x, L_y and L_z, then data with all the features and one label is constructed. For each feature x_k, $IG(x_k)$ w.r.t each label is computed separately. Feature x_k having an average of all three values w.r.t. L_x, L_y and L_z, above a threshold, is considered by the algorithm. The used threshold value is 0.01. Spider graph is used for visualization of performance, and comparison is made using the R framework.

3.6 Label correlation and dependency-based MLC algorithms

Label cardinality and label density were introduced by Tsoumakas G. et al. [15]. These two characteristics denote that datasets having equal label cardinality and unequal label density can possess varying characteristics and behave differently for MLC methods. Former denotes the average count of labels per example, whereas the latter indicates a ratio of label cardinality to the size of label space.

M. L. Zhang et al. [7] as described in section 2.3.2, J. Arunadevi et al. [32] and Liang Sun et al. [40] as described in section 3.1, have used label correlation.

M. L. Zhang et al. [57] have encoded conditional dependencies of labels and feature set using a Bayesian network structure. They treated the whole feature set as the common parent of all the labels. Bayesian network characterized the joint probability of all labels on the feature set with the help of DAG. Then a binary classifier was developed for each label with the help of parent labels in DAG as added features.

Z. H. Zhou et al. [58] have explored the relationship between labels as asymmetric. If labels B_y and B_z are relevant, then hypothesis generated for label B_y may help for the other label B_z. If $R_s(m, n)$ is the reuse score from label n to m, then $R_s(m, n)$ is not necessarily the same as $R_s(n, m)$. Authors employed a boosting approach with hypothesis reuse. The system produced an estimate of the label relationship as output. Authors investigated three kinds of possible relationships among labels, namely reuse score, co-occurrence relationship and Φ-coefficient relationship.

The basic idea behind [59] is that label relations may be shared by only a subset of instances rather than all the instances. Exploiting such global relationships may be misleading and may hurt the classifier performance by predicting some irrelevant labels. The approach used is to separate training data into m groups $\{G_1 \dots G_m\}$ where instances in the same group G_x share same label correlations. These groups are created using k-means clustering by finding the similarity in label vectors, instead of feature vectors. Each group G_x represents label correlations R_x. Each G_x is represented by a prototype vector P_x. For m groups, there are m prototype vectors $\{P_1 \dots P_m\}$. Find the similarity of each instance x_k with these prototype vectors P_k to get LOC code vector $L_k = \{L_{k1} \dots L_{km}\}$ where L_{ko} is the local influence of R_o on instance x_k. Then train m regression models with the original features as input and LOC codes as outputs. For an unseen instance x_u, first obtain LOC code $L_u = L_{u1} \dots L_{um}$ using m regression models. Then get the final label vector C_u using x_u and L_u. As a future scope, authors mentioned the use of different clustering algorithm and different loss function.

Ying Yu [60] has proposed two techniques MLRS and MLRS-LC in 2014. In both methods, the rough set model based on equivalence relation and equivalence classes is used. Samples are said to be equivalent if their attribute values are identical to each other. It computes neighbors of each instance X_n for each label C_m. More the neighbors with label C_m, higher is the probability of X_n related to C_m. This information is computed globally for MLRS and locally for MLRS-LC, respectively. Global computation involves all the instances in the dataset, and local calculation includes a small subset of instances, thereby resulting in better results compared to a global one. The author has suggested high dimensionality reduction as a future direction.

Chi-square is univariate and scores each feature individually. They are hence used with problem transformation generally like BR and LP. Mutual information [61] is multivariate and useful to find a joint score of relevant features. Therefore mutual information is suitable for multi-label classification.

3.7 kNN-based methods

kNN has always remained the first choice of many researchers because of its simplicity. Many researchers have been inspired to adapt kNN to design MLC. Min-Ling Zhang and Zhi-Hua Zhou [12] [89] have proposed ML-kNN method. It has proven to be the state-of-the-art method. Authors experimented using only one dataset earlier in 2005 [89]. Later three datasets were used for experimentation in 2007 [12]. It is designed by adapting conventional kNN such that it will suit for multi-label data. The basic idea behind the work is as follows. Let k neighbors for an instance X are computed. Then neighbors of an instance X belonging to each label C_m of X is counted. Also, neighbors of an instance X belonging to each label C_m that is not belonging to X is counted. Next likelihood probability is computed using these two counts. Prior probabilities are also obtained from the training set by counting instances having label C_m and not having label C_m respectively. Next labels of a new instance are obtained using Maximum a posteriori that is based on Bayes theorem [10]. The posterior probability for each label C_m is then computed for an unseen instance. ML-kNN [13] [18] [19] [20] [21] [22] [12] has proven to be the state-of-the-art algorithm. This algorithm has inspired many researchers though it has one limitation of not considering label relationship.

E. Spyromitros et al. [37] have proposed an algorithm BRkNN using a lazy approach in 2008. Instead of searching kNN separately for each label, BRkNN performs a single search of kNN, followed by independent predictions made for each label. Initially, kNN is applied on the multi-label data to obtain k neighbors. Once neighbors are obtained, then BR classifier uses these neighbors independently for prediction of each label. Two variations of BRkNN are also implemented by the authors. In case there is no relevant label predicted, the first variation returns the most probable label and the second variation returns p most probable labels where p is equal to an average of a count of labels belonging to k neighbors. These methods never output empty sets. Authors have compared their methods with LPkNN and MLkNN [12]. Authors do not safely argue that high label density datasets lead to improved performance of the LPkNN algorithm.

Z. Younes et al. generalized ML-kNN and considered dependencies between class labels [103]. DMLkNN does not use a particular label but considers different labels in the neighborhood. If membership of instance x is the same as membership of instance x_i in neighborhood for all the labels, then x_i is used further for extracting required information.

ML-kNN that follows an algorithm adaptation approach does not consider label correlation and thus results in lesser prediction accuracy. A new method called CML-kNN proposed by Chunming Liu and Longbing Cao [91], exploits label correlation using both intra-coupling and inter-coupling label similarities between the labels to provide better accuracy than that of ML-kNN. They consider labels in pair as L_1 and L_2 and compute CLS (Coupling Label Similarity). Let L_1 and L_2 take values v_1 and v_2, respectively. How many instances take values (v_1, v_2) for (L_1, L_2) in the dataset is counted for computing intra-CLS. Then inter-CLS is calculated for (v_1, v_2) values of (L_1, L_2) along with each feature value v_f. These intra and inter CLS values are used for likelihood estimation.

Veloso et al. [94] proposed MLAC that is lazy. It performs the training process for test-instance only when it arrives. It introduces multi-label class association rules as a way to model label correlations and dependencies among labels.

Some researchers find k nearest neighbors and use their information further in the MLC algorithm. The correlation between labels can be considered in these algorithms to improve performance [96].

There are few attempts to use fuzzy set theory for MLC. Young et al. [82] have proposed ML classification using FV-kNN algorithm using an adaptation of k-nearest neighbor with the help of fuzzy sets and veristic variables in 2010. It uses the context of the veristic variables. Veristic variables can take more than one values, hence are similar to multivalued variables. FV-kNN considers each neighbor according to its distance. Less distance means more weightage to that neighbor. Then for each instance in training data, its membership in each class is computed. Knowledge obtained from the computation of class membership for instances is represented using veristic statements. Then the knowledge of all the veristic statements is combined, and the set of labels is predicted. Experiments are conducted on three datasets, namely emotion, scene and yeast datasets.

Jiang et al. [42] proposed FSKNN that is a Fuzzy Similarity-based approach using kNN in 2012. It performs text classification. In Multi-Label k Nearest Neighbor, kNN is modified for handling ML data. But high computation cost for finding neighbors is the main overhead in ML-kNN. Authors handle this issue using clusters. Clusters of similar documents are formed using the technique of fuzzy similarity measure (FSM) that helps to minimize the search space of neighbors. The similarity of cluster data and new data is computed by FSM. If it is higher than some threshold, then the result label set for that new data is obtained using the prior information and likelihood information based on MAP rule whose base is the Bayes theorem. Performance of FSKNN algorithm is compared with that of three algorithms. Experiments are conducted using datasets Reuters-21578, RCV1, and 20 Newsgroups for evaluation using micro-averaged F1 and breakeven point (BEP). Experimentation indicates that FSKNN outperforms as compared to three competing methods showing improved execution time and precision.

3.8 Motivation

Classification methods that handle multi-label data follow one of the two approaches. Those methods that transform data provide simplicity at the cost of loss of information. This drawback is overcome by an algorithm adaptation approach that is found to be superior when compared with the problem transformation approach.

MLkNN appears currently to be the best algorithm. This state of the art method adapts kNN (k nearest neighbors) to find neighbors that are followed by extraction of information. This information is useful for further computation. However, the algorithm has a drawback of considering each label separately and thereby not considering label relationships.

The selection of the most appropriate neighbours is crucial for any kNN-based algorithm. Most of the work use only features to measure the similarity between instances. Computation of feature similarity has been commonly used in existing approaches, including MLkNN. It does not use labels for the selection of neighbors. However, in the case of multi-label classification, an instance is associated with more than one label. Hence it would be better to consider labels also in addition to features for the selection of neighbors to improve classifier performance further. Labels can be utilized to measure dissimilarity as instances having common labels generally indicate identical label correlations. Thus in the multi-label context, a new approach may be devised for further performance enhancements by considering label dissimilarity in addition to feature similarity.

The use of the Euclidean distance metric for computation of feature similarity is very common in existing approaches, including MLkNN. Investigation of the performance of the devised algorithm with the use of other distance metrics such as Manhattan and Minkowski for feature similarity and label dissimilarity is also needed w.r.t. various performance parameters and computation time.

Study and performance analysis of devised algorithm with variation in input parameters such as 'k' (number of neighbours), threshold, and smoothing parameter is required to be carried out.

Feature and instance selection are often used in the literature to reduce computation time while improving the performance of the classifier. In the case of multi-label classification, labels may be related to different features. So feature selection becomes tricky. Thus, the performance of the devised algorithm with and without feature and instance selection needs to be observed.

Some datasets consist of a large number of examples as compared to the number of features and labels. It would be interesting to test the capability of the devised algorithm in the identification of appropriate neighbours for multi-label data.

Usually, outliers affect the performance of a conventional classifier. As per the literature studied, it seems that it has not been investigated for multi-label classification and thus should be investigated.

Recognizing the need for handling some of the issues mentioned above and dealing with them motivated us to carry out the research work undertaken to overcome some of these issues associated with kNN-based multi-label classification methods.

Chapter 4

Methodology and Proposed Algorithms

As discussed in previous chapters, multi-label classification is applicable in different day to day applications. Hence it has become a key concept in the field of classification and machine learning. A review of prominent research work in the literature is presented in chapter 3. This chapter deals with the details of proposed algorithms for multi-label classification. In this chapter, two novel algorithms for multi-label classification, namely MLFLD and MLFLD-MAXP, are described in detail. How multi-label data can be preprocessed before feeding it to proposed algorithms is also discussed using three different algorithms, namely MLFS, MLIS and MLFSIS, respectively.

4.1 Methodology

A general framework of a multi-label classifier is as shown in Figure 4.1. Various forms of preprocessing such as normalization, feature and/or instance selection, treatment for missing values etc. are performed on input instances that many times help to improve the performance of a classifier. A classifier is then trained using labelled input instances by either following the "problem transformation" or "algorithm adaptation" approach. Once the classifier is trained, it is used to predict the label(s) for unseen instances.

Though the process looks straightforward, some requirements should be considered for multi-label classification.

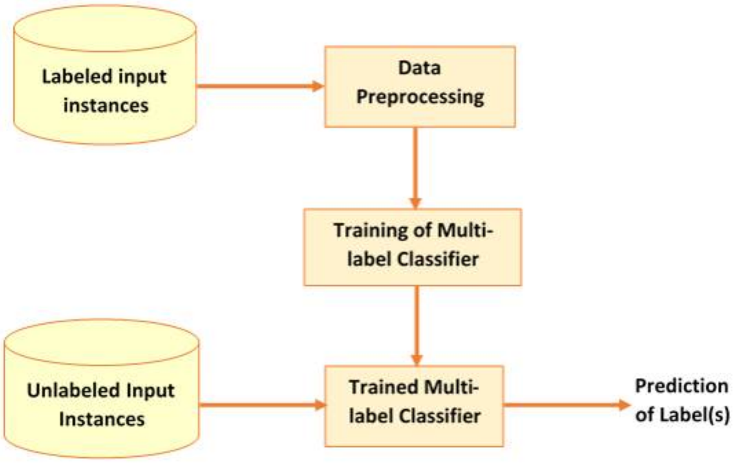


FIGURE 4.1: General Framework for Multi-label Classification

- Need to decide which forms of data preprocessing should be applied. It is observed that preprocessed data has always improved classifier performance.
- Whether problem transformation or algorithm adaptation should be used, each one has pros and cons. The first approach is simple but may lose label correlation leading to performance degradation. The later approach considers label correlation and thus provides better performance, but is complex to implement.
- Whether "single label", "pair of labels", "subset of labels" or "all the labels" should be considered at a time. Single label technique loses label correlation completely. Pair of labels technique considers relation between two labels involved in the pair only but performs better than a single label technique. A subset of labels technique operates on a subset, hence works better than both single label and pair of labels methods, but it is more complicated when compared with the other two methods. Considering all the labels increases complexity compared to others, but may perform better comparatively. In this work, "all the labels" technique is used.
- When classification or ranking of labels should be performed, generally, classification is carried out based on votes and ranking is carried out based on probabilities and can be used for classification as well. In the case of ranking, the decision of threshold is very crucial as it directly affects the predictive performance of an algorithm.

A general methodology for k-Nearest Neighbors (kNN) and Maximum a Posteriori (MAP) based multi-label algorithms is shown in Figure 4.2. MLDB in Figure 4.2 stands for a multi-label dataset.

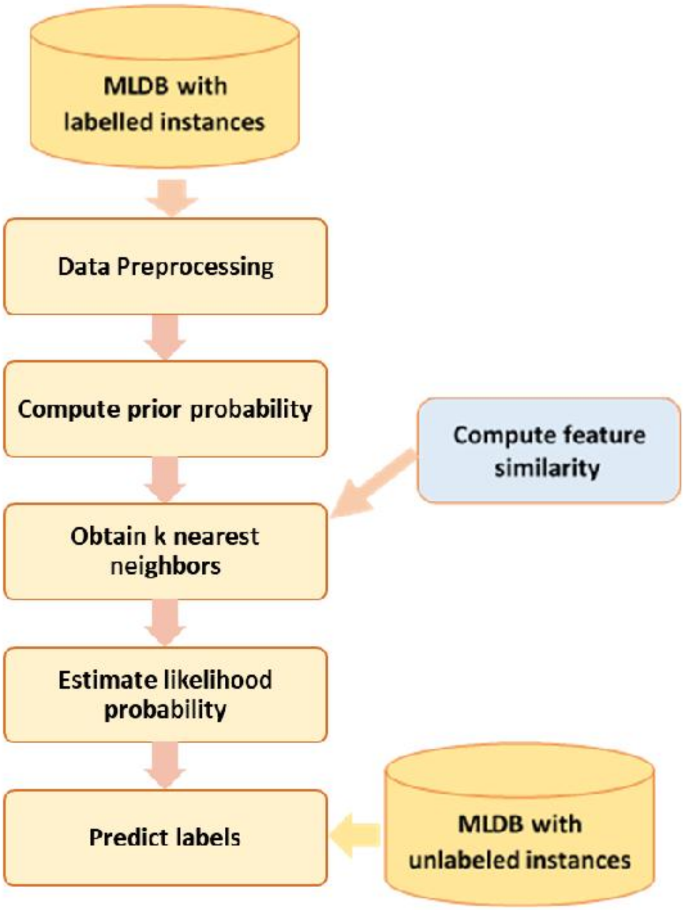


FIGURE 4.2: General Framework for kNN and MAP based Multi-label algorithms

4.2 Proposed algorithms

As proposed algorithms are based on two key concepts, namely feature similarity and label dissimilarity, some requirements need to be considered.

- A mechanism to compute feature similarity
- A mechanism to compute label dissimilarity

4.2.1 Proposed algorithm MLFLD

Proposed algorithm for Multi-Label classification by exploring Feature Similarities and Label Dissimilarities (MLFLD) aims to improve the performance of the multi-label classifier through proper selection of neighbors. It uses labels of known instances along with their features while searching for the neighbors. Then information extracted from

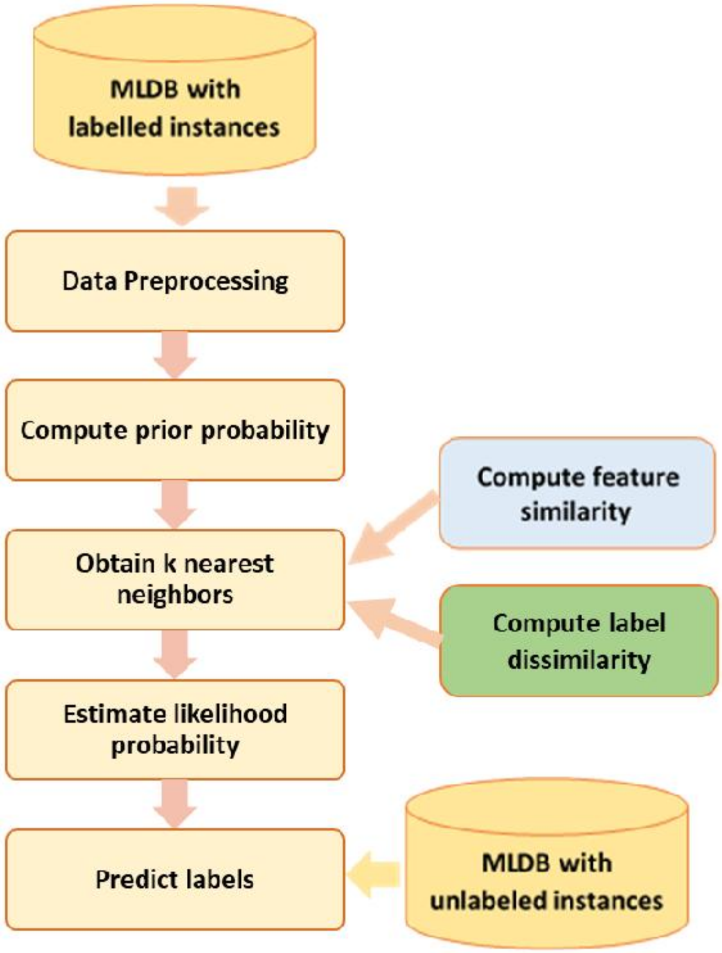


FIGURE 4.3: Framework for MLFLD

obtained neighbors is utilized for the estimation of likelihood probabilities of each label. These probabilities, along with computed prior probabilities of the particular label, are further used to predict that label for an unlabeled instance. Framework for MLFLD is shown in Figure 4.3.

The algorithm takes the following input parameters.

1. **MLDB:** Dataset having q labelled instances $\{X_1 \ldots X_q\}$. Let each instance X_j be represented by a pair of vectors, (x_j, y_c), where vector x_j, $(j = 1, 2 \ldots f)$ be the set of features and vector y_c $(c = 1, 2 \ldots l)$ be a set of labels. Knowledge obtained from these instances is utilized to select neighbors.

2. **The number of neighbors (k):** It decides how many nearest neighbors of each instance are to be considered by the algorithm.

3. **Threshold (Th):** It is a user-defined value between 0 and 1. It is used to decide whether a particular label should be associated with the underlying instance or not denoted by 1 and 0 respectively. In Eq. (6), when the ratio is greater than or equal to a threshold, then the corresponding label is set to 1. Otherwise, it is set to zero. The threshold can be user-defined or calibrated. Default value used for experimentation in this work is 0.5 as suggested in the literature [20] [12] [37] [42] [89].

4. **Smoothing parameter (p):** It is used in Eq. 4.1, Eq. 4.3 and Eq. 4.4. It is generally used to avoid resulting zero value of an operation. Default value used for experimentation in this work is 1 that denotes Laplace smoothing
citeR19 [12] [37] [42] [89].

5. **Fdistance:** Parameter that denotes distance metric used to compute feature similarity (Default metric: Euclidean distance).

6. **Ldistance:** Parameter that denotes distance metric used to compute label dissimilarity (Default metric: Hamming distance).

Output: Prediction of labels for unseen instance t

Pseudocode for MLFLD is given in Algorithm 1. It takes MLDB, k, Th, p, Fdistance and Ldistance as input.

MLFLD consists of two stages. Stage one is divided into three sub-stages:

1. *Computation of prior probability distribution* (Lines 2-4): Initially instances in MLDB associated with label c is counted. This count $cnt^{(c)}$, p (smoothing parameter) and q (size of MLDB) are used to compute prior probabilities of every label c using Eq.4.1 and Eq.4.2. For each label c, two probabilities are calculated:

   (a) Probability $P(H_c = 1)$ of the event that "an instance belongs to label c".

$$P(H_c = 1) = (p + cnt^{(c)})/(2 \times p + q) \tag{4.1}$$

   (b) Probability $P(H_c = 0)$ of the event that "an instance does not belong to label c".

$$P(H_c = 0) = 1 - P(H_c = 1) \tag{4.2}$$

Algorithm 1: MLFLD

Input : MLDB, k, Th, p, Fdistance, Ldistance

Output: Prediction of labels for unseen instance t

begin
foreach *label c in each instance* $\in$ *MLDB* **do**
  Compute $Prior_c$**:** $P(H_c = 1)$ **and** $P(H_c = 0)$ **using Eq.4.1 and Eq.4.2**
end
foreach *instance* $X_i \in MLDB$ $(1 \leq i \leq q)$ **do**
  $N_i = \phi$ // **Neighbors of** X_i
  foreach *instance* $X_j \in MLDB$ $(1 \leq j \leq q), i \neq j$ **do**
    // **fs() and ld() use Fdistance and Ldistance parameters**
    $W_j = fs(X_i, X_j) + diff(X_i, X_j) + ld(y_i, y_j)$
    if $|N_i| \leq k$ **then**
      $N_i = N_i \bigcup \{X_j\}$
    end
    else
      Find an instance $X_m \in N_i$ **having max weight** W_m
      if $W_m > W_j$ **then**
        // **Replace** X_m **by** X_j
        $N_i = N_i - \{X_m\}$
        $N_i = N_i \bigcup \{X_j\}$
      end
    end
  end
end
foreach *label c in j neighbors* $(0 \leq j \leq k)$ **do**
  Estimate $Likelihood_c$**:** $P(E = j|H_c = 1)$ **and** $P(E = j|H_c = 0)$ **using Eq.4.3 and Eq.4.4 respectively**
end
$N_t = \phi$ // **Neighbors of instance t**
foreach *instance* $X_i \in MLDB$ $(1 \leq i \leq q)$ *and instance t* **do**
  // **fs() uses Fdistance parameter** $W_i = fs(X_i, X_t) + diff(X_i, X_t)$
  if $|N_t| \leq k$ **then**
    $N_t = N_t \bigcup \{X_i\}$
  end
  else
    Find an instance $X_m \in N_t$ **having max weight** W_m
    if $W_m > W_i$ **then**
      // **Replace** X_m **by** X_i
      $N_t = N_t - \{X_m\}$
      $N_t = N_t \bigcup \{X_i\}$
    end
  end
end
foreach *label c* **do**
  Predict t_c **for an instance t using** $Prior_c$ **and** $Likelihood_c$ **using Eq.4.5 and Eq.4.6 respectively**
end
end

2. *Selection of k nearest neighbors* (Lines 5-21): After calculating prior probabilities, likelihood probabilities are estimated from the knowledge obtained from k nearest neighbors (kNN). Neighbors are obtained for each instance X in MLDB. MLFLD takes into account features as well as class labels while deciding the nearest neighbors as follows (Line 8):

   (a) *Function fs(.):* is used for checking similarity of features between the instances using metric in Fdistance parameter.

   (b) *Function ld(.):* uses metric in Ldistance parameter to find label dissimilarity.

   (c) *Function diff(.):* is used to compute the difference between the values of features between the two instances. This function returns summation of absolute values of differences in features.

   Thus the information obtained from features as well as labels together is used to weigh neighbors. Initial k computed weights, for instance, X_i is considered as its k neighbors denoted by set N_i (Lines 9-11). After that, the largest weight in set N_i is replaced by newly calculated weight if new weight is smaller (Lines 12-19).

3. *Estimation of a likelihood probability distribution* (Lines 22-24): MLFLD decides how many instances in MLDB have a total number of $0, 1 \ldots k$ neighbors where each neighbor is related with label c. This information is stored in $F_1^{(c)}[0 \ldots k]$ and $F_0^{(c)}[0 \ldots k]$ arrays respectively, depending on whether instance under consideration whose neighbors are observed, is related or not related with label c. This knowledge is utilized to estimate likelihood probabilities. Two probabilities are estimated:

   (a) The probability that an instance x has j neighbors related with label c when "an instance x belongs to the label c".

$$P(E = j|H_c = 1) = \frac{p + F_1^{(c)}[j]}{p \times (1 + k) + \Sigma_{r=0}^{k} F_1^{(c)}[r])}, 0 \leq j \leq k \tag{4.3}$$

   (b) The probability that an instance x has j neighbors associated with label c when "an instance x does not belong to the label c".

$$P(E = j|H_c = 0) = \frac{p + F_0^{(c)}[j]}{p \times (1 + k) + \Sigma_{r=0}^{k} F_0^{(c)}[r])}, 0 \leq j \leq k \tag{4.4}$$

Stage two is further divided into two sub-stages:

1. *Searching k nearest neighbors of an unlabeled instance* (Lines 25-39): Computation of feature similarity using *Fdistance* metric and difference of features of an unseen instance t with each instance in MLDB is done using fs(.) and diff(.) respectively (Line 27). It is followed by selection of k nearest neighbors for the unseen instance t denoted by set N_t (Lines 28-38).

2. *Predicting labels for the unlabeled instance* (Lines 40-42): Number of neighbors of an unseen instance t from set N_t related with each label c is measured using Eq.4.5. This count, along with prior and likelihood probabilities, is used to find the ratio in Eq.4.6 to decide whether the unseen instance t is associated with the label c or not.

$$j = \sum_{m=1}^{k} N_m^{(c)} \tag{4.5}$$

$$t_c = 1, if \frac{P(H_c = 1) \times P(E = j|H_c = 1)}{P(H_c = 1) \times P(E = j|H_c = 1) + P(H_c = 0) \times P(E = j|H_c = 0)} \geq Th \tag{4.6}$$

As shown in chapter 6, experimental results show that among all the competing algorithms, ML-kNN has shown better performance. Hence time complexity of MLFLD is compared with that of ML-kNN which uses only feature similarity. Average time required for both the algorithms namely

1. ML-kNN proposed by Zhang and Zhou [12] and

2. Proposed MLFLD Algorithm

is compared. ML-kNN has a time complexity [25] of $O(q^2.f + c.q.k)$ for computing prior and likelihood probabilities and $O(q.f + l.k)$ for computation related to unlabeled instances. Whereas MLFLD has a time complexity of $O(q^2.x + c.q.k)$ and $O(q.x + l.k)$. Here *k, f, l*, and q represent a count of nearest neighbors, features, labels and instances in MLDB respectively. x denotes sum of f and l. Thus time complexity of MLFLD is more than that of ML-kNN. However, MLFLD shows better performance in terms of various performance parameters presented in chapter 6.

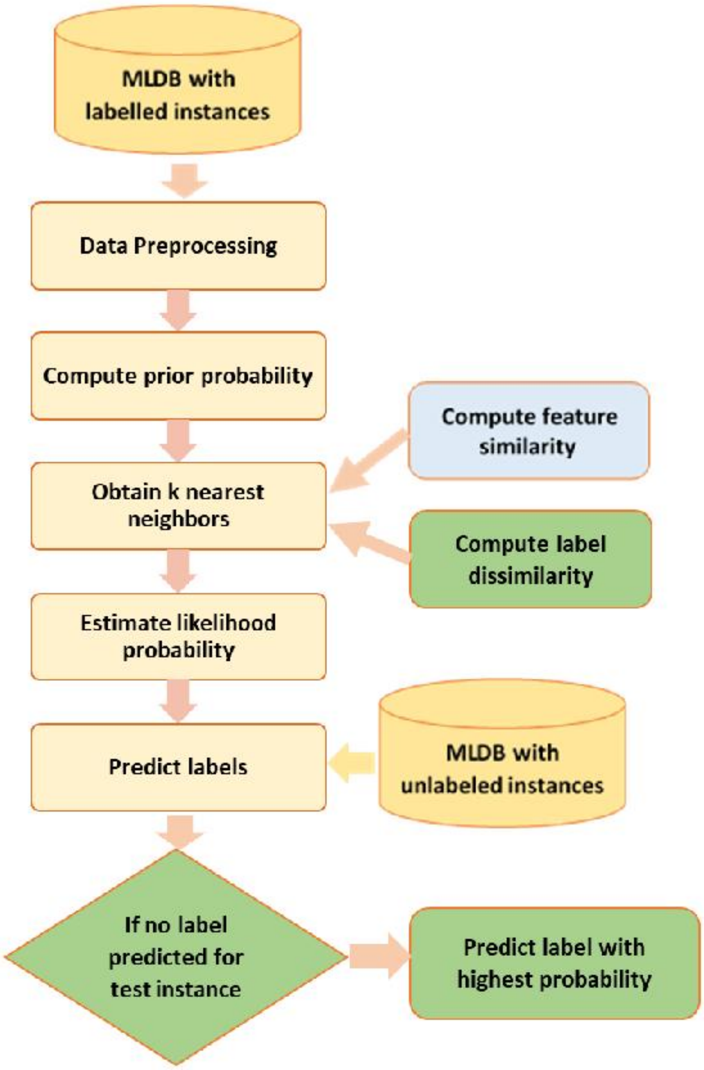


Figure 4.4: Framework for MLFLD-MAXP

Advantage of MLFLD is that it considers all the labels to find dissimilarity between labels. Thus it overcomes the drawback of the competing algorithm ML-kNN of not considering the relationship between labels, at the cost of requiring slightly more time.

4.2.2 Proposed algorithm MLFLD-MAXP

Generally it is assumed in MLC that set of labels has at least one element [97] [99] [100] [101] [102]. With this assumption, an instance is considered useless if it is not associated with any label. Otherwise, an instance is related to any number of labels.

Proposed algorithm MLFLD discussed in the previous section does not predict any label for some test instances. For such cases, MLFLD algorithm is extended to avoid no label prediction cases. Extended algorithm MLFLD with MAXimum Probability (MLFLD-MAXP) predicts that label which is the most probable for instance, under consideration among all the labels in the label set. Figure 4.4 shows the framework for Algorithm MLFLD-MAXP.

MLFLD estimates probabilities for all labels. Those labels having probabilities below the user-defined threshold are not associated with corresponding instances. For instance, under consideration, if probabilities of all labels are less than the threshold, then no label is associated with that instance. This scenario is handled by proposed algorithm MLFLD-MAXP (MLFLD with MAXimum Probability) as shown in pseudocode of the Algorithm in Figure 4.2.2 (b). It takes input parameters same as that of MLFLD, namely MLDB, number of neighbors (k), threshold (Th), smoothing parameter (p), Fdistance and Ldistance.

Algorithm 2: MLFLD-MAXP

Input : MLDB, k, Th, p, Fdistance, Ldistance

Output: Prediction of labels for unlabeled instance t

1 **begin**
2 **1-42: Prediction of labels for instance t using Algorithm MLFLD**
3 **//if no predicted label, predict label with the highest probability**
4 **if** $\forall_{c=1}^{l} t_c = 0$ **then**
5 $\quad x = argmax_c \frac{P(H_c=1) \times P(E=j|H_c=1)}{P(H_c=1) \times P(E=j|H_c=1) + P(H_c=0) \times P(E=j|H_c=0)} \geq Th$
6 $\quad$ **Set** $t_x = 1$
7 **end**
8 **end**

Lines 1-42 in Algorithm 2 are the same as that in Algorithm 1 for MLFLD. These lines predict labels, for instance, *t*. It calculates and checks ratio of the probabilities for each label *c*. If the ratio is above threshold *Th*, then label *c* is associated with instance *t*, otherwise not. If no label is associated with instance *t*, then MLFLD-MAXP algorithm predicts that label which has the highest probability computed in Line 5, because in real applications every instance belongs to at least one label to a certain extent in some context.

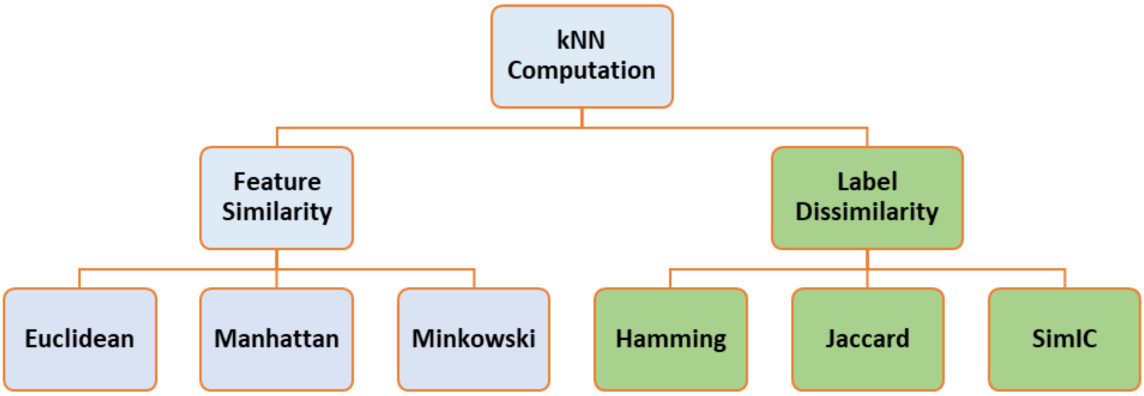


FIGURE 4.5: Distance metrics used for computing feature similarity and label dissimilarity

4.2.3 Distance metrics used for computing feature similarity and label dissimilarity

Both MLFLD and MLFLD-MAXP algorithms compute feature similarity and label dissimilarity using Fdistance and Ldistance parameters. These parameters can take values shown in Figure 4.5.

4.2.3.1 Algorithm to find feature similarity between two instances

Fdistance parameter that denotes distance metric used to compute feature similarity can take the following values:

- Euc: Use Euclidean distance (default).
- Man: Use Manhattan distance.
- Min: Use Minkowski distance.

Pseudocode for the algorithm to find feature similarity between two instances X_i and X_j is shown in Algorithm 3.

Among three distances, Euclidean distance is the most popular measure used to find feature similarity. The criterion used affects the computation time of the algorithm.

Algorithm 3: fs

Input : X_i, X_j
global: Fdistance, Number of features f

Output: Feature similarity between X_i and X_j instances

```
begin
if Fdistance = Euc then
    return sqrt(sum_{m=1}^{f} (X_im - X_jm)^2)
end
if Fdistance = Man then
    return sum_{m=1}^{f} |X_im - X_jm|
end
if Fdistance = Min then
    return (sum_{m=1}^{f} (X_im - X_jm)^3)^{1/3}
end
end
```

1. **begin**
2. **if** $Fdistance = Euc$ **then**
3. **return** $\sqrt{\sum_{m=1}^{f}(X_{im} - X_{jm})^2}$
4. **end**
5. **if** $Fdistance = Man$ **then**
6. **return** $\sum_{m=1}^{f} |X_{im} - X_{jm}|$
7. **end**
8. **if** $Fdistance = Min$ **then**
9. **return** $(\sum_{m=1}^{f}(X_{im} - X_{jm})^3)^{1/3}$
10. **end**
11. **end**

4.2.3.2 Algorithm to find label dissimilarity between two instances

Ldistance parameter that denotes distance metric used to compute label dissimilarity can take the following values:

- Hamming: Use Hamming distance (default).
- Jaccard: Use Jaccard distance.
- SimIC: Use SimIC distance.

Algorithm to find label dissimilarity is described in Algorithm 4.

Hamming distance between two strings is the number of positions where the characters of two strings are not the same. MLFLD uses a similar method to obtain the statistics from the total number of distinct labels of two instances collectively and the total number of common labels between two instances. The difference between these two values divided by a total number of labels is used to calculate label dissimilarity between two instances [30] [31]. Jaccard distance also uses union and intersection of labels to compute distance [92] [93]. For proposed SimIC distance, IC(.) denotes information content. The idea is taken from Similarity for Graphical Information Content (SimGIC) [92] [93]. It is used when labels are arranged as a hierarchy. It is different from Jaccard and Hamming in the sense that it does not count terms. It uses the information of label node like its frequency of

Algorithm 4: ld

Input : X_i, X_j
global: Ldistance, Number of labels c

Output: Label dissimilarity between X_i and X_j instances

```
begin
if Ldistance = Hamming then
    return (|Labels(X_i) ⋃ Labels(X_j)| − |Labels(X_i) ⋂ Labels(X_j)|) / c
end
if Ldistance = Jaccard then
    return 1 − |Labels(X_i) ⋂ Labels(X_j)| / |Labels(X_i) ⋃ Labels(X_j)|
end
if Ldistance = SimIC then
    return 1 − IC(Labels(X_i) ⋂ Labels(X_j)) / IC(Labels(X_i) ⋃ Labels(X_j))
end
end
```

occurrence in the hierarchy. To find IC of a set, it takes the logarithm of the multiplication of probabilities of each set element.

SimIC computes information content for class label c utilizing the prior probability of that label in the dataset denoted by p(c) as shown in Eq.4.7.

$$IC(c) = -logp(c) \tag{4.7}$$

$$IC(\{L_1, L_2 \ldots L_n\}) = \sum_{i=1}^{n} -logp(L_i) \tag{4.8}$$

For a set $A = \{L_1, L_2 \ldots L_n\}$ of labels, $IC(A)$ is calculated using the summation of the information content of each label in the set A. It adds logarithm of the probability of each set element.

4.2.4 Algorithm Multi-label Feature Selection (MLFS)

Feature selection is used in the literature effectively as it reduces the number of features. It is useful for raising the classifier performance as well as speed up the process. For multi-label classification (MLC), many researchers have proposed various methods to perform feature selection, as seen in section 3.6 of Chapter 3.

In this work, problem transformation followed by feature selection is used to find features to which each label is related. It is done for each label independently. Once features

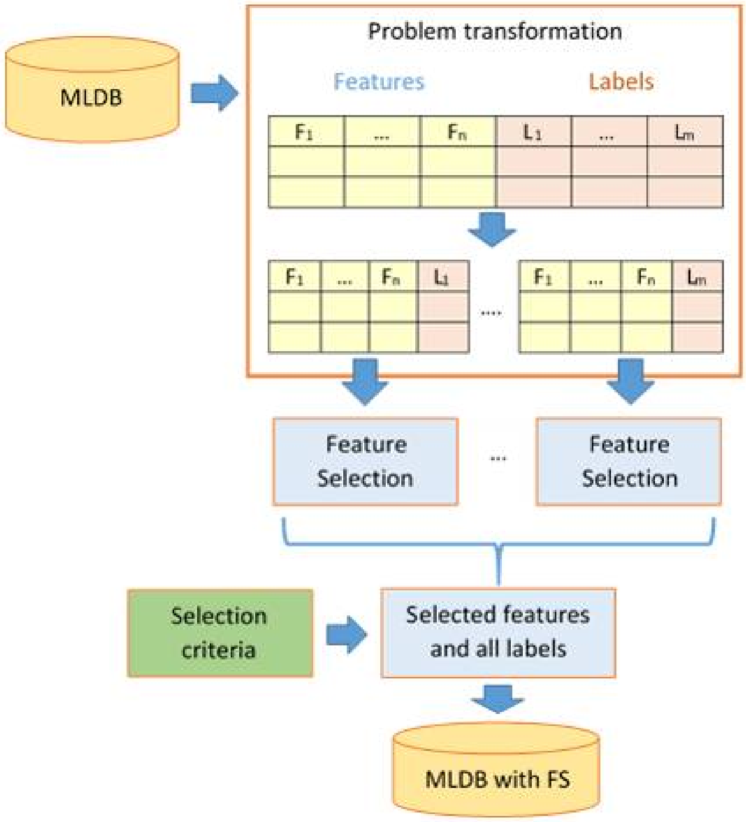


FIGURE 4.6: Framework for the Proposed Algorithm MLFS

are selected, they all are combined along with all the labels to form MLDB with selected features (MLDB with FS) as shown in Figure 4.6.

MLFS algorithm combines all the selected features for all the labels. It considers only those features that follow specific criteria during selection. Framework and pseudocode for MLFS are shown in Figure 4.6 and Algorithm 5, respectively.

Algorithm 5: MLFS

Input : MLDB Q_{fxl} with f features, l labels and q instances
Feature selection criteria θ

Output: MLDB QF_{gxl} with g features ($g \leq f$), l labels and q instances

begin
foreach *label* c **do**
    // Construct dataset with all features and only label *c*.
    $Q_c = \prod F_{1...f} L_c$
    Apply feature selection constraint θ on Q_c to get QF_c.
end
// Combine all selected features for all labels to form MLDB.
$QF = \bigcup_{c=1}^{l} QF_c$
end

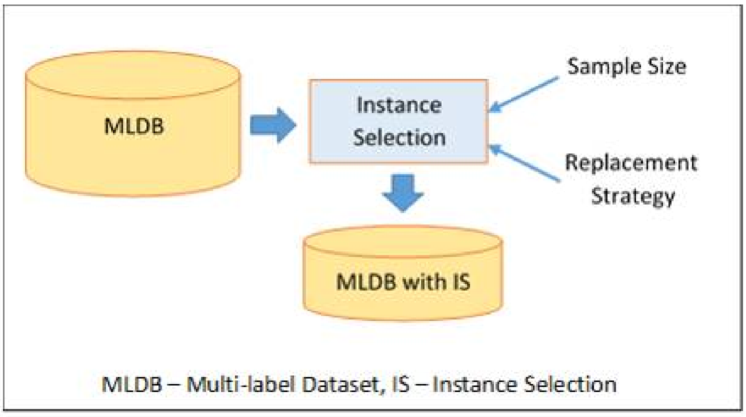


FIGURE 4.7: Framework for Algorithm MLIS

4.2.5 Algorithm Multi-label Instance Selection (MLIS)

Instance selection (sampling) can be made using two ways: with replacement and without replacement. In the former one, an instance that is already selected may be selected again (an instance may be selected one or more number of times). In the later method, an instance chosen once is not considered again for selection. Also, the sample size needs to be decided that tells count of instances to be selected from MLDB. Sampling also helps to speed up the process of classification by reducing the size of the input dataset.

Framework and pseudocode for MLIS are shown in Figure 4.7 and Algorithm 6, respectively. Algorithm MLIS takes MLDB and sampling parameters as input. Sample size parameter denotes percent of instances to be retained in MLDB.

Algorithm 6: MLIS

Input : MLDB Q_{fxl} with f features, l labels and q instances
Sampling parameters:
- Replacement strategy α (With replacement/Without replacement)
- Sample size β

Output: MLDB QI_{fxl} with f features, l labels and r instances ($r \leq q$)

begin
// Apply sampling strategy α to select β instances.
$QI = \delta_{\alpha,\beta}Q$
end

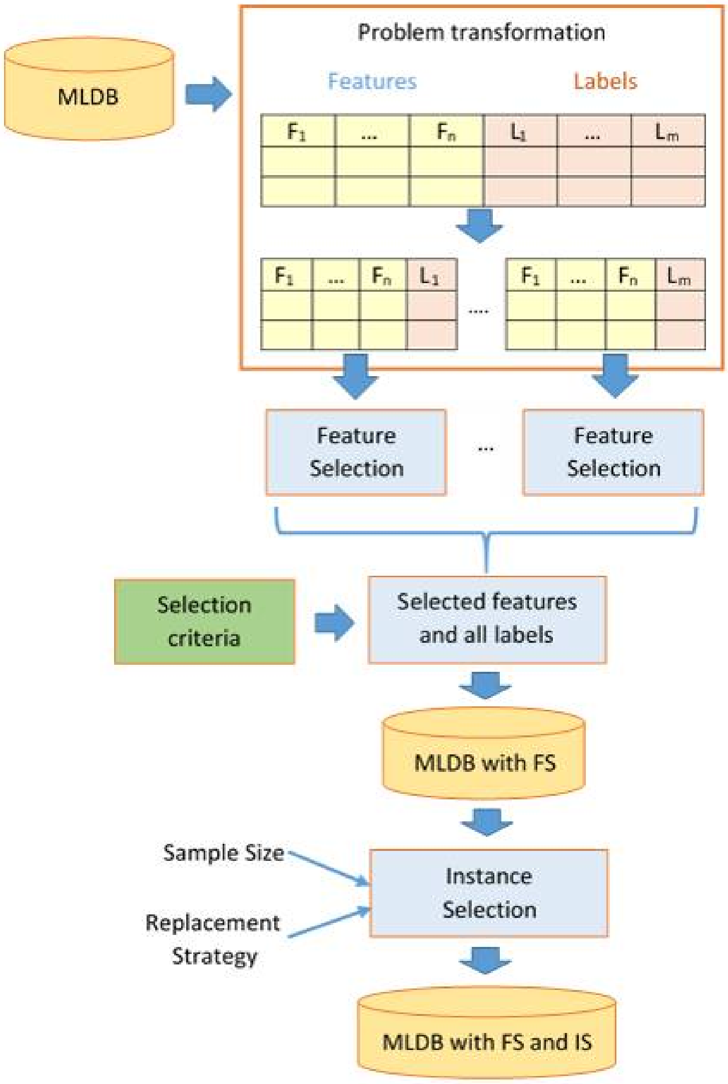


FIGURE 4.8: Framework for Algorithm MLFSIS

4.2.6 Algorithm Multi-label Feature and Instance Selection (MLFSIS)

Algorithm MLFSIS takes MLDB, feature selection criteria and sampling parameters as input. First, it performs feature selection using problem transformation. Then on the obtained MLDB sampling is performed. Framework and pseudocode for MLIS are shown in Figure 4.8 and Algorithm 7, respectively.

4.3 Expected Behaviors of Algorithms

Based on the analysis of pseudocodes for the proposed algorithms presented in section 4.2,

Algorithm 7: MLFSIS

Input : MLDB Q_{fxl} with f features, l labels and q instances
Feature selection criteria θ
Sampling parameters:
- Replacement strategy α (With replacement/Without replacement)
- Sample size β

Output: MLDB QFI_{gxl} with g features ($g \leq f$), l labels and r instances $r \leq q$

begin
foreach *label* c **do**
  // Construct dataset with all features and only label c.
  $Q_c = \prod F_{1 \dots f} L_c$
  Apply feature selection constraint θ on Q_c to get QF_c.
end
// Combine all selected features for all labels to form MLDB.
$QF = \bigcup_{c=1}^{l} QF_c$
// Apply sampling strategy α to select β instances.
$QI = \delta_{\alpha,\beta} Q$
end

- Proposed Algorithm MLFLD may result in better hamming loss than proposed algorithm MLFLD-MAXP as later algorithm assigns at least one (most probable) label to the unseen instance if no label is predicted. Forcing MLFLD-MAXP to assign at least one label may improve accuracy, subset accuracy and F1 measure, but may or may not improve hamming loss.

- Proposed Algorithms using Manhattan distance for computation of feature similarity would require minimum time and that using Minkowski would require maximum time. Euclidean distance would require time in between that of both distance metrics. It is because of square and cube operations involved in Euclidean and Minkowski respectively.

- Performance of the proposed Algorithms may vary for different distance measures used for label dissimilarity.

- Proposed Algorithms should result in better F1 measure, accuracy and subset accuracy.

- Use of multi-label feature selection followed by proposed algorithms may improve performance.

- Datasets preprocessed by multi-label feature and instance selection fed to the proposed algorithms should result in raised performance.
- Proposed algorithms should perform better on datasets when outliers are removed.
- They should finish on large datasets.
- Changing the smoothing factor and number of neighbors may not show more variation in performance.
- Threshold variation may show different performance according to a dataset.

Chapter 5

Experimental setup

Algorithms to perform multi-label classification by exploring feature similarities and label dissimilarities (MLFLD) and its extension using maximum probability (MLFLD-MAXP) are presented in chapter 4. Algorithms MLFS, MLIS and MLFSIS to describe how feature and/or instance selection can be performed before multi-label classification are also presented in the previous chapter. In this chapter, the experimental setup used for the execution of proposed algorithms and datasets used for experimentation is described in detail.

5.1 Multi-label Data

Multi-label datasets representing data from different domains are available in the literature. Some of them are Emotions from multimedia, Yeast from biology, Reuters from text and Enron from the web. Multi-label datasets are available from different resources like Mulan, MEKA and LibSVM [16-19]. Table 5.1 and Table 5.2 show brief information of benchmark multi-label datasets used for experimentation. All the datasets consist of numeric features. Datasets in the tables are roughly ordered by F x L x E. Larger datasets are handled separately using train-test splits. Cross-validation on them is not possible on the configuration used for experimentation because of memory limitation. All the datasets are normalized before using.

5.1.1 Characteristics of datasets

Elisseeff and Weston preprocessed Yeast dataset that contains information of 2417 genes [98]. Each gene is described by 103 numeric features that are associated with a subset of 14 functional classes. Each gene belongs to categories that are arranged as a hierarchy of four levels. Dataset used in this work uses only classes at the top level as used by many researchers in the literature. Some of the categories are energy and protein synthesis.

Emotional categorization of music is available in Emotions dataset. 593 tracks of music are described using 72 features and belong to a subset of 6 categories of emotions. In this dataset, Relaxing and Quiet-still are emotions that are associated with a maximum and a minimum number of tracks of songs. In Scene dataset, each instance is a still scene of the environment. Each scene is made up of 294 visual features and belongs to a subset of 6 contexts. Total of 2407 scenes is described.

Image dataset is also similar to Scene. It consists of 2000 instances where each instance is an image of a natural scene. Each image is assigned to a subset of 5 labels manually by experts [12] based on 294 features. For example, an image may describe tree and sea or sunset and mountain.

CAL500 is the only dataset among all the datasets used for experimentation in this work that has labels almost three times more than features. In the remaining datasets, feature count is lesser or equal to that of labels. Note that every label set in CAL500 is unique and occurs precisely once. That is a percentage of unique label sets is 100%.

It should be noted that all the datasets have avg. number of labels (LC) is less than five except for CAL500 that has LC 26. Cbmi09-bow and Mediamill are datasets which contain information about 43907 videos. These videos belong to 101 concepts and are described by 100 and 120 features in both datasets respectively. Cbmi09-bow has train and test splits of 22000 and 21907 respectively whereas Mediamill has train and test splits of 30993 and 12914 respectively.

First five datasets in Table 5.1 are comparatively smaller hence are used for cross-validation experiments. Whereas the last two datasets in Table 5.1 have a large number of examples. They, along with all datasets in Table 5.2, are used for train-test experiments only.

Table 5.1: Characteristics of Datasets

Datasets	Type	F	L	E	Cardinality	Density	%Unique
Emotions	Media	72	6	593	1.868	0.311	4.6
Image	Media	294	5	2000	1.236	0.247	1.0
Scene	Media	294	6	2407	1.074	0.179	0.6
Yeast	Bio	103	14	2417	4.237	0.303	8.2
CAL500	Media	68	174	502	26.044	0.15	100.0
Large Datasets							
Cbmi09-bow	Media	100	101	43907	4.376	0.043	14.9
Mediamill	Media	120	101	43907	4.376	0.043	14.9

F: #Features, L: #Labels, E: #Examples

5.1.2 Label Distribution

- **Label Cardinality, Density and Unique**

Table 5.1 and 5.3 show label cardinality (LCardinality) and label density (LDensity) of datasets [13] [18]. LCardinality and LDensity denote the average number of labels per example and the ratio of LCardinality to the number of labels, respectively [5] [24]. Unique (referred to as label diversity by some researchers [12]) represents how many combinations of labels in the dataset are distinct. Figure 5.1 shows their relation for 7 datasets.

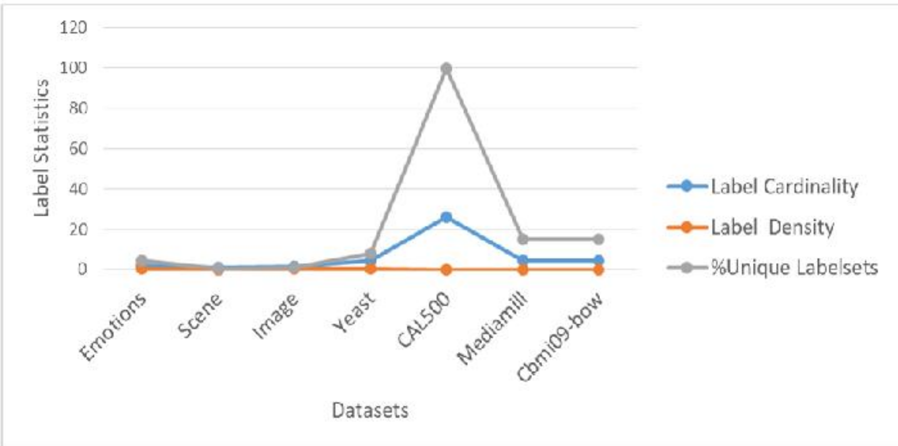


Figure 5.1: Label statistics for datasets

LCardinality of Emotions, Scene and Image is one means many instances are associated with the single label only. Yeast, Mediamill and Cbmi09-bow having LCardinality 4 indicate many instances associated with approx. 4 labels. Only CAL500 is having larger cardinality showing instances associated with approx. 26 labels.

TABLE 5.2: Statistics of datasets

Dataset	% ZLE	% MLE	%Cardinality of Ex.			%Ex/Label			%Skew	%Outlier
			Min	Avg.	Max	Min	Avg.	Max		
Emotions	0	70.0	16.7	33.3	50	25.0	31.0	44.5	13.7	18.9
Image	0	22.9	20.0	20.0	60	20.5	24.7	29.0	18.9	86.2
Scene	0	7.4	16.7	16.7	50	15.1	17.9	22.1	16.8	72.2
Yeast	0	98.7	7.1	28.6	78.6	1.4	30.2	75.1	9.8	29.6
CAL500	0	100	7.5	14.9	27.6	1.0	14.9	88.4	0.2	16.3
Large Datasets										
Cbmi09-bow	3.9	89.6	0	4.0	17.8	0.1	4.3	77.1	5.4	22.7
Mediamill	3.9	89.6	0	4.0	17.8	0.1	4.3	77.1	5.4	5.2

MLE: #Multi-Label Examples (Examples with #labels > 1)
ZLE: #Zero-Label Examples (Example with no/zero label)
Skew: Proportion of most frequent label set
Outlier: Feature having std. deviation ±1.5 (3) *from mean*

LDensity of all datasets is very small, except Emotions and Yeast followed by Image. In the first two datasets, around 30% while in the later around 25% of labels are related to almost every example.

Every label set in CAL500 is unique and occurs exactly once. That is the percentage of Unique label sets is 100%. It shows that its labelling scheme is highly irregular compared to remaining datasets. Mediamill and Cbmi09-bow have 14% unique label sets.

Only Cbmi09-bow and Mediamill datasets possess records having no (zero) relevant labels denoted by ZLE. All other datasets have no ZLE. Scene has only 7% record related to two or more labels denoted by MLE. Image has 22% MLE. All the remaining datasets have more than 70% MLE (Figure 5.2).

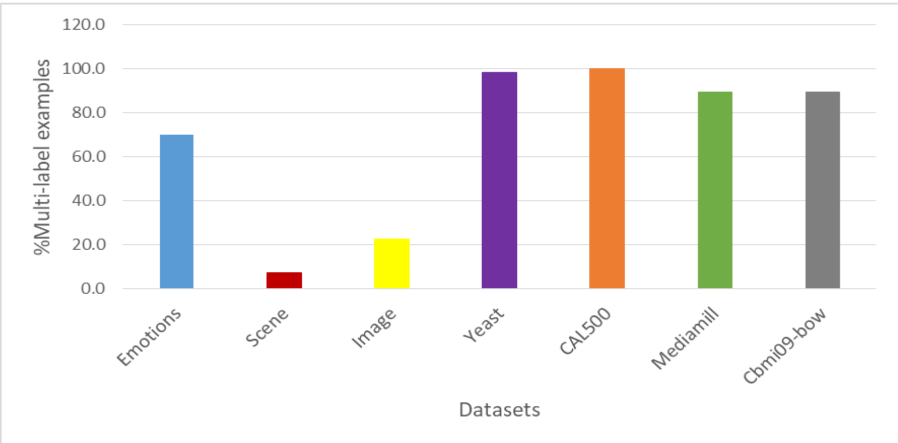


FIGURE 5.2: Statistics of Multi-label examples in datasets

Image and Scene have comparatively high label skew as shown by %Skew followed by Emotions and Yeast. That is a relatively large number of examples are related with

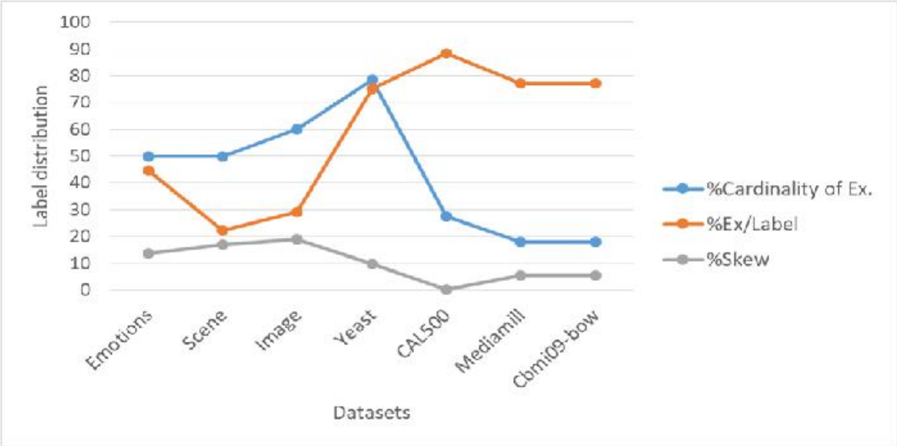


FIGURE 5.3: Label distribution for datasets

most frequently occurring label combination while remaining examples occur with rare label combination. CAL500, Cbmi09-bow and Mediamill have smaller label skew.

Examples/label (Ex/Label) also reflect the skew of labels. Observe that Scene and Image have larger skew, which is indicated by a small value of maximum %Ex/Label in Table 5.2 and Figure 5.3.

Outliers deviate performance of a classifier [75]. Generally, when a data point lies specified standard deviations away from the mean value, then it is termed as an outlier. The default value is 3. %Outlier in Table 5.2 represents the same. It can be observed that Image and Scene have a large percentage of outlier values, namely 86 and 72, respectively.

Yahoo dataset consists of data that is grouped into different categories from text-domain. It describes data of web pages. Each type describes different data. This data represents things that are linked "from web pages that belong to "yahoo.com". Data set is available at "http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.gz". These datasets have high dimensionality. Hence they are preprocessed by researchers [12] using document frequency to select terms. Nine preprocessed datasets from Yahoo shown in Table 5.3 are used for experimentation in this work.

Unique label set percent lies among 5 to 16 for Yahoo datasets. That is, label sets have less skew. Many instances are relevant to similar label sets.

Table 5.4 shows that datasets have 20% to 40% outliers approx. Reference and Society train sets have almost 48% outliers.

TABLE 5.3: Characteristics of Yahoo datasets

Datasets	F	L	Train/ Test	E	Cardi- nality	Density	Unique %
Arts	462	26	train	2000	1.63	0.06	12.7
			test	3000	1.64		11.4
Business	438	30	train	2000	1.59	0.05	4.8
			test	3000	1.59		4.4
Education	550	33	train	2000	1.47	0.04	10.0
			test	3000	1.46		7.4
Entertainment	640	21	train	2000	1.43	0.07	7.4
			test	3000	1.42		5.9
Health	612	32	train	2000	1.67	0.05	8.2
			test	3000	1.66		6.5
Reference	793	33	train	2000	1.16	0.04	6.6
			test	3000	1.18		5.4
Science	743	40	train	2000	1.49	0.04	13.1
			test	3000	1.43		9.2
Social	1047	39	train	2000	1.27	0.03	6.9
			test	3000	1.29		6.0
Society	636	27	train	2000	1.70	0.06	16.5
			test	3000	1.68		13.8

F: #Features, L: #Labels, E: #Examples

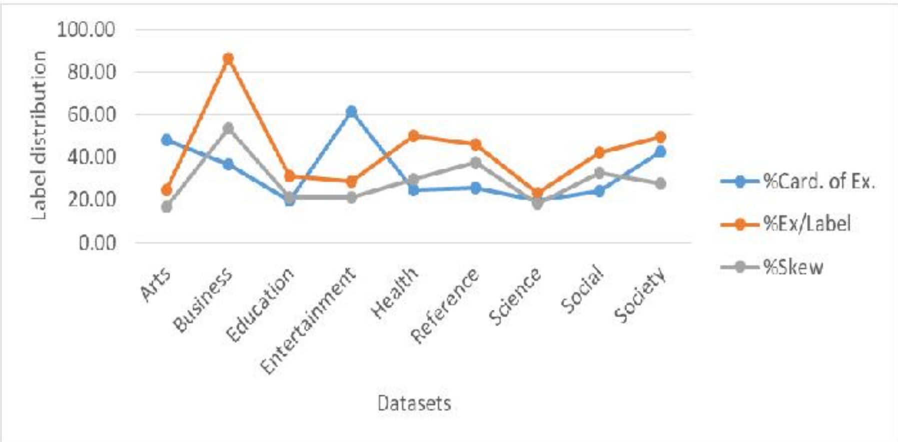


FIGURE 5.4: Label distribution for Yahoo datasets

Yahoo datasets have comparatively less skew except for Business whose labels show 53% Skew that is also reflected by 86% Ex/Label.

Observe that %Ex/label increases when %Skew decreases and vice-versa in Figure 5.3. But in Figure 5.4, it is in contrast. %Ex/label increases (decreases) when %Skew increases (decreases). In Figure 5.4, %Skew (grey) line shows opposite behavior to that of %Ex/Label (orange) line. That is, for less skew, %Ex/label is more and vice-versa.

Table 5.4: Statistics of Yahoo datasets

Datasets	**Type**	**% ZLE**	**% MLE**	**%Card. of Ex.**			**%Ex/Label**			**% Skew**	**% Outlier**
				Min	**Avg.**	**Max**	**Min**	**Avg.**	**Max**		
Arts	train	0	44.5	0.0	3.8	42.3	0	6.3	24.5	16.4	33.1
	test	0	43.6	3.8	3.8	53.8	0	6.3	24.8	17.2	34.1
Business	train	0	42.2	3.3	3.3	33.3	0	5.3	86.8	53.3	38.0
	test	0	41.9	3.3	3.3	40.0	0	5.3	86.3	53.9	38.7
Education	train	0	33.5	3.0	3.0	21.2	0.1	4.4	30.6	21.0	42.5
	test	0	33.7	3.0	3.0	18.2	0	4.4	32.2	22.2	40.7
Entertainment	train	0	29.3	4.8	4.8	42.9	0.1	6.8	28.4	21.3	32.0
	test	0	28.2	4.8	4.8	81.0	0.1	6.7	28.9	21.7	33.1
Health	Train	0	48.1	3.1	3.1	21.9	0	5.2	50.4	29.8	44.2
	Test	0	47.2	3.1	3.1	28.1	0	5.2	50.6	30.3	37.2
Reference	Train	0	13.8	3.0	3.0	15.2	0	3.5	45.9	37.7	48.5
	Test	0	14.6	3.0	3.0	36.4	0	3.5	47.0	37.6	41.1
Science	Train	0	34.9	2.5	2.5	17.5	0	3.7	22.5	17.6	35.7
	Test	0	30.6	2.5	2.5	22.5	0.1	3.5	24.1	19.0	43.4
Social	Train	0	21.0	2.6	2.6	23.1	0	3.3	40.8	32.2	28.9
	Test	0	22.8	2.6	2.6	25.6	0	3.3	43.6	34.0	31.4
Society	Train	0	41.9	3.7	3.7	48.1	0	6.3	49.1	26.7	49.9
	Test	0	40.0	3.7	3.7	37.0	0	6.2	50.2	28.7	39.9

MLE: #Multi-Label Examples (Examples with #labels > 1)
ZLE: #Zero-Label Examples (Example with no/zero label)
Skew: Proportion of most frequent label set
Outlier: Feature having std. deviation ± 1.5 (3) *from mean*

In Figure 5.4, %Skew (grey) line shows similar behavior to that of %Ex/Label (orange) line. That is, for less skew, %Ex/label is also less and vice-versa.

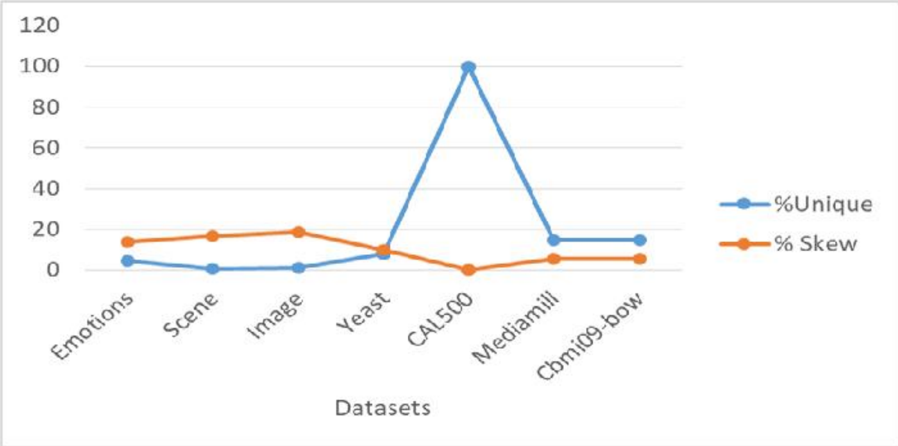


FIGURE 5.5: Skew vs Unique

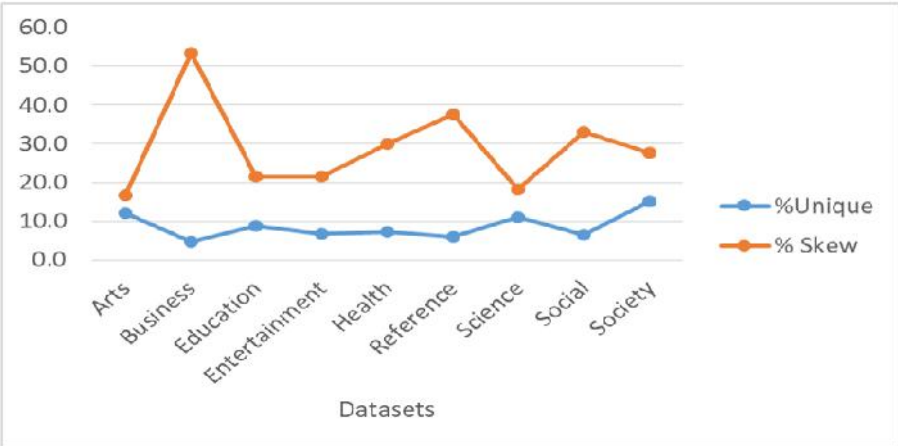


FIGURE 5.6: Skew vs Unique for Yahoo datasets

For datasets having comparatively larger skew and less unique label sets, proposed algorithms performed very well. For ex. Image, Scene, Business, Health and Social (Figure 5.6).

From Figure 5.7, it is observed that though datasets have 3 to 101 labels, almost all datasets have examples related to very less number of labels. Datasets in Figure 5.7 have a size of label space varying from 1 to 14 per example.

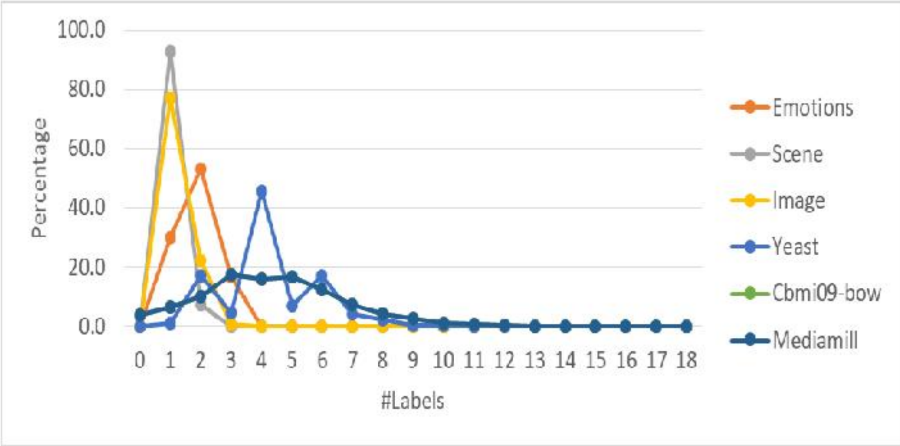


FIGURE 5.7: Percentage cardinality of labels (Number of Labels per Example) for datasets

5.2 Performance Parameters for Experimental Setup

Multi-label evaluation can be performed using different assessment parameters, as mentioned in section 2.5 of Chapter 2. Parameters that are used in this work are as follows.

Let AL_i and PL_i denote a set of actual labels of instance x_i and a set of predicted labels by $g(.)$ for the same. Let E and S be the number of instances in a dataset to be evaluated and labels in the predefined label set, respectively.

5.2.1 Example-based measures

Performance measures that compute data from individual instances and then make an average of data obtained are termed as example-based measures [25] [31].

5.2.1.1 Hamming loss

It counts the number of times actual labels of an instance do not match predicted labels.

$$HL(g_c) = \frac{1}{|E|} \sum_{i=1}^{|E|} \frac{|V(PL_i \Theta AL_i|)}{|S|} \tag{5.1}$$

where Θ denotes symmetric difference. $V(.) = 0$ if all predicted labels PL_i are the same as AL_i for an instance i, else it is 1. $HL(g_c) = 0$ means all instances are correctly classified. Smaller $HL(g_c)$ indicates better performance.

5.2.1.2 Subset Accuracy

It finds average from the exact match of the instance-wise actual label set and corresponding predicted label set for all the instances [14-20].

$$SA(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|} V(PL_i = AL_i) \tag{5.2}$$

where $V(.) = 1$ if AL_i and PL_i of instance I match, else $V(.) = 0$.

5.2.1.3 Accuracy

$$Acc(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|} \frac{|PL_i \bigcap AL_i|}{|PL_i \bigcup AL_i|} \tag{5.3}$$

5.2.1.4 Example-based F-Measure

F-Measure is used for evaluation instead of using precision and recall because it provides a balanced representation of both precision and recall measures.

$$F1(g_c) = \frac{1}{|E|}\sum_{i=1}^{|E|} \frac{2 \times |PL_i \bigcap AL_i|}{|AL_i| + |PL_i|} \tag{5.4}$$

Four ranking measures [19] given below, use a ranking function $\mu(.)$. Let $\mu(l, i)$ denotes relevance of label l with an instance i. Assume that smaller $\mu(l, i)$ shows the higher relevance of l for i.

5.2.1.5 Ranking loss

It computes whether a relevant label is ranked below a particular irrelevant label [25].

$$RL(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|} \frac{1}{|AL_i|.|\overline{AL_i}|}\{(y_r, y_{ir})|\mu(y_r, x_i) \geq \mu(y_{ir}, x_i)\}| \tag{5.5}$$

Here $\overline{AL_i}$ denotes complement of a set of relevant labels of an instance i. Elements y_r and y_{ir} are members of sets AL_i and $\overline{AL_i}$ respectively. $RL(g_r) = 0$ indicates all relevant labels are ranked above irrelevant labels for all instances. Smaller $RL(g_r)$ is desired for better performance.

5.2.1.6 Coverage

It observes the list of predicted labels to find a number of steps for inclusion of all relevant labels of each instance and computes average over all the instances. The assumption is that the most relevant label appears at the start of the list. Smaller CG(gr) indicates excellent performance.

$$CG(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|} max_{y_r \in AL_i}\mu(y_r, x_i) - 1 \tag{5.6}$$

5.2.1.7 Average precision

It determines an average value from all relevant labels ranked higher than a particular relevant label. More $AP(gr)$ indicates better performance.

$$AP(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{1}{|AL_i|}\sum_{y_{r1} \in AL_i}\frac{|\{y_{r2} \in AL_i | \mu(y_{r2}, x_i) \leq \mu(y_{r1}, x_i)\}|}{\mu(y_{r1}, x_i)} \tag{5.7}$$

Both y_{r1} and y_{r2} labels are relevant.

5.2.1.8 One-error

It determines the number of times an irrelevant label is predicted with the top rank (measures how many times a predicted label at the top rank is not in the list of relevant labels of an instance). An optimal value for $OE(g_r)$ is zero. Smaller $OE(g_r)$, better the performance [15]-[21].

$$OE(g_r) = \frac{1}{|E|}\sum_{i=1}^{|E|} argmin_{y \in S}\mu(y, x_i) \notin AL_i \tag{5.8}$$

$V(.)$ returns 0 in case of false condition, else it returns 1.

5.2.2 Label-based measures

Measures that calculate average performance from that of individual labels are termed as label-based measures. These are binary metrics based on a count of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) [15]-[21].

5.2.2.1 Macro-averaged F1 (Macro-F1)

Macro-averaging gives equal importance to all the labels. In other words, it finds an average across all the labels [3].

$$MaF1 = \frac{1}{|S|} \sum_{c=1}^{|S|} \frac{2 \times TP_c}{2 \times TP_c + FP_c + FN_c} \tag{5.9}$$

5.2.2.2 Micro-averaged F1 (Micro-F1)

Micro-averaging gives equal importance to all the instances. It finds average across all the example/label pairs [3].

$$MiF1 = \frac{2 \times \sum_{c=1}^{|S|} TP_c}{2 \times \sum_{c=1}^{|S|} TP_c + \sum_{c=1}^{|S|} FP_c + \sum_{c=1}^{|S|} FN_c} \tag{5.10}$$

Both macro and micro F1 are used in this work for evaluation. They are influenced by rare and frequent labels, respectively [3].

The macro-F measure tends to support rare labels, whereas the micro-F tends to smooth out their effect on total performance, hence being more influenced by frequent labels [3] [66].

5.3 Experimental Setup

As stated in chapter 4, feature similarity and label dissimilarity is the most crucial part of the proposed algorithms. According to distance metrics used to measure feature similarities and label dissimilarities, the performance of proposed algorithms is observed for different distance metrics. Distance metrics used for feature similarity computation are:

- Euclidean distance
- Manhattan distance
- Minkowski distance

Distance metrics used for label dissimilarity computation are:

- Hamming distance
- Jaccard distance
- SimIC distance

For experimental evaluation, the following performance metrics are used to measure the efficiency of proposed algorithms:

- *Example-based:* Hamming loss, ranking loss, one error, coverage, average precision, F1 measure, accuracy, subset accuracy
- *Label-based:* Macro-F1 and Micro-F1

The set of experiments carried out are divided into thirteen categories according to datasets and performance metrics used. The details of various experimental setups are as follows.

Set 1 Performance of MLFLD algorithm with cross-validation using Hamming distance for label dissimilarity

1.1 Evaluation of MLFLD using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

1.2 Evaluation of MLFLD using 5 datasets and 10 measures using *Manhattan* distance for feature similarity and *Hamming* distance for label dissimilarity

1.3 Evaluation of MLFLD using 5 datasets and 10 measures using *Minkowski* distance for feature similarity and *Hamming* distance for label dissimilarity

1.4 Evaluation of 7 competing algorithms using 5 datasets and 10 measures

The number of experiments conducted in set 1 is 15 (3 distance measures X 5 datasets) + 35 (7 algorithms X 5 Datasets) = 50.

Set 2 Performance of MLFLD-MAXP algorithm with cross-validation using Hamming distance for label dissimilarity

2.1 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

2.2 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Manhattan* distance for feature similarity and *Hamming* distance for label dissimilarity

2.3 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Minkowski* distance for feature similarity and *Hamming* distance for label dissimilarity

The number of experiments conducted in set 2 is 15 (3 distance measures X 5 datasets).

Set 3 Performance of MLFLD algorithm with train-test splits of datasets using Hamming distance for label dissimilarity

3.1 Evaluation of MLFLD using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

3.2 Evaluation of MLFLD using 13 datasets and 10 measures using *Manhattan* distance for feature similarity and *Hamming* distance for label dissimilarity

3.3 Evaluation of MLFLD using 13 datasets and 10 measures using *Minkowski* distance for feature similarity and *Hamming* distance for label dissimilarity

3.4 Evaluation of 7 competing algorithms using 13 datasets and 10 measures

The number of experiments conducted in set 3 is 39 (3 distance measures X 13 datasets) + 91 (7 algorithms X 13 Datasets) = 130.

Set 4 Performance of MLFLD-MAXP algorithm with train-test splits of datasets using Hamming distance for label dissimilarity

4.1 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

4.2 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Manhattan* distance for feature similarity and *Hamming* distance for label dissimilarity

4.3 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Minkowski* distance for feature similarity and *Hamming* distance for label dissimilarity

The number of experiments conducted in set 3 is 39 (3 distance measures X 13 datasets).

Set 5 Performance of MLFLD and MLFLD-MAXP algorithms with cross-validation after outlier removal from datasets

5.1 Evaluation of MLFLD using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

5.2 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

5.3 Evaluation of MLkNN using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

The number of experiments conducted in set 5 is 15 (3 algorithms X 5 datasets).

Set 6 Performance of MLFLD and MLFLD-MAXP algorithms with train-test splits of datasets after outlier removal from datasets

6.1 Evaluation of MLFLD using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

6.2 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

6.3 Evaluation of MLkNN using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

The number of experiments conducted in set 6 is 39 (3 algorithms X 13 datasets).

Set 7 Performance of MLFLD for large datasets

7.1 Evaluation of MLFLD using 2 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

7.2 Evaluation of MLFLD using 2 datasets and 10 measures using *Manhattan* distance for feature similarity and *Hamming* distance for label dissimilarity

7.3 Evaluation of MLFLD using 2 datasets and 10 measures using *Minkowski* distance for feature similarity and *Hamming* distance for label dissimilarity

7.4 Evaluation of 7 competing algorithms using 2 datasets and 10 measures

The number of experiments conducted in set 7 is 6 (3 distance measures X 2 datasets) + 14 (7 algorithms X 2 Datasets) = 20.

Set 8 Performance of MLFLD-MAXP for large datasets

8.1 Evaluation of MLFLD-MAXP using 2 datasets and 10 measures using *Euclidean* distance for feature similarity and *Hamming* distance for label dissimilarity

8.2 Evaluation of MLFLD-MAXP using 2 datasets and 10 measures using *Manhattan* distance for feature similarity and *Hamming* distance for label dissimilarity
8.3 Evaluation of MLFLD-MAXP using 2 datasets and 10 measures using *Minkowski* distance for feature similarity and *Hamming* distance for label dissimilarity

The number of experiments conducted in set 8 is 6 (3 distance measures X 2 datasets).

Set 9 Performance of MLFLD algorithm with train-test splits of datasets using Jaccard distance for label dissimilarity
9.1 Evaluation of MLFLD using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Jaccard* distance for label dissimilarity
9.2 Evaluation of MLFLD using 13 datasets and 10 measures using *Manhattan* distance for feature similarity and *Jaccard* distance for label dissimilarity
9.3 Evaluation of MLFLD using 13 datasets and 10 measures using *Minkowski* distance for feature similarity and *Jaccard* distance for label dissimilarity

The number of experiments conducted in set 9 is 39 (3 distance measures X 13 datasets).

Set 10 Performance of MLFLD-MAXP algorithm with train-test splits of datasets using Jaccard distance for label dissimilarity
10.1 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *Jaccard* distance for label dissimilarity
10.2 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Manhattan* distance for feature similarity and *Jaccard* distance for label dissimilarity
10.3 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Minkowski* distance for feature similarity and *Jaccard* distance for label dissimilarity

The number of experiments conducted in set 10 is 39 (3 distance measures X 13 datasets).

Set 11 Performance of MLFLD algorithm with cross-validation using Jaccard distance for label dissimilarity
11.1 Evaluation of MLFLD using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Jaccard* distance for label dissimilarity
11.2 Evaluation of MLFLD using 5 datasets and 10 measures using *Manhattan* distance for

feature similarity and *Jaccard* distance for label dissimilarity

11.3 Evaluation of MLFLD using 5 datasets and 10 measures using *Minkowski* distance for feature similarity and *Jaccard* distance for label dissimilarity

The number of experiments conducted in set 11 is 15 (3 distance measures X 5 datasets).

Set 12 Performance of MLFLD-MAXP algorithm with cross-validation using Jaccard distance for label dissimilarity

12.1 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *Jaccard* distance for label dissimilarity

12.2 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Manhattan* distance for feature similarity and *Jaccard* distance for label dissimilarity

12.3 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Minkowski* distance for feature similarity and *Jaccard* distance for label dissimilarity

The number of experiments conducted in set 12 is 15 (3 distance measures X 5 datasets).

Set 13 Performance of MLFLD algorithm with train-test splits of datasets using SimIC distance for label dissimilarity

13.1 Evaluation of MLFLD using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *SimIC* distance for label dissimilarity

13.2 Evaluation of MLFLD using 13 datasets and 10 measures using *Manhattan* distance for feature similarity and *SimIC* distance for label dissimilarity

13.3 Evaluation of MLFLD using 13 datasets and 10 measures using *Minkowski* distance for feature similarity and *SimIC* distance for label dissimilarity

The number of experiments conducted in set 13 is 39 (3 distance measures X 13 datasets).

Set 14 Performance of MLFLD-MAXP algorithm with train-test splits of datasets using SimIC distance for label dissimilarity

14.1 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Euclidean* distance for feature similarity and *SimIC* distance for label dissimilarity

14.2 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Manhattan* distance for feature similarity and *SimIC* distance for label dissimilarity

14.3 Evaluation of MLFLD-MAXP using 13 datasets and 10 measures using *Minkowski* distance for feature similarity and *SimIC* distance for label dissimilarity

The number of experiments conducted in set 14 is 39 (3 distance measures X 13 datasets).

Set 15 Performance of MLFLD algorithm with cross-validation using SimIC distance for label dissimilarity

15.1 Evaluation of MLFLD using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *SimIC* distance for label dissimilarity

15.2 Evaluation of MLFLD using 5 datasets and 10 measures using *Manhattan* distance for feature similarity and *SimIC* distance for label dissimilarity

15.3 Evaluation of MLFLD using 5 datasets and 10 measures using *Minkowski* distance for feature similarity and *SimIC* distance for label dissimilarity

The number of experiments conducted in set 15 is 15 (3 distance measures X 5 datasets).

Set 16 Performance of MLFLD-MAXP algorithm with cross-validation using SimIC distance for label dissimilarity

16.1 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Euclidean* distance for feature similarity and *SimIC* distance for label dissimilarity

16.2 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Manhattan* distance for feature similarity and *SimIC* distance for label dissimilarity

16.3 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures using *Minkowski* distance for feature similarity and *SimIC* distance for label dissimilarity

The number of experiments conducted in set 16 is 15 (3 distance measures X 5 datasets).

Set 17 Performance of MLFLD and MLFLD-MAXP for feature selection

17.1 Performing attribute selection on 5 datasets

17.2 Evaluation of MLFLD using 5 datasets and 10 measures following attribute selection

17.3 Evaluation of MLFLD-MAXP using 5 datasets and 10 measures following attribute selection

The number of experiments conducted in set 17 is 5 (attribute selection on 5 datasets) + 10 (2 algorithms X 5 datasets) = 15.

Set 18 Performance of MLFLD and MLFLD-MAXP for instance selection

18.1 Performing instance selection with a replacement for 60% sample size on 5 datasets
18.2 Performing instance selection with a replacement for 70% sample size on 5 datasets
18.3 Performing instance selection with a replacement for 80% sample size on 5 datasets
18.4 Evaluation of MLFLD on 5 datasets with 60% sample size
18.5 Evaluation of MLFLD on 5 datasets with 70% sample size
18.6 Evaluation of MLFLD on 5 datasets with 80% sample size
18.7 Evaluation of MLFLD-MAXP on 5 datasets with 60% sample size
18.8 Evaluation of MLFLD-MAXP on 5 datasets with 70% sample size
18.9 Evaluation of MLFLD-MAXP on 5 datasets with 80% sample size

The number of experiments conducted in set 18 is 15 (instance selection on 5 datasets X 3 sample sizes) + 30 (2 algorithms X 5 datasets X 3 sample sizes) = 45.

Set 19 Performance of MLFLD and MLFLD-MAXP for a feature and instance selection

19.1 Instance selection with a replacement for 60% sample size following attribute selection on 5 datasets
19.2 Instance selection with a replacement for 70% sample size following attribute selection on 5 datasets
19.3 Instance selection with a replacement for 80% sample size following attribute selection on 5 datasets
19.4 Evaluation of MLFLD on 5 datasets with 60% sample size
19.5 Evaluation of MLFLD on 5 datasets with 70% sample size
19.6 Evaluation of MLFLD on 5 datasets with 80% sample size
19.7 Evaluation of MLFLD-MAXP on 5 datasets with 60% sample size
19.8 Evaluation of MLFLD-MAXP on 5 datasets with 70% sample size
19.9 Evaluation of MLFLD-MAXP on 5 datasets with 80% sample size

The number of experiments conducted in set 19 is 15 (instance selection on 5 datasets X 3 sample sizes) + 30 (2 algorithms X 5 datasets X 3 sample sizes) = 45.

Set 20 Performance of MLFLD for k variation

20.1 Evaluation of MLFLD for k ranging from 5 to 15 on Emotions dataset for 10 performance measures

20.2 Evaluation of MLFLD for k ranging from 5 to 15 on Image dataset for 10 performance measures

20.3 Evaluation of MLFLD for k ranging from 5 to 15 on Scene dataset for 10 performance measures

20.4 Evaluation of MLFLD for k ranging from 5 to 15 on Yeast dataset for 10 performance measures

The number of experiments conducted in set 20 is 44 (4 datasets X 11 values of k).

Set 21 Performance of MLFLD for threshold variation

21.1 Evaluation of MLFLD for 9 values of threshold on Emotions dataset

21.2 Evaluation of MLFLD for 9 values of threshold on an Image dataset

21.3 Evaluation of MLFLD for 9 values of threshold on Scene dataset

21.4 Evaluation of MLFLD for 9 values of threshold on Yeast dataset

The number of experiments conducted in set 21 is 36 (4 datasets X 9 values of a threshold).

Set 22 Performance of MLFLD for smoothing factor variation

22.1 Evaluation of MLFLD for 4 values of smoothing factor on Emotions dataset

22.2 Evaluation of MLFLD for 4 values of smoothing factor on an Image dataset

22.3 Evaluation of MLFLD for 4 values of smoothing factor on Scene dataset

22.4 Evaluation of MLFLD for 4 values of smoothing factor on Yeast dataset

The number of experiments conducted in set 22 is 16 (4 datasets X 4 values of smoothing factor).

5.4 Experimental Process

All experiments from Set 1 to Set 22 mentioned earlier in this chapter are conducted as detailed below.

1. All the experiments in Set 1 are carried using MLFLD algorithm with cross-validation.

2. All the experiments in Set 2 are performed similarly as in Set 1 except algorithm that is MLFLD-MAXP.

3. Set 3 and Set 4 are similar to Set 1 and Set 2. The only change is the use of train-test splits of datasets instead of cross-validation.

4. Sets 5-6 are implemented using both proposed and one competing algorithm datasets free from outliers.

5. Sets 7-8 are similar to Sets 3-4 except datasets. They are run on large datasets.

6. Sets 9-12 and Sets 13-16 repeat Sets 1-4 for Jaccard and SimIC distance respectively instead of Hamming.

7. Sets 17, 18, 19 are executed out experiments to observe the performance along with feature selection, instance selection and both feature and instance selection respectively.

8. Set 1 is repeated independently for different values of k, threshold and smoothing factor resulting in Sets 20, 21 and 22, respectively.

For experimentation, java program is written and tested on Intel(R) Core(TM) i5-6200U CPU @2.30 GHz with 8GB RAM. Libraries supported by Mulan, MEKA and WEKA are imported while implementing the algorithm [10] [17] [18] [19] [20].

Description of empirical evaluation for all the sets is presented in Chapter 6.

Details of experiments are shown in Table 5.5.

Table 5.5: Details of experiments

Set No.	Details of Experiment	#Expts
1	Performance of MLFLD algorithm with cross-validation using Hamming distance for label dissimilarity	50
2	Performance of MLFLD-MAXP algorithm with cross-validation using Hamming distance for label dissimilarity	15
3	Performance of MLFLD algorithm with train-test splits of datasets using Hamming distance for label dissimilarity	130
4	Performance of MLFLD-MAXP algorithm with train-test splits of datasets using Hamming distance for label dissimilarity	39
5	Performance of MLFLD and MLFLD-MAXP algorithms with cross-validation after outlier removal from datasets	15
6	Performance of MLFLD and MLFLD-MAXP algorithms with train-test splits of datasets after outlier removal from datasets	39
7	Performance of MLFLD for large datasets	20
8	Performance of MLFLD-MAXP for large datasets	6
9	Performance of MLFLD algorithm with train-test splits of datasets using Jaccard distance for label dissimilarity	39
10	Performance of MLFLD-MAXP algorithm with train-test splits of datasets using Jaccard distance for label dissimilarity	39
11	Performance of MLFLD algorithm with cross-validation using Jaccard distance for label dissimilarity	15
12	Performance of MLFLD-MAXP algorithm with cross-validation using Jaccard distance for label dissimilarity	15
13	Performance of MLFLD algorithm with train-test splits of datasets using SimIC distance for label dissimilarity	39
14	Performance of MLFLD-MAXP algorithm with train-test splits of datasets using SimIC distance for label dissimilarity	39
15	Performance of MLFLD algorithm with cross-validation using SimIC distance for label dissimilarity	15
16	Performance of MLFLD-MAXP algorithm with cross-validation using SimIC distance for label dissimilarity	15
17	Performance of MLFLD and MLFLD-MAXP for feature selection	15
18	Performance of MLFLD and MLFLD-MAXP for instance selection	45
19	Performance of MLFLD and MLFLD-MAXP for a feature and instance selection	45
20	Performance of MLFLD for k variation	44
21	Performance of MLFLD for threshold variation	36
22	Performance of MLFLD for smoothing factor variation	16
Total no. of experiments		**695**

Chapter 6

Experimentation and Results

Chapter 4 presented a general methodology used for multi-label classification. Proposed algorithms to perform multi-label classification by exploring feature similarities and label dissimilarities (MLFLD) and its extension MLFLD-MAXP (MLFLD using maximum probability) are presented in chapter 4. Algorithms MLFS, MLIS, and MLFSIS to describe how feature and/or instance selection can be accomplished before multi-label classification are also presented in the previous chapter.

In this chapter, the performance of algorithms described in chapter 4 is analyzed with several experiments carried out, as described in chapter 5. Section 6.1 describes the different parameters and values used by algorithms. Both algorithms are first executed with Euclidean and Hamming distance for feature similarity and label dissimilarity. Distance variation for feature similarity is also carried out. Euclidean, Manhattan, and Minkowski [48] distance measures are used for feature similarity. The setup is repeated once with cross-validation and then with train-test splits of datasets as described in sections 6.2 and 6.3, respectively. All experiments are again executed with Jaccard and SimIC distance for label dissimilarity, as described in section 6.6. All the tests involved are completed using five datasets for cross-validation, thirteen datasets for train-test splits, and two large datasets also. Datasets are described in chapter 5. Datasets are also examined for outlier data. Outliers are removed from datasets, and on these datasets, experimentation is carried out using MLFLD and MLFLD-MAXP with Euclidean and Hamming distance as described in section 6.4. Evaluation for large datasets is described in section 6.5. Effect of applying algorithms MLFS, MLIS, and MLFSIS on five datasets, followed by MLFLD and MLFLD-MAXP, is explained in section 6.7 to 6.9. For all these experiments, an evaluation is carried out using eight example-based measures, namely hamming loss, ranking loss, coverage, one error, average precision, subset accuracy and accuracy, F1 measure, and two label-based macro-F1 and micro-F1 measures described in chapter 5. The performance of MLFLD for variation of parameter k, threshold, and smoothing factor is shown in the last three sections. Standard deviations are minimal, hence not shown here.

For performance comparison, MLkNN is found as the primary contestant. Hence for MLkNN, published results from [12] are used for Image and Yeast datasets for cross-validation experiments and train-test splits of Yahoo datasets. The remaining results are taken from Mulan experiments [74]. In all sections, Tables 1-10 show the performance of ten parameters. Top row in the table shows algorithms used for evaluation. The first column shows datasets used. At the end of each section, the summary table shows the algorithm-wise average of each parameter obtained over all datasets. The summary table contains performance parameters and names of algorithms in the first column and top row, respectively.

Metrics used have different desired values. Also, different metrics may affect other metrics. Hence it is not possible to improve all the metrics simultaneously. Therefore for comparison, an algorithm-wise average of each metric is obtained over all datasets. Then metric-wise, each algorithm is given rank with the best performing algorithm getting position 1. The second-best performing algorithm is getting position 2, and so on. All algorithms showing the same performance for a particular metric will get the same rank for that metric. When ranks of all algorithms are available for ten parameters, then the average rank of each algorithm is computed. The algorithm-wise count of rank one is treated as #Wins (number of wins) for that algorithm. Algorithm having the smallest average rank and maximum #Wins is considered as the best performer for a given setup of experimentation.

6.1 Values for various input parameters used for experiments

The state-of-the-art multi-label classifiers that include BR, LP, CC, RAkEL, BRkNN, BPMLL, and MLkNN are available in Mulan [74]. Parameters used by these

Table 6.1: Values for various input parameters used by competing algorithms

Sr. No.	Algorithm	Base classifier	Other parameters
1	BR	Decision tree (J48)	-
2	LP	Decision tree (J48)	-
3	CC	Decision tree (J48)	-
4	RAkEL	LP with J48	k = 3, m = 6 Threshold = 0.5
5	BRkNN	-	k = 10
6	BPMLL	-	Default
7	MLkNN	-	k=10, Smoothing factor=1

Table 6.2: Values for various input parameters used for MLFLD and MLFLD-MAXP

Sr. No.	Parameter	Value	Description
1	k	10	Number of neighbors
2	P	1	Smoothing factor
3	Th	0.5	Threshold
4	Fdistance	Euc/Man/Min	Distance metric used to compute feature similarity (Euclidean / Manhattan / Minkowski)
5	Ldistance	H/J/S	Distance metric used to compute label dissimilarity (Hamming / Jaccard / SimIC)

competing algorithms and MLFLD and MLFLD-MAXP are shown in Table 6.1 and Table 6.2, respectively.

All experiments are carried on Intel(R) Core(TM) i5-6200U CPU @2.30 GHz with 8GB RAM. Libraries in Mulan, as well as MEKA and WEKA [73] [74] [75] are used with Java.

6.2 Performance of proposed algorithms with cross-validation using Hamming distance for label dissimilarity

This section presents an evaluation of proposed algorithms with cross-validation. Here Hamming distance is used for label dissimilarity. Symbols $\downarrow$ and $\uparrow$ used throughout the chapter denote smaller and higher values desired for the corresponding metrics, respectively. Desired hamming and ranking loss, coverage, and one error are lower whereas expected avg precision, subset accuracy, accuracy, example-based F measure (hereafter used as Ex-F1), macro and micro F1 are higher.

Note that among competing algorithms, MLkNN has performed better among all. Hence the performance of proposed algorithms is mainly compared with MLkNN, and percentage improvement over MLkNN is mentioned.

6.2.1 Performance of MLFLD algorithm with cross-validation (CV) using Hamming distance

Evaluation of MLFLD is carried out using Euclidian and Hamming distance for feature similarity and label dissimilarity, respectively. Tables 6.3 to 6.12 show the assessment of ten parameters and summarized in Table 6.13.

TABLE 6.3: Performance of MLFLD (CV) for Hamming loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.2425	0.2704	0.2534	0.2412	**0.1922**	0.2104	0.1959	0.1938
Image	0.2277	0.2310	0.2266	0.1958	0.1729	0.5794	0.1690	**0.1631**
Scene	0.1316	0.1476	0.1379	0.1188	0.0924	0.2507	0.0861	**0.0797**
Yeast	0.2469	0.2752	0.2675	0.2487	0.1952	0.2247	**0.1940**	0.1981
CAL500	0.1608	0.2000	0.1760	0.1539	0.1425	0.2501	**0.1388**	0.1394
Average	0.2019	0.2248	0.2123	0.1917	0.1590	0.3031	0.1568	**0.1548**
Rank	5	7	6	4	3	8	2	1

TABLE 6.4: Performance of MLFLD (CV) for Ranking loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.3042	0.3407	0.2964	0.2228	0.1593	0.1595	0.1594	**0.1483**
Image	0.3051	0.3062	0.2967	0.2045	0.1805	0.4450	0.1680	**0.1570**
Scene	0.2391	0.2216	0.2323	0.1315	0.0936	0.1645	0.0775	**0.0682**
Yeast	0.3110	0.3966	0.3227	0.3559	0.1778	0.1845	**0.1670**	0.1689
CAL500	0.3023	0.6508	0.3679	0.6111	0.2310	**0.1773**	0.1828	0.1835
Average	0.2923	0.3832	0.3032	0.3052	0.1684	0.2262	0.1509	**0.1452**
Rank	5	8	6	7	3	4	2	1

Table 6.5: Performance of MLFLD (CV) for One Error (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.3948	0.4267	0.3929	0.3373	0.2597	0.2900	0.2699	**0.2492**
Image	0.4730	0.4645	0.4360	0.3440	0.3300	0.6855	0.3000	**0.2916**
Scene	0.4117	0.4067	0.3722	0.3079	0.2655	0.5393	0.2256	**0.2050**
Yeast	0.4013	0.5123	0.3554	0.2975	0.2309	0.2441	**0.2300**	0.2378
CAL500	0.7312	0.9880	0.6975	0.7669	0.1893	0.1376	0.1176	**0.1160**
Average	0.4824	0.5596	0.4508	0.4107	0.2551	0.3793	0.2286	**0.2199**
Rank	7	8	6	5	3	4	2	1

Table 6.6: Performance of MLFLD (CV) for Coverage (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	2.5896	2.7083	2.5206	2.1349	1.7831	1.7343	1.7764	**1.7102**
Image	1.4885	1.4855	1.4570	1.0835	0.9845	2.0025	0.9390	**0.8964**
Scene	1.2958	1.2085	1.2671	0.7478	0.5551	0.9032	0.4753	**0.4258**
Yeast	9.2345	9.3515	8.8229	10.0333	6.5245	6.5208	**6.2750**	6.2905
CAL500	169.50	170.85	170.15	170.97	150.74	**128.72**	130.56	130.52
Average	36.822	37.122	36.843	36.994	32.119	27.977	28.005	**27.969**
Rank	5	8	6	7	4	2	3	1

Table 6.7: Performance of MLFLD (CV) for Average Precision (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.6938	0.6707	0.6996	0.7519	0.8060	0.8026	0.8034	**0.8183**
Image	0.6778	0.6786	0.6960	0.7709	0.7867	0.5378	0.8030	**0.8105**
Scene	0.7148	0.7222	0.7336	0.8061	0.8412	0.6929	0.8652	**0.8785**
Yeast	0.6203	0.5740	0.6310	0.6190	0.7599	0.7477	**0.7650**	0.7648
CAL500	0.3548	0.1171	0.3156	0.1391	0.4589	**0.5081**	0.4942	0.4918
Average	0.6123	0.5525	0.6152	0.6174	0.7305	0.6578	0.7462	**0.7528**
Rank	7	8	6	5	3	4	2	1

Table 6.8: Performance of MLFLD (CV) for Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.4549	0.4490	0.4739	0.4871	0.5186	**0.5573**	0.5340	0.5483
Image	0.4417	0.4902	0.5046	0.5275	0.4643	0.2098	0.4937	**0.5588**
Scene	0.5461	0.5791	0.6049	0.6034	0.6204	0.3780	0.6635	**0.7083**
Yeast	0.4376	0.4162	0.4287	0.3844	0.5002	**0.5197**	0.5162	0.5116
CAL500	0.2085	0.2036	0.2293	0.0243	0.1856	**0.2969**	0.1972	0.2023
Average	0.4178	0.4276	0.4483	0.4053	0.4578	0.3923	0.4809	**0.5059**
Rank	6	5	4	7	3	8	2	1

Table 6.9: Performance of MLFLD (CV) for Subset Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.1956	0.2092	0.2329	0.2057	0.2917	0.2767	0.2934	**0.3051**
Image	0.2885	0.3755	0.3880	0.3915	0.4025	0.0210	0.4090	**0.4632**
Scene	0.4449	0.5351	0.5521	0.5239	0.5974	0.0694	0.6248	**0.6629**
Yeast	0.0674	0.1324	0.1539	0.0385	0.1982	0.1403	0.1874	**0.2046**
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.1993	0.2504	0.2654	0.2319	0.2980	0.1015	0.3029	**0.3272**
Rank	7	5	4	6	3	8	2	1

Table 6.10: Performance of MLFLD (CV) for Ex-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.5414	0.5315	0.5542	0.5788	0.5936	**0.6488**	0.6141	0.6274
Image	0.4970	0.5302	0.5454	0.5750	0.4852	0.3100	0.5223	**0.5916**
Scene	0.5815	0.5940	0.6227	0.6303	0.6281	0.4995	0.6764	**0.7235**
Yeast	0.5620	0.5199	0.5288	0.5112	0.5984	**0.6315**	0.6204	0.6109
CAL500	0.3396	0.3277	0.3623	0.0461	0.3059	**0.4486**	0.3240	0.3311
Average	0.5043	0.5007	0.5227	0.4683	0.5222	0.5077	0.5514	**0.5769**
Rank	6	7	3	8	4	5	2	1

TABLE 6.11: Performance of MLFLD (CV) for Macro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.5736	0.5520	0.5798	0.6228	0.6282	**0.6630**	0.6226	0.6584
Image	0.5406	0.5319	0.5447	0.5941	0.5492	0.3254	0.5815	**0.6287**
Scene	0.6423	0.5992	0.6318	0.6709	0.6996	0.5558	0.7364	**0.7683**
Yeast	0.3911	0.3834	0.3987	0.2732	0.3960	**0.4339**	0.3853	NaN
CAL500	0.2134	0.1937	0.2435	0.1233	0.1893	**0.2445**	0.1714	NaN
Average	0.4722	0.4520	0.4797	0.4569	0.4925	0.4445	0.4994	**0.6851**
Rank	5	7	4	6	3	8	2	1

**NaN: denotes Not a Number*

TABLE 6.12: Performance of MLFLD (CV) for Micro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.5970	0.5603	0.5908	0.6238	0.6539	**0.6774**	0.6610	0.6727
Image	0.5384	0.5312	0.5438	0.5933	0.5542	0.3537	0.5842	**0.6259**
Scene	0.6312	0.5885	0.6189	0.6627	0.7006	0.5341	0.7332	**0.7617**
Yeast	0.5840	0.5436	0.5507	0.5343	0.6344	**0.6472**	0.6471	0.6426
CAL500	0.3421	0.3322	0.3664	0.0464	0.3085	**0.4566**	0.3209	0.3294
Average	0.5385	0.5112	0.5341	0.4921	0.5703	0.5338	0.5893	**0.6065**
Rank	4	7	5	8	3	6	2	1

Observations: When the performance of all algorithms is compared over the average rank of ten metrics obtained for five datasets, MLFLD has outperformed, showing the smallest avg rank in Table 6.13. Also, it has demonstrated 10 on 10 wins. Metric-wise performance of MLFLD is as given below:

- MLFLD has outshined in subset accuracy for all datasets with an overall 8% improvement. For CAL500, it is not able to improve but is similar to other algorithms.
- For accuracy, a total of 5% improvement is seen over that of MLkNN though only Image and Scene are showing growth for MLFLD.

- 4%, 6%, and 3% overall improvement are seen for Ex-F1, macro, and micro F1 respectively while showing improved F measures for Image and Scene only. For Yeast and CAL500, MLFLD is not able to measure macro F1 denoted by NaN (Not a Number).
- MLFLD has outperformed by a 1% improvement with the smallest average hamming loss over MLkNN that has shown the second-lowest average hamming loss. However, MLFLD has demonstrated the least misclassification for Image and Scene datasets only.
- It has demonstrated improved ranking loss and one error for three datasets and an overall 4% improvement over MLkNN for both metrics.
- MLFLD has shown improved coverage and avg precision for three datasets and an overall 1% improvement over MLkNN for avg precision while coverage is similar to that of MLkNN.

Table 6.13: Summary of MLFLD (CV) performance using Hamming distance

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
HamLoss	0.2019	0.2248	0.2123	0.1917	0.159	0.3031	0.1568	**0.1548**
RankLoss	0.2923	0.3832	0.3032	0.3052	0.1684	0.2262	0.1509	**0.1452**
OneError	0.4824	0.5596	0.4508	0.4107	0.2551	0.3793	0.2286	**0.2199**
Coverage	36.822	37.122	36.843	36.994	32.119	27.977	28.005	**27.969**
AvgPrec	0.6123	0.5525	0.6152	0.6174	0.7305	0.6578	0.7462	**0.7528**
Accuracy	0.4178	0.4276	0.4483	0.4053	0.4578	0.3923	0.4809	**0.5059**
SubAcc	0.1993	0.2504	0.2654	0.2319	0.298	0.1015	0.3029	**0.3272**
Ex-F1	0.5043	0.5007	0.5227	0.4683	0.5222	0.5077	0.5514	**0.5769**
Macro-F1	0.4722	0.452	0.4797	0.4569	0.4925	0.4445	0.4994	**0.6851**
Micro-F1	0.5385	0.5112	0.5341	0.4921	0.5703	0.5338	0.5893	**0.6065**
Avg Rank	5.7	7.0	5.0	6.3	3.2	5.7	2.1	**1.0**
#Wins	0	0	0	0	0	0	0	**10**

6.2.2 Performance of MLFLD-MAXP algorithm with cross-validation using Hamming distance

In this section, the evaluation of MLFLD-MAXP carried out using Euclidian and Hamming distance for feature similarity and label dissimilarity, respectively, is presented in Table 6.14 to 6.23. MLFLD-MAXP is denoted by MAXP in the following tables.

TABLE 6.14: Performance of MLFLD-MAXP (CV) for Hamming loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.2425	0.2704	0.2534	0.2412	**0.1922**	0.2104	0.1959	0.1938
Image	0.2277	0.2310	0.2266	0.1958	0.1729	0.5794	0.1690	**0.1656**
Scene	0.1316	0.1476	0.1379	0.1188	0.0924	0.2507	0.0861	**0.0812**
Yeast	0.2469	0.2752	0.2675	0.2487	0.1952	0.2247	**0.1940**	0.1977
CAL500	0.1608	0.2000	0.1760	0.1539	0.1425	0.2501	**0.1388**	0.1394
Average	0.2019	0.2248	0.2123	0.1917	0.1590	0.3031	0.1568	**0.1555**
Rank	5	7	6	4	3	8	2	1

MLFLD-MAXP has shown a 0.7% improvement in avg hamming loss for five datasets though it has shown an improved hamming loss for only Image and Scene individually.

TABLE 6.15: Performance of MLFLD-MAXP (CV) for Ranking loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.3042	0.3407	0.2964	0.2228	0.1593	0.1595	0.1594	**0.1483**
Image	0.3051	0.3062	0.2967	0.2045	0.1805	0.4450	0.1680	**0.1570**
Scene	0.2391	0.2216	0.2323	0.1315	0.0936	0.1645	0.0775	**0.0682**
Yeast	0.3110	0.3966	0.3227	0.3559	0.1778	0.1845	**0.1670**	0.1689
CAL500	0.3023	0.6508	0.3679	0.6111	0.2310	**0.1773**	0.1828	0.1835
Average	0.2923	0.3832	0.3032	0.3052	0.1684	0.2262	0.1509	**0.1452**
Rank	5	8	6	7	3	4	2	1

Table 6.16: Performance of MLFLD-MAXP (CV) for One Error (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.3948	0.4267	0.3929	0.3373	0.2597	0.2900	0.2699	**0.2492**
Image	0.4730	0.4645	0.4360	0.3440	0.3300	0.6855	0.3000	**0.2916**
Scene	0.4117	0.4067	0.3722	0.3079	0.2655	0.5393	0.2256	**0.2050**
Yeast	0.4013	0.5123	0.3554	0.2975	0.2309	0.2441	**0.2300**	0.2378
CAL500	0.7312	0.9880	0.6975	0.7669	0.1893	0.1376	0.1176	**0.1160**
Average	0.4824	0.5596	0.4508	0.4107	0.2551	0.3793	0.2286	**0.2199**
Rank	7	8	6	5	3	4	2	1

For one error, overall improvement is 3% while four datasets are showing reduced one error performance with MLFLD-MAXP.

Table 6.17: Performance of MLFLD-MAXP (CV) for Coverage (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	2.5896	2.7083	2.5206	2.1349	1.7831	1.7343	1.7764	**1.7102**
Image	1.4885	1.4855	1.4570	1.0835	0.9845	2.0025	0.9390	**0.8964**
Scene	1.2958	1.2085	1.2671	0.7478	0.5551	0.9032	0.4753	**0.4258**
Yeast	9.2345	9.3515	8.8229	10.033	6.5245	6.5208	**6.2750**	6.2905
CAL500	169.51	170.86	170.15	170.97	150.75	**128.73**	130.56	130.52
Average	36.823	37.122	36.844	36.994	32.119	27.978	28.006	**27.969**
Rank	5	8	6	7	4	2	3	1

Coverage for the first three datasets is improved, though improvement for coverage is only 0.1% that is performance is almost similar for the proposed and competing algorithm.

Table 6.18: Performance of MLFLD-MAXP (CV) for Average Precision (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.6938	0.6707	0.6996	0.7519	0.8060	0.8026	0.8034	**0.8183**
Image	0.6778	0.6786	0.6960	0.7709	0.7867	0.5378	0.8030	**0.8105**
Scene	0.7148	0.7222	0.7336	0.8061	0.8412	0.6929	0.8652	**0.8785**
Yeast	0.6203	0.5740	0.6310	0.6190	0.7599	0.7477	**0.7650**	0.7648
CAL500	0.3548	0.1171	0.3156	0.1391	0.4589	**0.5081**	0.4942	0.4918
Average	0.6123	0.5525	0.6152	0.6174	0.7305	0.6578	0.7462	**0.7528**
Rank	7	8	6	5	3	4	2	1

Table 6.19: Performance of MLFLD-MAXP (CV) for Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.4549	0.4490	0.4739	0.4871	0.5186	0.5573	0.5340	**0.5627**
Image	0.4417	0.4902	0.5046	0.5275	0.4643	0.2098	0.4937	**0.6169**
Scene	0.5461	0.5791	0.6049	0.6034	0.6204	0.3780	0.6635	**0.7599**
Yeast	0.4376	0.4162	0.4287	0.3844	0.5002	**0.5197**	0.5162	0.5140
CAL500	0.2085	0.2036	0.2293	0.0243	0.1856	**0.2969**	0.1972	0.2023
Average	0.4178	0.4276	0.4483	0.4053	0.4578	0.3923	0.4809	**0.5312**
Rank	6	5	4	7	3	8	2	1

Table 6.20: Performance of MLFLD-MAXP (CV) for Subset Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.1956	0.2092	0.2329	0.2057	0.2917	0.2767	0.2934	**0.3136**
Image	0.2885	0.3755	0.3880	0.3915	0.4025	0.0210	0.4090	**0.5108**
Scene	0.4449	0.5351	0.5521	0.5239	0.5974	0.0694	0.6248	**0.7117**
Yeast	0.0674	0.1324	0.1539	0.0385	0.1982	0.1403	0.1874	**0.2046**
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.1993	0.2504	0.2654	0.2319	0.2980	0.1015	0.3029	**0.3481**
Rank	7	5	4	6	3	8	2	1

Table 6.21: Performance of MLFLD-MAXP (CV) for Ex-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.5414	0.5315	0.5542	0.5788	0.5936	**0.6488**	0.6141	0.6441
Image	0.4970	0.5302	0.5454	0.5750	0.4852	0.3100	0.5223	**0.6532**
Scene	0.5815	0.5940	0.6227	0.6303	0.6281	0.4995	0.6764	**0.7761**
Yeast	0.5620	0.5199	0.5288	0.5112	0.5984	**0.6315**	0.6204	0.6145
CAL500	0.3396	0.3277	0.3623	0.0461	0.3059	**0.4486**	0.3240	0.3311
Average	0.5043	0.5007	0.5227	0.4683	0.5222	0.5077	0.5514	**0.6038**
Rank	6	7	3	8	4	5	2	1

Table 6.22: Performance of MLFLD-MAXP (CV) for Macro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.5736	0.5520	0.5798	0.6228	0.6282	**0.6630**	0.6226	0.6609
Image	0.5406	0.5319	0.5447	0.5941	0.5492	0.3254	0.5815	**0.6482**
Scene	0.6423	0.5992	0.6318	0.6709	0.6996	0.5558	0.7364	**0.7795**
Yeast	0.3911	0.3834	0.3987	0.2732	0.3960	**0.4339**	0.3853	NaN
CAL500	0.2134	0.1937	0.2435	0.1233	0.1893	**0.2445**	0.1714	NaN
Average	0.4722	0.4520	0.4797	0.4569	0.4925	0.4445	0.4994	**0.6962**
Rank	5	7	4	6	3	8	2	1

Table 6.23: Performance of MLFLD-MAXP (CV) for Micro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.5970	0.5603	0.5908	0.6238	0.6539	**0.6774**	0.6610	0.6766
Image	0.5384	0.5312	0.5438	0.5933	0.5542	0.3537	0.5842	**0.6449**
Scene	0.6312	0.5885	0.6189	0.6627	0.7006	0.5341	0.7332	**0.7706**
Yeast	0.5840	0.5436	0.5507	0.5343	0.6344	**0.6472**	0.6471	0.6439
CAL500	0.3421	0.3322	0.3664	0.0464	0.3085	**0.4566**	0.3209	0.3294
Average	0.5385	0.5112	0.5341	0.4921	0.5703	0.5338	0.5893	**0.6131**
Rank	4	7	5	8	3	6	2	1

For all F measures, MLFLD-MAXP ranked first with a 9% rise for ex-F1. It also outperformed in overall micro-F by 4% that shows MLFLD-MAXP more influenced by frequent labels.

TABLE 6.24: Summary of MLFLD-MAXP (CV) performance using Hamming distance

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
HamLoss	0.2019	0.2248	0.2123	0.1917	0.1590	0.3031	0.1568	**0.1555**
RankLoss	0.2923	0.3832	0.3032	0.3052	0.1684	0.2262	0.1509	**0.1452**
OneError	0.4824	0.5596	0.4508	0.4107	0.2551	0.3793	0.2286	**0.2199**
Coverage	36.823	37.122	36.844	36.994	32.119	27.978	28.006	**27.969**
AvgPrec	0.6123	0.5525	0.6152	0.6174	0.7305	0.6578	0.7462	**0.7528**
Accuracy	0.4178	0.4276	0.4483	0.4053	0.4578	0.3923	0.4809	**0.5312**
SubAcc	0.1993	0.2504	0.2654	0.2319	0.2980	0.1015	0.3029	**0.3481**
Ex-F1	0.5043	0.5007	0.5227	0.4683	0.5222	0.5077	0.5514	**0.6038**
Macro-F1	0.4722	0.4520	0.4797	0.4569	0.4925	0.4445	0.4994	**0.6962**
Micro-F1	0.5385	0.5112	0.5341	0.4921	0.5703	0.5338	0.5893	**0.6131**
Avg Rank	5.7	7	5	6.3	3.2	5.7	2.1	**1**
#Wins	0	0	0	0	0	0	0	**10**

Observations: Average values of ten metrics for the performance of MLFLD-MAXP is shown in Table 6.24. Again the proposed algorithm has won with the smallest avg rank 1 and 10 wins. Metric wise observations are:

- MLFLD-MAXP has shown a 15% improvement over the second performing MLkNN. Four datasets have got better subset accuracy denoting that MLFLD-MAXP has improved the ability to predict all the labels of an instance correctly.
- It has also shown improved ability of accurate prediction by 10%.
- Ex-F1, macro, and micro F1 are improved by 9%, 7%, and 4% overall, respectively, and only for Image and Scene individually.
- For both one error and rank loss, overall improvement is 3%, while four and three datasets are showing reduced values with MLFLD-MAXP, respectively.
- 1% improvement for avg precision is seen with three datasets showing better performance.

- MLFLD-MAXP has shown a 0.7% improvement in avg hamming loss for five datasets though it has shown an improved hamming loss for only Image and Scene individually.
- Coverage for the first three datasets is improved, though improvement for coverage is only 0.1%, denoting that performance is almost similar for the proposed and comparing algorithm.

6.2.3 Comparison of MLFLD and MLFLD-MAXP performance with cross-validation using Hamming distance

The performance of both proposed algorithms is compared in this section for evaluation carried out using Euclidian and Hamming distance in Table 6.25 to 6.34.

TABLE 6.25: Performance of MLFLD and MLFLD-MAXP (CV) for Hamming loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.2425	0.2704	0.2534	0.2412	**0.1922**	0.2104	0.1959	0.1938	0.1938
Image	0.2277	0.2310	0.2266	0.1958	0.1729	0.5794	0.1690	**0.1631**	0.1656
Scene	0.1316	0.1476	0.1379	0.1188	0.0924	0.2507	0.0861	**0.0797**	0.0812
Yeast	0.2469	0.2752	0.2675	0.2487	0.1952	0.2247	**0.1940**	0.1981	0.1977
CAL500	0.1608	0.2000	0.1760	0.1539	0.1425	0.2501	**0.1388**	0.1394	0.1394
Average	0.2019	0.2248	0.2123	0.1917	0.1590	0.3031	0.1568	**0.1548**	0.1555
Rank	6	8	7	5	4	9	3	1	2

Both algorithms have shown improved hamming loss by 1% and 0.7% over competing algorithms. MLFLD has resulted in smaller hamming loss than MLFLD-MAXP for Image and Scene datasets.

TABLE 6.26: Performance of MLFLD and MLFLD-MAXP (CV) for Ranking loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3042	0.3407	0.2964	0.2228	0.1593	0.1595	0.1594	**0.1483**	**0.1483**
Image	0.3051	0.3062	0.2967	0.2045	0.1805	0.4450	0.1680	**0.1570**	**0.1570**
Scene	0.2391	0.2216	0.2323	0.1315	0.0936	0.1645	0.0775	**0.0682**	**0.0682**
Yeast	0.3110	0.3966	0.3227	0.3559	0.1778	0.1845	**0.1670**	0.1689	0.1689
CAL500	0.3023	0.6508	0.3679	0.6111	0.2310	**0.1773**	0.1828	0.1835	0.1835
Average	0.2923	0.3832	0.3032	0.3052	0.1684	0.2262	0.1509	**0.1452**	**0.1452**
Rank	6	9	7	8	4	5	3	1	1

Table 6.27: Performance of MLFLD and MLFLD-MAXP (CV) for One error (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3948	0.4267	0.3929	0.3373	0.2597	0.2900	0.2699	**0.2492**	**0.2492**
Image	0.4730	0.4645	0.4360	0.3440	0.3300	0.6855	0.3000	**0.2916**	**0.2916**
Scene	0.4117	0.4067	0.3722	0.3079	0.2655	0.5393	0.2256	**0.2050**	**0.2050**
Yeast	0.4013	0.5123	0.3554	0.2975	0.2309	0.2441	**0.2300**	0.2378	0.2378
CAL500	0.7312	0.9880	0.6975	0.7669	0.1893	0.1376	0.1176	**0.1160**	**0.1160**
Average	0.4824	0.5596	0.4508	0.4107	0.2551	0.3793	0.2286	**0.2199**	**0.2199**
Rank	8	9	7	6	4	5	3	1	1

Overall approx. 4% improvement is shown by both algorithms for hamming and rank loss. Reduction is seen for both losses with Emotions, Image, and Scene. Rank loss is the only metric that proposed algorithms could improve for CAL500.

Table 6.28: Performance of MLFLD and MLFLD-MAXP (CV) for Coverage (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	2.5896	2.7083	2.5206	2.1349	1.7831	1.7343	1.7764	**1.7102**	**1.7102**
Image	1.4885	1.4855	1.4570	1.0835	0.9845	2.0025	0.9390	**0.8964**	**0.8964**
Scene	1.2958	1.2085	1.2671	0.7478	0.5551	0.9032	0.4753	**0.4258**	**0.4258**
Yeast	9.2345	9.3515	8.8229	10.033	6.5245	6.5208	**6.2750**	6.2905	6.2905
CAL500	169.51	170.86	170.15	170.97	150.75	**128.73**	130.56	130.52	130.52
Average	36.823	37.122	36.844	36.994	32.119	27.978	28.006	**27.969**	**27.969**
Rank	6	9	7	8	5	3	4	1	1

Table 6.29: Performance of MLFLD and MLFLD-MAXP (CV) for Average Precision (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.6938	0.6707	0.6996	0.7519	0.8060	0.8026	0.8034	**0.8183**	**0.8183**
Image	0.6778	0.6786	0.6960	0.7709	0.7867	0.5378	0.8030	**0.8105**	**0.8105**
Scene	0.7148	0.7222	0.7336	0.8061	0.8412	0.6929	0.8652	**0.8785**	**0.8785**
Yeast	0.6203	0.5740	0.6310	0.6190	0.7599	0.7477	**0.7650**	0.7648	0.7648
CAL500	0.3548	0.1171	0.3156	0.1391	0.4589	**0.5081**	0.4942	0.4918	0.4918
Average	0.6123	0.5525	0.6152	0.6174	0.7305	0.6578	0.7462	**0.7528**	**0.7528**
Rank	8	9	7	6	4	5	3	1	1

TABLE 6.30: Performance of MLFLD and MLFLD-MAXP (CV) for Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.4549	0.4490	0.4739	0.4871	0.5186	0.5573	0.5340	0.5483	**0.5627**
Image	0.4417	0.4902	0.5046	0.5275	0.4643	0.2098	0.4937	0.5588	**0.6169**
Scene	0.5461	0.5791	0.6049	0.6034	0.6204	0.3780	0.6635	0.7083	**0.7599**
Yeast	0.4376	0.4162	0.4287	0.3844	0.5002	**0.5197**	0.5162	0.5116	0.5140
CAL500	0.2085	0.2036	0.2293	0.0243	0.1856	**0.2969**	0.1972	0.2023	0.2023
Average	0.4178	0.4276	0.4483	0.4053	0.4578	0.3923	0.4809	0.5059	**0.5312**
Rank	7	6	5	8	4	9	3	2	1

TABLE 6.31: Performance of MLFLD and MLFLD-MAXP (CV) for Subset Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.1956	0.2092	0.2329	0.2057	0.2917	0.2767	0.2934	0.3051	**0.3136**
Image	0.2885	0.3755	0.3880	0.3915	0.4025	0.0210	0.4090	0.4632	**0.5108**
Scene	0.4449	0.5351	0.5521	0.5239	0.5974	0.0694	0.6248	0.6629	**0.7117**
Yeast	0.0674	0.1324	0.1539	0.0385	0.1982	0.1403	0.1874	**0.2046**	**0.2046**
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.1993	0.2504	0.2654	0.2319	0.2980	0.1015	0.3029	0.3272	**0.3481**
Rank	8	6	5	7	4	9	3	2	1

TABLE 6.32: Performance of MLFLD and MLFLD-MAXP (CV) for Ex-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.5414	0.5315	0.5542	0.5788	0.5936	**0.6488**	0.6141	0.6274	0.6441
Image	0.4970	0.5302	0.5454	0.5750	0.4852	0.3100	0.5223	0.5916	**0.6532**
Scene	0.5815	0.5940	0.6227	0.6303	0.6281	0.4995	0.6764	0.7235	**0.7761**
Yeast	0.5620	0.5199	0.5288	0.5112	0.5984	**0.6315**	0.6204	0.6109	0.6145
CAL500	0.3396	0.3277	0.3623	0.0461	0.3059	**0.4486**	0.3240	0.3311	0.3311
Average	0.5043	0.5007	0.5227	0.4683	0.5222	0.5077	0.5514	0.5769	**0.6038**
Rank	7	8	4	9	5	6	3	2	1

14% improvement for a sub. accuracy and 10% for accuracy and Ex-F1 by MLFLD-MAXP, that is almost twice that of MLFLD.

TABLE 6.33: Performance of MLFLD and MLFLD-MAXP (CV) for Macro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.5736	0.5520	0.5798	0.6228	0.6282	**0.6630**	0.6226	0.6584	0.6609
Image	0.5406	0.5319	0.5447	0.5941	0.5492	0.3254	0.5815	0.6287	**0.6482**
Scene	0.6423	0.5992	0.6318	0.6709	0.6996	0.5558	0.7364	0.7683	**0.7795**
Yeast	0.3911	0.3834	0.3987	0.2732	0.3960	**0.4339**	0.3853	NaN	NaN
CAL500	0.2134	0.1937	0.2435	0.1233	0.1893	**0.2445**	0.1714	NaN	NaN
Average	0.4722	0.4520	0.4797	0.4569	0.4925	0.4445	0.4994	0.6851	**0.6962**
Rank	6	8	5	7	4	9	3	2	1

TABLE 6.34: Performance of MLFLD and MLFLD-MAXP (CV) for Micro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.5970	0.5603	0.5908	0.6238	0.6539	**0.6774**	0.6610	0.6727	0.6766
Image	0.5384	0.5312	0.5438	0.5933	0.5542	0.3537	0.5842	0.6259	**0.6449**
Scene	0.6312	0.5885	0.6189	0.6627	0.7006	0.5341	0.7332	0.7617	**0.7706**
Yeast	0.5840	0.5436	0.5507	0.5343	0.6344	**0.6472**	0.6471	0.6426	0.6439
CAL500	0.3421	0.3322	0.3664	0.0464	0.3085	**0.4566**	0.3209	0.3294	0.3294
Average	0.5385	0.5112	0.5341	0.4921	0.5703	0.5338	0.5893	0.6065	**0.6131**
Rank	5	8	6	9	4	7	3	2	1

Again for micro-F1, MLFLD-MAXP is seen 4% enhanced that is double compared to MLFLD.

TABLE 6.35: Summary of MLFLD and MLFLD-MAXP (CV) performance for Micro-F1 (↑) using Hamming distance

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
HamLoss	0.2019	0.2248	0.2123	0.1917	0.1590	0.3031	0.1568	**0.1548**	0.1555
RankLoss	0.2923	0.3832	0.3032	0.3052	0.1684	0.2262	0.1509	**0.1452**	**0.1452**
OneError	0.4824	0.5596	0.4508	0.4107	0.2551	0.3793	0.2286	**0.2199**	**0.2199**
Coverage	36.823	37.122	36.844	36.994	32.119	27.978	28.006	**27.969**	**27.969**
AvgPrec	0.6123	0.5525	0.6152	0.6174	0.7305	0.6578	0.7462	**0.7528**	**0.7528**
Accuracy	0.4178	0.4276	0.4483	0.4053	0.4578	0.3923	0.4809	0.5059	**0.5312**
SubAcc	0.1993	0.2504	0.2654	0.2319	0.2980	0.1015	0.3029	0.3272	**0.3481**
Ex-F1	0.5043	0.5007	0.5227	0.4683	0.5222	0.5077	0.5514	0.5769	**0.6038**
Macro-F1	0.4722	0.4520	0.4797	0.4569	0.4925	0.4445	0.4994	0.6851	**0.6962**
Micro-F1	0.5385	0.5112	0.5341	0.4921	0.5703	0.5338	0.5893	0.6065	**0.6131**
Avg Rank	6.7	8	6	7.3	4.2	6.7	3.1	**1.5**	**1.1**
#Wins	0	0	0	0	0	0	0	**5**	**9**

Observations: Table 6.35 has shown a comparison of proposed algorithms with 7 other algorithms where MLFLD-MAXP has obtained the smallest avg rank followed by MLFLD, while both are showing improvement over remaining. They have got 9 and 5 wins respectively. To summarize,

- For both accuracy and three F measure metrics, MLFLD-MAXP has better performance than MLFLD while it is the same for coverage, one error, avg precision and rank loss.
- Example-wise predictions for all labels are 15%, and 8% improved shown by subset accuracy. However, accuracy, subset accuracy, and ex-F1 measure improvements of MLFLD-MAXP are almost double compared to that of MLFLD and more than approx. 1% for label-based measures.
- MLFLD has done fewer misclassifications than MLFLD-MAXP showing better HamLoss.

Few more points are realized. They are as follows.

- If individual datasets are monitored, then proposed algorithms using Euclidean and Hamming distances have shown
  - All ten metrics improved for Image and Scene.
  - Improvement is seen in 5 metrics for Emotions.
  - Improved one metric each for Yeast and CAL500.
  - Same performance for coverage, one error, rank loss, and avg precision.
- Observations from Table 5.3 and 5.4 in Chapter 5 regarding datasets that may affect performance of classifier are
  - Scene and Image have maximum outliers among all datasets followed by Emotions.
  - Scene and Image show more skew that is also reflected by less percent of maximum Ex/Label, followed by slightly less skew and more Ex/Label for Emotions among all datasets.

Table 6.36: Comparison of MLFLD and MLFLD-MAXP Performance (cross-validation) with Hamming distance

- The Unique number of label sets is minimum for Scene and Image and slightly more for Emotions.
- All these observations implicate that proposed algorithms are sensitive to the presence of outliers. They are also affected by skew and the unique characteristics of datasets. That's why improvement for all metrics for Image and Scene and half metric for Emotions.
- CAL500 has every label combination unique shown by 100%, followed by Yeast. CAL500 has a maximum cardinality (26.044) followed by 4.237 by Yeast among all. It is reflected by both datasets with almost all examples possessing multiple labels shown by 100 and 98 %MLE (multi-label examples), respectively.
- It can be concluded that proposed algorithms may be more prone to datasets having very high MLE.

6.2.4 Effect of distance variation for feature similarity on the performance of proposed algorithms using Hamming distance for label dissimilarity

By keeping Hamming distance for label dissimilarity the same, change for feature similarity distance is done and evaluated in this section for MLFLD and MLFLD-MAXP. Euclidian, Manhattan, and Minkowski measures are used for feature similarity.

TABLE 6.37: Effect of distance variation on Hamming Loss (↓) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1959	0.1938	0.1929	**0.1918**	0.1938	0.1932	0.1929
Image	0.1690	0.1631	0.1630	**0.1620**	0.1656	0.1651	0.1638
Scene	0.0861	0.0797	**0.0792**	**0.0792**	0.0812	0.0810	0.0805
Yeast	**0.1940**	0.1981	0.1941	0.1990	0.1977	0.1940	0.1989
CAL500	**0.1388**	0.1394	0.1399	0.1398	0.1394	0.1399	0.1398
Average	0.1568	0.1548	**0.1538**	0.1544	0.1555	0.1546	0.1552
Rank	7	4	1	2	6	3	5

Table 6.38: Effect of distance variation on Ranking Loss (↓) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1594	**0.1483**	0.1493	0.1526	**0.1483**	0.1493	0.1526
Image	0.1680	0.1570	0.1570	**0.1565**	0.1570	0.1570	**0.1565**
Scene	0.0775	0.0682	0.0690	**0.0652**	0.0682	0.0690	**0.0652**
Yeast	0.1670	0.1689	**0.1666**	0.1731	0.1689	**0.1666**	0.1731
CAL500	**0.1828**	0.1835	0.1833	0.1834	0.1835	0.1833	0.1834
Average	0.1509	0.1452	**0.1450**	0.1462	0.1452	**0.1450**	0.1462
Rank	7	3	1	5	3	1	5

Table 6.39: Effect of distance variation on One Error (↓) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2699	**0.2492**	0.2525	0.2576	**0.2492**	0.2525	0.2576
Image	0.3000	0.2916	0.2906	**0.2896**	0.2916	0.2906	**0.2896**
Scene	0.2256	0.2050	0.2062	**0.2008**	0.2050	0.2062	**0.2008**
Yeast	**0.2300**	0.2378	0.2320	0.2402	0.2378	0.2320	0.2402
CAL500	0.1176	**0.1160**	**0.1160**	0.1240	**0.1160**	**0.1160**	0.1240
Average	0.2286	0.2199	**0.2195**	0.2224	0.2199	**0.2195**	0.2224
Rank	7	3	1	5	3	1	5

Table 6.40: Effect of distance variation on Coverage (↓) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	1.7764	**1.7102**	1.7254	1.7356	**1.7102**	1.7254	1.7356
Image	0.9390	0.8964	0.8994	**0.8920**	0.8964	0.8994	**0.8920**
Scene	0.4753	0.4258	0.4300	**0.4071**	0.4258	0.4300	**0.4071**
Yeast	6.2750	6.2905	**6.2573**	6.3386	6.2905	**6.2573**	6.3386
CAL500	130.56	130.524	**130.284**	130.678	130.524	**130.284**	130.678
Average	28.006	27.9694	**27.9192**	28.0103	27.9694	**27.9192**	28.0103
Rank	5	3	1	6	3	1	6

TABLE 6.41: Effect of distance variation on Average Precision (↑) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.8034	**0.8183**	0.8150	0.8101	**0.8183**	0.8150	0.8101
Image	0.8030	0.8105	0.8105	**0.8125**	0.8105	0.8105	**0.8125**
Scene	0.8652	0.8785	0.8777	**0.8829**	0.8785	0.8777	**0.8829**
Yeast	0.7650	0.7648	**0.7670**	0.7612	0.7648	**0.7670**	0.7612
CAL500	**0.4942**	0.4918	0.4918	0.4901	0.4918	0.4918	0.4901
Average	0.7462	**0.7528**	0.7524	0.7514	**0.7528**	0.7524	0.7514
Rank	7	1	3	5	1	3	5

Both algorithms show the same performance for coverage, rank loss, oneErr, and avgPrec with improvement 3-4% for the first 2 metrics and 0.1-0.9% for the remaining 2 metrics over that of competing algorithm.

TABLE 6.42: Effect of distance variation on Accuracy (↑) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5340	0.5483	0.5488	0.5465	**0.5627**	0.5612	0.5620
Image	0.4937	0.5588	0.5460	0.5613	0.6169	0.6158	**0.6203**
Scene	0.6635	0.7083	0.7094	0.7107	0.7599	0.7605	**0.7642**
Yeast	0.5162	0.5116	0.5208	0.5121	0.5140	**0.5211**	0.5132
CAL500	0.1972	0.2023	0.1975	**0.2034**	0.2023	0.1975	**0.2034**
Average	0.4809	0.5059	0.5045	0.5068	0.5312	0.5312	**0.5326**
Rank	7	5	6	4	2	2	1

TABLE 6.43: Effect of distance variation on Subset Accuracy (↑) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2934	0.3051	0.3017	0.2966	**0.3136**	0.3085	0.3017
Image	0.4090	0.4632	0.4552	0.4612	0.5108	0.5123	**0.5138**
Scene	0.6248	0.6629	0.6671	0.6658	0.7117	0.7150	**0.7167**
Yeast	0.1874	**0.2046**	0.2021	0.1979	0.2046	0.2021	0.1979
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3029	0.3272	0.3252	0.3243	**0.3481**	0.3476	0.3460
Rank	7	4	5	6	1	2	3

Table 6.44: Effect of distance variation on Ex-F1 (↑) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6141	0.6274	0.6292	0.6282	0.6441	0.6436	**0.6475**
Image	0.5223	0.5916	0.5770	0.5954	0.6532	0.6511	**0.6565**
Scene	0.6764	0.7235	0.7237	0.7258	0.7761	0.7758	**0.7802**
Yeast	0.6204	0.6109	0.6220	0.6113	0.6145	**0.6226**	0.6131
CAL500	0.3240	0.3311	0.3249	**0.3325**	0.3311	0.3249	**0.3325**
Average	0.5514	0.5769	0.5754	0.5786	0.6038	0.6036	**0.6060**
Rank	7	5	6	4	2	3	1

All MAXP variations have shown a 15% improvement while 10% by MLFLD variations for subAcc. Whereas prior shows 10% improvement in accuracy and Ex-F1 that is twice than that of MLFLD.

Table 6.45: Effect of distance variation on Macro-F1 (↑) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6226	0.6584	0.6500	0.6633	0.6609	0.6522	**0.6686**
Image	0.5815	0.6287	0.6203	0.6308	0.6482	0.6455	**0.6512**
Scene	0.7364	0.7683	0.7689	0.7673	0.7795	0.7799	**0.7816**
Yeast	**0.3853**	NaN	NaN	NaN	NaN	NaN	NaN
CAL500	**0.1714**	NaN	NaN	NaN	NaN	NaN	NaN
Average	0.4994	0.6851	0.6797	0.6871	0.6962	0.6925	**0.7005**
Rank	7	5	6	4	2	3	1

Table 6.46: Effect of distance variation on Micro-F1 (↑) using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6610	0.6727	0.6733	0.6746	0.6766	0.6765	**0.6787**
Image	0.5842	0.6259	0.6170	0.6298	0.6449	0.6420	**0.6485**
Scene	0.7332	0.7617	0.7625	0.7622	0.7706	0.7707	**0.7725**
Yeast	0.6471	0.6426	0.6509	0.6415	0.6439	**0.6511**	0.6421
CAL500	0.3209	0.3294	0.3224	**0.3306**	0.3294	0.3224	**0.3306**
Average	0.5893	0.6065	0.6052	0.6077	0.6131	0.6125	**0.6145**
Rank	7	5	6	4	2	3	1

Again MAXP variations have beaten MLFLD in micro and macro F1 over five and three datasets, respectively. MAXP-Minkowski combination has shown better F measure performance among all at the cost of more computation time.

Table 6.47: Summary of effect of distance variation on MLFLD and MLFLD-MAXP performance using Hamming distance and cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	0.1568	0.1548	**0.1538**	0.1544	0.1555	0.1546	0.1552
RankLoss	0.1509	0.1452	**0.1450**	0.1462	0.1452	**0.1450**	0.1462
OneError	0.2286	0.2199	**0.2195**	0.2224	0.2199	**0.2195**	0.22-24
Coverage	28.006	27.9694	**27.9192**	28.0103	27.9694	**27.9192**	28.0103
AvgPrec	0.7462	**0.7528**	0.7524	0.7514	**0.7528**	0.7524	0.7514
Accuracy	0.4809	0.5059	0.5045	0.5068	0.5312	0.5312	**0.5326**
SubAcc	0.3029	0.3272	0.3252	0.3243	**0.3481**	0.3476	0.3460
Ex-F1	0.5514	0.5769	0.5754	0.5786	0.6038	0.6036	**0.6060**
Macro-F1	0.4994	0.6851	0.6797	0.6871	0.6962	0.6925	**0.7005**
Micro-F1	0.5893	0.6065	0.6052	0.6077	0.6131	0.6125	**0.6145**
ExecTime	17	60	57	70	58	54	65
Avg Rank	6.8	3.8	3.6	4.5	2.5	**2.2**	3.3
#Wins	0	1	4	0	2	**3**	4

Table 6.47 has summarized the performance of proposed algorithms by changing measures for feature similarity while using Hamming distance for label dissimilarity. When the performance is compared with MLkNN, it is noticed that

- MLFLD-MAXP with Manhattan has outperformed with the smallest avg rank among all seven experimentations, but having only 3 wins. It takes minimum time among our six setups. All setups require 3-4 times extra time than MLkNN.
- MLFLD with Manhattan and MLFLD-MAXP with Minkowski, both have four wins showing similar avg rank. Former requires less execution time than later.
- MLFLD-MAXP with X distance measure is better than MLFLD with the same measure for both accuracies and three F measures; same for OneErr, Coverage, AvgPrec, and RankLoss; but no improvement for only HamLoss.
- All MLFLD-MAXP variations are better than MLFLD variations for 2 accuracies and 3 F measures.

- Observations for Image and Scene in Table 6.37 to Table 6.46 have shown that the MAXP-Minkowski combination has worked better for them, indicating that this combination is less prone to outlier and skew.

6.3 Performance of proposed algorithms with train-test splits of datasets using Hamming distance for label dissimilarity

For thirteen datasets that are used by various researchers in the form of train and test data, experiments carried in section 6.2 are repeated. In this section, Train-Test splits of datasets are abbreviated as **TrTe**.

6.3.1 Performance of MLFLD algorithm using train and test splits of datasets with Hamming distance

The previous section focused on cross-validation experiments. This section describes the performance of the MLFLD algorithm using train and test splits of datasets with Euclidean and Hamming distance shown in Table 6.48 to 6.57.

From Table 6.48, MLFLD is found to improve ham loss for Scene and Image. It stood at rank two among eight competing algorithms though it is showing performance slightly less than MLkNN. For the first three datasets, MLFLD has reduced one err, rank loss, coverage, and increased avg precision, and accuracy. It stood at rank 2 for the first 5 metrics, among others.

Table 6.48: Performance of MLFLD (TrTe) for Hamming Loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.3144	0.3226	0.3317	0.3053	0.2170	0.2467	**0.2162**	0.2195
Scene	0.1364	0.1469	0.1377	0.1307	0.1080	0.2395	0.0962	**0.0863**
Image	0.1390	0.2323	0.1683	0.1840	0.1153	0.2713	0.1147	**0.1127**
Yeast	0.2766	0.2977	0.2898	0.2757	0.2029	0.2422	**0.2008**	0.2072
Arts Humanity	0.0703	0.0891	0.0737	0.0677	0.0912	0.7743	**0.0612**	0.0628
Business Eco.	0.0332	0.0383	0.0337	0.0309	0.0285	0.4181	**0.0269**	0.0285
Education	0.0494	0.0633	0.0530	0.0481	0.0406	0.5215	**0.0387**	0.0465
Entertainment	0.0692	0.0817	0.0713	0.0681	0.0887	0.5909	**0.0604**	0.0722
Health	0.0425	0.0512	0.0439	0.0502	0.0936	0.3693	**0.0458**	0.0512
Reference	0.0320	0.0416	0.0320	0.0314	0.0622	0.4103	**0.0314**	0.0354
Science	0.0403	0.0554	0.0454	0.0387	0.0351	0.6759	**0.0325**	0.0358
Social Science	0.0268	0.0340	0.0265	0.0335	0.0290	0.0331	**0.0218**	0.0287
Society Culture	0.0682	0.0844	0.0678	0.0669	0.0555	0.5076	**0.0537**	0.0585
Average	0.0999	0.1183	0.1058	0.1024	0.0898	0.4077	**0.0769**	0.0804
Rank	4	7	6	5	3	8	1	2

Table 6.49: Performance of MLFLD (TrTe) for Ranking Loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.3650	0.4050	0.4086	0.2951	0.1694	0.1952	0.1781	**0.1570**
Scene	0.2315	0.2171	0.2350	0.1591	0.1173	0.1740	0.0930	**0.0830**
Image	0.1382	0.2240	0.1999	0.1769	0.0924	0.3337	0.1154	**0.0888**
Yeast	0.3551	0.4311	0.3397	0.3888	0.1902	0.2011	**0.1766**	0.1839
Arts Humanity	0.2645	0.3958	0.2481	0.4067	0.2670	0.4292	**0.1514**	0.1707
Business Eco.	0.1150	0.2946	0.1239	0.2689	0.0729	0.1635	**0.0373**	0.0454
Education	0.2270	0.5558	0.2138	0.4859	0.1744	0.3746	**0.0800**	0.1112
Entertainment	0.2353	0.4822	0.2650	0.4707	0.2755	0.4254	**0.1151**	0.1735
Health	0.1502	0.4289	0.1484	0.6860	0.3145	0.2459	**0.0605**	0.0788
Reference	0.1831	0.4526	0.1787	0.4217	0.2656	0.2894	**0.0919**	0.1367
Science	0.2485	0.4828	0.2653	0.5390	0.2719	0.4789	**0.1167**	0.1551
Social Science	0.1511	0.3441	0.1440	0.6310	0.1299	0.4045	**0.0561**	0.0767
Society Culture	0.2720	0.4048	0.2168	0.4602	0.2093	0.4622	**0.1338**	0.1543
Average	0.2259	0.3938	0.2298	0.4146	0.1962	0.3214	**0.1081**	0.1242
Rank	4	7	5	8	3	6	1	2

TABLE 6.50: Performance of MLFLD (TrTe) for One Error (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.4356	0.5396	0.4901	0.4059	0.3069	0.3267	0.3218	**0.2970**
Scene	0.4189	0.4047	0.3687	0.3470	0.3010	0.5510	0.2425	**0.2191**
Image	0.2417	0.4100	0.3383	0.3350	0.2267	0.6483	0.2517	**0.2183**
Yeast	0.3915	0.5703	0.3479	0.3631	0.2595	0.3217	**0.2519**	0.2835
Arts Humanity	0.6413	0.7153	**0.6243**	0.7960	0.9043	0.9817	0.6330	0.7323
Business Eco.	0.2653	0.3443	0.2270	0.1843	0.1273	0.9877	**0.1213**	0.1343
Education	0.6317	0.7647	0.6313	0.7340	0.5983	0.9957	**0.5207**	0.6710
Entertainment	0.5887	0.6213	0.5713	0.6387	0.7487	0.9640	**0.5300**	0.6897
Health	**0.4027**	0.5200	0.4167	0.8090	0.7307	0.9937	0.4190	0.5070
Reference	0.5110	0.5823	0.5230	0.5937	0.9520	0.9823	**0.4730**	0.5227
Science	0.6827	0.7847	0.6870	0.8780	0.7507	0.9490	**0.5810**	0.7423
Social Science	0.4047	0.4773	0.4040	0.9223	0.5580	0.9933	**0.3270**	0.4467
Society Culture	0.5927	0.6803	0.5220	0.9267	0.4553	0.9403	**0.4357**	0.4870
Average	0.4776	0.5704	0.4732	0.6103	0.5323	0.8181	**0.3930**	0.4578
Rank	4	6	3	7	5	8	1	2

TABLE 6.51: Performance of MLFLD (TrTe) for Coverage (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	3.0050	3.1634	3.2030	2.6089	1.9158	2.0644	1.9356	**1.8119**
Scene	1.2834	1.1982	1.3035	0.9013	0.6873	0.9724	0.5661	**0.5184**
Image	0.7333	1.0250	0.9550	0.8583	0.5100	1.4883	0.6083	**0.5000**
Yeast	9.8244	9.8571	9.2072	10.5125	6.7764	6.7481	**6.4318**	6.5540
Arts Humanity	9.0557	12.3843	8.5843	12.6653	8.8693	12.3893	**5.4313**	5.9870
Business Eco.	5.6803	12.0833	6.1823	12.5133	4.0303	6.3847	**2.1840**	2.4683
Education	9.4910	20.0320	8.8017	18.0113	7.4220	13.1420	**3.4973**	4.5247
Entertainment	6.0390	10.9297	6.6727	10.8017	6.7330	9.2197	**3.1467**	4.3117
Health	7.2900	16.6443	7.1783	24.8783	13.1273	9.5870	**3.3043**	4.0317
Reference	6.7697	15.7433	6.6327	14.6760	9.5627	9.8253	**3.5420**	5.0580
Science	12.1370	21.2330	13.0420	23.5560	12.8283	20.5630	**6.0470**	7.6283
Social Science	7.4227	15.3023	7.1950	25.7307	6.5350	16.7437	**3.0340**	3.9590
Society Culture	9.7363	13.1083	8.1703	14.4827	7.9757	13.9627	**5.3653**	5.9630
Average	6.8052	11.7465	6.7022	13.2459	6.6902	9.4685	**3.4687**	4.1012
Rank	5	7	4	8	3	6	1	2

Table 6.52: Performance of MLFLD (TrTe) for Average Precision (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.6540	0.6082	0.6270	0.6946	0.7916	0.7664	0.7810	**0.8024**
Scene	0.7143	0.7247	0.7312	0.7751	0.8154	0.6810	0.8511	**0.8653**
Image	0.8377	0.7342	0.7712	0.7862	0.8692	0.5899	0.8456	**0.8718**
Yeast	0.5859	0.5399	0.6150	0.5836	0.7440	0.7155	**0.7505**	0.7396
Arts Humanity	0.4635	0.3603	0.4780	0.2937	0.3250	0.1441	**0.5097**	0.4459
Business Eco.	0.7596	0.6165	0.7760	0.6826	0.8606	0.2442	**0.8798**	0.8657
Education	0.4848	0.2620	0.4856	0.2949	0.5086	0.1125	**0.5993**	0.4806
Entertainment	0.5327	0.4024	0.5301	0.3951	0.4263	0.1493	**0.6013**	0.4652
Health	0.6502	0.4639	0.6436	0.1905	0.3126	0.1993	**0.6817**	0.6055
Reference	0.5816	0.4243	0.5720	0.4224	0.2899	0.1514	**0.6194**	0.5445
Science	0.4203	0.2471	0.4142	0.1600	0.3647	0.0933	**0.5324**	0.4019
Social Science	0.6641	0.5089	0.6652	0.1071	0.5584	0.0860	**0.7481**	0.6581
Society Culture	0.4746	0.3386	0.5274	0.1513	0.5645	0.1451	**0.6128**	0.5651
Average	0.6018	0.4793	0.6028	0.4259	0.5716	0.3137	**0.6933**	0.6394
Rank	4	6	3	7	5	8	1	2

Table 6.53: Performance of MLFLD (TrTe) for Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.3173	0.3774	0.3859	0.3672	0.4612	0.4905	0.4818	**0.5136**
Scene	0.5173	0.5787	0.5975	0.5596	0.5439	0.3871	0.6597	**0.6749**
Image	0.6308	0.5794	0.6165	0.5444	0.6294	0.2992	0.6492	**0.7008**
Yeast	0.3965	0.3714	0.4120	0.3296	0.4857	0.4976	**0.4998**	0.4802
Arts Humanity	0.2332	0.2579	**0.2895**	0.1095	0.0564	0.0651	0.0331	0.0262
Business Eco.	0.6292	0.6176	0.6310	0.6412	0.6811	0.0827	**0.6967**	0.6813
Education	0.2561	0.2430	**0.2987**	0.1723	0.1397	0.0592	0.1560	0.0433
Entertainment	0.3105	**0.3787**	0.3370	0.2870	0.2012	0.0836	0.1862	0.1340
Health	0.4495	0.4725	**0.4828**	0.1362	0.1088	0.0629	0.3390	0.3533
Reference	0.3968	**0.4089**	0.3979	0.3259	0.0397	0.0578	0.1032	0.0358
Science	0.2122	0.2127	**0.2553**	0.0897	0.0397	0.0364	0.0695	0.0120
Social Science	0.4924	0.4974	**0.5012**	0.0560	0.1718	0.0000	0.2996	0.3686
Society Culture	0.2894	0.2888	**0.3690**	0.0235	0.2313	0.0402	0.2431	0.1770
Average	0.3947	0.4065	**0.4288**	0.2802	0.2915	0.1663	0.3398	0.3232
Rank	3	2	1	7	6	8	4	5

Table 6.54: Performance of MLFLD (TrTe) for Subset Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.1238	0.1485	0.1485	0.1089	0.2129	0.2129	0.2178	**0.2574**
Scene	0.4080	0.5401	0.5376	0.4724	0.5167	0.0962	0.6012	**0.6279**
Image	0.5150	0.5067	0.5533	0.4750	0.5900	0.2550	0.5983	**0.6350**
Yeast	0.0371	0.0687	0.1047	0.0153	**0.1810**	0.1069	0.1647	**0.1810**
Arts Humanity	0.1380	0.1867	**0.2040**	0.0703	0.0457	0.0000	0.0277	0.0223
Business Eco.	0.4420	0.4407	0.4543	0.4830	0.5140	0.0000	0.5353	**0.5357**
Education	0.1577	0.1737	**0.2083**	0.1160	0.1180	0.0000	0.1310	0.0293
Entertainment	0.2130	**0.3153**	0.2490	0.2150	0.1797	0.0000	0.1687	0.1157
Health	0.2997	**0.3580**	0.3560	0.0690	0.0327	0.0000	0.2403	0.2517
Reference	0.3250	**0.3590**	0.3317	0.2943	0.0360	0.0000	0.0963	0.0313
Science	0.1437	0.1663	**0.1880**	0.0637	0.0357	0.0000	0.0603	0.0110
Social Science	0.3983	**0.4403**	0.4303	0.0470	0.1597	0.0000	0.2700	0.3313
Society Culture	0.1763	0.1927	**0.2563**	0.0077	0.1917	0.0000	0.2010	0.1450
Average	0.2598	0.2997	**0.3094**	0.1875	0.2164	0.0516	0.2548	0.2442
Rank	3	2	1	7	6	8	4	5

Table 6.55: Performance of MLFLD (TrTe) for Ex-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.3936	0.4589	0.4749	0.4630	0.5416	0.5795	0.5662	**0.5954**
Scene	0.5551	0.5917	0.6177	0.5893	0.5530	0.5029	0.6793	**0.6906**
Image	0.6713	0.6055	0.6378	0.5683	0.6428	0.3139	0.6667	**0.7233**
Yeast	0.5239	0.4845	0.5244	0.4510	0.5868	**0.6148**	0.6067	0.5805
Arts Humanity	0.2707	0.2869	**0.3223**	0.1244	0.0608	0.1198	0.0352	0.0277
Business Eco.	0.6951	0.6817	0.6932	0.7012	0.7407	0.1477	**0.7546**	0.7357
Education	0.2924	0.2699	**0.3316**	0.1933	0.1472	0.1093	0.1647	0.0481
Entertainment	0.3475	**0.4029**	0.3699	0.3143	0.2096	0.1491	0.1924	0.1398
Health	0.5035	0.5157	**0.5289**	0.1609	0.1453	0.1128	0.3772	0.3923
Reference	0.4224	**0.4267**	0.4214	0.3368	0.0410	0.1072	0.1055	0.0372
Science	0.2386	0.2305	**0.2807**	0.0998	0.0411	0.0691	0.0728	0.0124
Social Science	0.5262	0.5186	**0.5268**	0.0594	0.1761	0.0000	0.3100	0.3819
Society Culture	0.3343	0.3281	**0.4140**	0.0300	0.2466	0.0732	0.2594	0.1896
Average	0.4442	0.4463	**0.4726**	0.3147	0.3179	0.2230	0.3685	0.3503
Rank	3	2	1	7	6	8	4	5

TABLE 6.56: Performance of MLFLD (TrTe) for Macro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.4294	0.4563	0.4680	0.5063	0.5909	0.6090	0.5880	**0.6275**
Scene	0.6209	0.5938	0.6280	0.6388	0.6285	0.5697	0.7156	**0.7400**
Image	0.4930	0.4078	0.4721	0.4665	0.5666	0.2211	0.5904	**0.6104**
Yeast	0.3645	0.3498	0.3832	0.2482	0.3605	**0.4274**	0.3444	0.3887
Arts Humanity	0.1845	0.1358	**0.1853**	0.0706	0.0208	0.1044	0.0343	0.0176
Business Eco.	**0.2263**	0.1448	0.2185	0.1575	0.1281	0.1365	0.1817	NaN
Education	**0.1855**	0.1348	0.1842	0.0574	0.1400	0.1335	0.1421	NaN
Entertainment	**0.2241**	0.2139	0.2240	0.1635	0.0649	0.1130	0.1271	0.1031
Health	0.2955	0.2567	**0.3007**	0.1790	0.1077	0.1404	0.1567	NaN
Reference	**0.1978**	0.1695	0.1942	0.1085	0.0673	0.1185	0.0907	NaN
Science	0.1407	0.0897	**0.1513**	0.0538	0.0179	0.0633	0.0408	0.0072
Social Science	**0.2227**	0.1526	0.2035	0.0950	0.0890	0.0513	0.1175	NaN
Society Culture	**0.1327**	0.1099	0.1317	0.0513	0.0673	0.0949	0.0714	0.0343
Average	0.2860	0.2473	0.2881	0.2151	0.2192	0.2141	0.2462	**0.3161**
Rank	3	4	2	7	6	8	5	1

TABLE 6.57: Performance of MLFLD (TrTe) for Micro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Emotions	0.4356	0.4835	0.4911	0.5119	0.6104	0.6276	0.6278	**0.6472**
Scene	0.6132	0.5870	0.6185	0.6290	0.6380	0.5405	0.7156	**0.7387**
Image	0.6941	0.5459	0.6273	0.5856	0.7048	0.3361	0.7166	**0.7412**
Yeast	0.5461	0.5141	0.5414	0.4773	0.6193	0.6291	**0.6303**	0.6144
Arts Humanity	0.3131	0.2693	**0.3297**	0.1722	0.0580	0.1249	0.0480	0.0373
Business Eco.	0.6639	0.6244	0.6614	0.6486	0.6895	0.1736	**0.6990**	0.6739
Education	**0.3506**	0.2753	0.3504	0.2384	0.2336	0.1253	0.2541	0.0934
Entertainment	0.4023	0.3815	**0.4041**	0.3654	0.2020	0.1683	0.2696	0.1929
Health	0.5343	0.4909	**0.5353**	0.2374	0.1725	0.1604	0.4033	0.3834
Reference	**0.4780**	0.4057	0.4752	0.4102	0.0408	0.1267	0.1652	0.0645
Science	0.2876	0.2154	**0.2923**	0.1396	0.0599	0.0815	0.1063	0.0183
Social Science	**0.5424**	0.4693	0.5399	0.0943	0.2487	0.0000	0.3865	0.4244
Society Culture	0.3482	0.2975	**0.3821**	0.0521	0.2825	0.1235	0.2896	0.2207
Average	0.4776	0.4277	**0.4807**	0.3509	0.3508	0.2475	0.4086	0.3731
Rank	2	3	1	6	7	8	4	5

Table 6.58 provides summarized performance of MLFLD using Euclidean and Hamming distance on train-test datasets.

Table 6.58: Summary of MLFLD performance (TrTe) using Hamming distance

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
HamLoss	0.0999	0.1183	0.1058	0.1024	0.0898	0.4077	**0.0769**	0.0804
RankLoss	0.2259	0.3938	0.2298	0.4146	0.1962	0.3214	**0.1081**	0.1242
OneError	0.4776	0.5704	0.4732	0.6103	0.5323	0.8181	**0.3930**	0.4578
Coverage	6.8052	11.7465	6.7022	13.2459	6.6902	9.4685	**3.4687**	4.1012
AvgPrec	0.6018	0.4793	0.6028	0.4259	0.5716	0.3137	**0.6933**	0.6394
Accuracy	0.3947	0.4065	**0.4288**	0.2802	0.2915	0.1663	0.3398	0.3232
SubAcc	0.2598	0.2997	**0.3094**	0.1875	0.2164	0.0516	0.2548	0.2442
Ex-F1	0.4442	0.4463	**0.4726**	0.3147	0.3179	0.2230	0.3685	0.3503
Macro-F1	0.2860	0.2473	0.2881	0.2151	0.2192	0.2141	0.2462	**0.3161**
Micro-F1	0.4776	0.4277	**0.4807**	0.3509	0.3508	0.2475	0.4086	0.3731
Avg rank	3.5	4.6	2.7	6.9	5.0	7.6	**2.6**	3.1
#Wins	0	0	4	0	0	0	**5**	1

MLFLD performed with rank 2 for the first five metrics. For subset accuracy, MLFLD has shown improvement over five datasets, whereas for remaining parameters, it improved for 2-3 datasets only. It proves that our algorithm is better in the prediction of all labels of an example. MLkNN and CC both algorithms are performing very similarly, followed by the proposed algorithm MLFLD. It is slightly less in terms of average rank over ten metrics but seems unfortunate in terms of wins.

All neighbor based algorithms, namely BRkNN, MLkNN, and MLFLD, could not perform well on accuracy and F measure based metrics.

6.3.2 Performance of MLFLD-MAXP algorithm using train and test splits of datasets using Hamming distance

In this section, the performance of MLFLD-MAXP is observed for train-test splits (TrTe) using Euclidean distance for feature similarity shown in Table 6.59 to 6.68.

TABLE 6.59: Performance of MLFLD-MAXP (TrTe) for Hamming Loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.3144	0.3226	0.3317	0.3053	0.2170	0.2467	**0.2162**	0.2211
Scene	0.1364	0.1469	0.1377	0.1307	0.1080	0.2395	0.0962	**0.0886**
Image	0.1390	0.2323	0.1683	0.1840	0.1153	0.2713	**0.1147**	0.1160
Yeast	0.2766	0.2977	0.2898	0.2757	0.2029	0.2422	**0.2008**	0.2072
Arts Humanity	0.0703	0.0891	0.0737	0.0677	0.0912	0.7743	**0.0612**	0.0810
Business Eco.	0.0332	0.0383	0.0337	0.0309	0.0285	0.4181	**0.0269**	0.0285
Education	0.0494	0.0633	0.0530	0.0481	0.0406	0.5215	**0.0387**	0.0558
Entertainment	0.0692	0.0817	0.0713	0.0681	0.0887	0.5909	**0.0604**	0.0847
Health	**0.0425**	0.0512	0.0439	0.0502	0.0936	0.3693	0.0458	0.0519
Reference	0.0320	0.0416	0.0320	**0.0314**	0.0622	0.4103	**0.0314**	0.0371
Science	0.0403	0.0554	0.0454	0.0387	0.0351	0.6759	**0.0325**	0.0478
Social Science	0.0268	0.0340	0.0265	0.0335	0.0290	0.0331	**0.0218**	0.0309
Society Culture	0.0682	0.0844	0.0678	0.0669	0.0555	0.5076	**0.0537**	0.0614
Average	0.0999	0.1183	0.1058	0.1024	0.0898	0.4077	**0.0769**	0.0855
Rank	4	7	6	5	3	8	1	2

TABLE 6.60: Performance of MLFLD-MAXP (TrTe) for Ranking Loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.3650	0.4050	0.4086	0.2951	0.1694	0.1952	0.1781	**0.1570**
Scene	0.2315	0.2171	0.2350	0.1591	0.1173	0.1740	0.0930	**0.0830**
Image	0.1382	0.2240	0.1999	0.1769	0.0924	0.3337	0.1154	**0.0888**
Yeast	0.3551	0.4311	0.3397	0.3888	0.1902	0.2011	**0.1766**	0.1839
Arts Humanity	0.2645	0.3958	0.2481	0.4067	0.2670	0.4292	**0.1514**	0.1707
Business Eco.	0.1150	0.2946	0.1239	0.2689	0.0729	0.1635	**0.0373**	0.0454
Education	0.2270	0.5558	0.2138	0.4859	0.1744	0.3746	**0.0800**	0.1112
Entertainment	0.2353	0.4822	0.2650	0.4707	0.2755	0.4254	**0.1151**	0.1735
Health	0.1502	0.4289	0.1484	0.6860	0.3145	0.2459	**0.0605**	0.0788
Reference	0.1831	0.4526	0.1787	0.4217	0.2656	0.2894	**0.0919**	0.1367
Science	0.2485	0.4828	0.2653	0.5390	0.2719	0.4789	**0.1167**	0.1551
Social Science	0.1511	0.3441	0.1440	0.6310	0.1299	0.4045	**0.0561**	0.0767
Society Culture	0.2720	0.4048	0.2168	0.4602	0.2093	0.4622	**0.1338**	0.1543
Average	0.2259	0.3938	0.2298	0.4146	0.1962	0.3214	**0.1081**	0.1242
Rank	4	7	5	8	3	6	1	2

Table 6.61: Performance of MLFLD-MAXP (TrTe) for One Error (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.4356	0.5396	0.4901	0.4059	0.3069	0.3267	0.3218	**0.2970**
Scene	0.4189	0.4047	0.3687	0.3470	0.3010	0.5510	0.2425	**0.2191**
Image	0.2417	0.4100	0.3383	0.3350	0.2267	0.6483	0.2517	**0.2183**
Yeast	0.3915	0.5703	0.3479	0.3631	0.2595	0.3217	**0.2519**	0.2835
Arts Humanity	0.6413	0.7153	**0.6243**	0.7960	0.9043	0.9817	0.6330	0.7323
Business Eco.	0.2653	0.3443	0.2270	0.1843	0.1273	0.9877	**0.1213**	0.1343
Education	0.6317	0.7647	0.6313	0.7340	0.5983	0.9957	**0.5207**	0.6710
Entertainment	0.5887	0.6213	0.5713	0.6387	0.7487	0.9640	**0.5300**	0.6897
Health	**0.4027**	0.5200	0.4167	0.8090	0.7307	0.9937	0.4190	0.5070
Reference	0.5110	0.5823	0.5230	0.5937	0.9520	0.9823	**0.4730**	0.5227
Science	0.6827	0.7847	0.6870	0.8780	0.7507	0.9490	**0.5810**	0.7423
Social Science	0.4047	0.4773	0.4040	0.9223	0.5580	0.9933	**0.3270**	0.4467
Society Culture	0.5927	0.6803	0.5220	0.9267	0.4553	0.9403	**0.4357**	0.4870
Average	0.4776	0.5704	0.4732	0.6103	0.5323	0.8181	**0.3930**	0.4578
Rank	4	6	3	7	5	8	1	2

Table 6.62: Performance of MLFLD-MAXP (TrTe) for Coverage (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	3.0050	3.1634	3.2030	2.6089	1.9158	2.0644	1.9356	**1.8119**
Scene	1.2834	1.1982	1.3035	0.9013	0.6873	0.9724	0.5661	**0.5184**
Image	0.7333	1.0250	0.9550	0.8583	0.5100	1.4883	0.6083	**0.5000**
Yeast	9.8244	9.8571	9.2072	10.5125	6.7764	6.7481	**6.4318**	6.5540
Arts Humanity	9.0557	12.3843	8.5843	12.6653	8.8693	12.3893	**5.4313**	5.9870
Business Eco.	5.6803	12.0833	6.1823	12.5133	4.0303	6.3847	**2.1840**	2.4683
Education	9.4910	20.0320	8.8017	18.0113	7.4220	13.1420	**3.4973**	4.5247
Entertainment	6.0390	10.9297	6.6727	10.8017	6.7330	9.2197	**3.1467**	4.3117
Health	7.2900	16.6443	7.1783	24.8783	13.1273	9.5870	**3.3043**	4.0317
Reference	6.7697	15.7433	6.6327	14.6760	9.5627	9.8253	**3.5420**	5.0580
Science	12.1370	21.2330	13.0420	23.5560	12.8283	20.5630	**6.0470**	7.6283
Social Science	7.4227	15.3023	7.1950	25.7307	6.5350	16.7437	**3.0340**	3.9590
Society Culture	9.7363	13.1083	8.1703	14.4827	7.9757	13.9627	**5.3653**	5.9630
Average	6.8052	11.7465	6.7022	13.2459	6.6902	9.4685	**3.4687**	4.1012
Rank	5	7	4	8	3	6	1	2

Table 6.63: Performance of MLFLD-MAXP (TrTe) for Average Precision (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.6540	0.6082	0.6270	0.6946	0.7916	0.7664	0.7810	**0.8024**
Scene	0.7143	0.7247	0.7312	0.7751	0.8154	0.6810	0.8511	**0.8653**
Image	0.8377	0.7342	0.7712	0.7862	0.8692	0.5899	0.8456	**0.8718**
Yeast	0.5859	0.5399	0.6150	0.5836	0.7440	0.7155	**0.7505**	0.7396
Arts Humanity	0.4635	0.3603	0.4780	0.2937	0.3250	0.1441	**0.5097**	0.4459
Business Eco.	0.7596	0.6165	0.7760	0.6826	0.8606	0.2442	**0.8798**	0.8657
Education	0.4848	0.2620	0.4856	0.2949	0.5086	0.1125	**0.5993**	0.4806
Entertainment	0.5327	0.4024	0.5301	0.3951	0.4263	0.1493	**0.6013**	0.4652
Health	0.6502	0.4639	0.6436	0.1905	0.3126	0.1993	**0.6817**	0.6055
Reference	0.5816	0.4243	0.5720	0.4224	0.2899	0.1514	**0.6194**	0.5445
Science	0.4203	0.2471	0.4142	0.1600	0.3647	0.0933	**0.5324**	0.4019
Social Science	0.6641	0.5089	0.6652	0.1071	0.5584	0.0860	**0.7481**	0.6581
Society Culture	0.4746	0.3386	0.5274	0.1513	0.5645	0.1451	**0.6128**	0.5651
Average	0.6018	0.4793	0.6028	0.4259	0.5716	0.3137	**0.6933**	0.6394
Rank	4	6	3	7	5	8	1	2

Table 6.64: Performance of MLFLD-MAXP (TrTe) for Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.3173	0.3774	0.3859	0.3672	0.4612	0.4905	0.4818	**0.5202**
Scene	0.5173	0.5787	0.5975	0.5596	0.5439	0.3871	0.6597	**0.7389**
Image	0.6308	0.5794	0.6165	0.5444	0.6294	0.2992	0.6492	**0.7292**
Yeast	0.3965	0.3714	0.4120	0.3296	0.4857	0.4976	**0.4998**	0.4821
Arts Humanity	0.2332	0.2579	**0.2895**	0.1095	0.0564	0.0651	0.0331	0.2058
Business Eco.	0.6292	0.6176	0.6310	0.6412	0.6811	0.0827	**0.6967**	0.6825
Education	0.2561	0.2430	**0.2987**	0.1723	0.1397	0.0592	0.1560	0.2636
Entertainment	0.3105	**0.3787**	0.3370	0.2870	0.2012	0.0836	0.1862	0.2572
Health	0.4495	0.4725	**0.4828**	0.1362	0.1088	0.0629	0.3390	0.3714
Reference	0.3968	0.4089	0.3979	0.3259	0.0397	0.0578	0.1032	**0.4280**
Science	0.2122	0.2127	**0.2553**	0.0897	0.0397	0.0364	0.0695	0.2141
Social Science	0.4924	0.4974	**0.5012**	0.0560	0.1718	0.0000	0.2996	0.4938
Society Culture	0.2894	0.2888	0.3690	0.0235	0.2313	0.0402	0.2431	**0.3833**
Average	0.3947	0.4065	0.4288	0.2802	0.2915	0.1663	0.3398	**0.4439**
Rank	4	3	2	7	6	8	5	1

Table 6.65: Performance of MLFLD-MAXP (TrTe) for Subset Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.1238	0.1485	0.1485	0.1089	0.2129	0.2129	0.2178	**0.2574**
Scene	0.4080	0.5401	0.5376	0.4724	0.5167	0.0962	0.6012	**0.6890**
Image	0.5150	0.5067	0.5533	0.4750	0.5900	0.2550	0.5983	**0.6617**
Yeast	0.0371	0.0687	0.1047	0.0153	**0.1810**	0.1069	0.1647	**0.1810**
Arts Humanity	0.1380	0.1867	**0.2040**	0.0703	0.0457	0.0000	0.0277	0.1643
Business Eco.	0.4420	0.4407	0.4543	0.4830	0.5140	0.0000	0.5353	**0.5363**
Education	0.1577	0.1737	0.2083	0.1160	0.1180	0.0000	0.1310	**0.2117**
Entertainment	0.2130	**0.3153**	0.2490	0.2150	0.1797	0.0000	0.1687	0.2097
Health	0.2997	**0.3580**	0.3560	0.0690	0.0327	0.0000	0.2403	0.2637
Reference	0.3250	0.3590	0.3317	0.2943	0.0360	0.0000	0.0963	**0.3820**
Science	0.1437	0.1663	**0.1880**	0.0637	0.0357	0.0000	0.0603	0.1827
Social Science	0.3983	**0.4403**	0.4303	0.0470	0.1597	0.0000	0.2700	0.4380
Society Culture	0.1763	0.1927	0.2563	0.0077	0.1917	0.0000	0.2010	**0.2947**
Average	0.2598	0.2997	0.3094	0.1875	0.2164	0.0516	0.2548	**0.3440**
Rank	4	3	2	7	6	8	5	1

Table 6.66: Performance of MLFLD-MAXP (TrTe) for Ex-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.3936	0.4589	0.4749	0.4630	0.5416	0.5795	0.5662	**0.6045**
Scene	0.5551	0.5917	0.6177	0.5893	0.5530	0.5029	0.6793	**0.7556**
Image	0.6713	0.6055	0.6378	0.5683	0.6428	0.3139	0.6667	**0.7522**
Yeast	0.5239	0.4845	0.5244	0.4510	0.5868	**0.6148**	0.6067	0.5835
Arts Humanity	0.2707	0.2869	**0.3223**	0.1244	0.0608	0.1198	0.0352	0.2226
Business Eco.	0.6951	0.6817	0.6932	0.7012	0.7407	0.1477	**0.7546**	0.7370
Education	0.2924	0.2699	**0.3316**	0.1933	0.1472	0.1093	0.1647	0.2828
Entertainment	0.3475	**0.4029**	0.3699	0.3143	0.2096	0.1491	0.1924	0.2738
Health	0.5035	0.5157	**0.5289**	0.1609	0.1453	0.1128	0.3772	0.4128
Reference	0.4224	0.4267	0.4214	0.3368	0.0410	0.1072	0.1055	**0.4439**
Science	0.2386	0.2305	**0.2807**	0.0998	0.0411	0.0691	0.0728	0.2264
Social Science	0.5262	0.5186	**0.5268**	0.0594	0.1761	0.0000	0.3100	0.5139
Society Culture	0.3343	0.3281	0.4140	0.0300	0.2466	0.0732	0.2594	**0.4184**
Average	0.4442	0.4463	0.4726	0.3147	0.3179	0.2230	0.3685	**0.4790**
Rank	4	3	2	7	6	8	5	1

Table 6.67: Performance of MLFLD-MAXP (TrTe) for Macro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.4294	0.4563	0.4680	0.5063	0.5909	0.6090	0.5880	**0.6273**
Scene	0.6209	0.5938	0.6280	0.6388	0.6285	0.5697	0.7156	**0.7569**
Image	0.4930	0.4078	0.4721	0.4665	0.5666	0.2211	0.5904	**0.6202**
Yeast	0.3645	0.3498	0.3832	0.2482	0.3605	**0.4274**	0.3444	0.3891
Arts Humanity	0.1845	0.1358	**0.1853**	0.0706	0.0208	0.1044	0.0343	0.0583
Business Eco.	**0.2263**	0.1448	0.2185	0.1575	0.1281	0.1365	0.1817	NaN
Education	**0.1855**	0.1348	0.1842	0.0574	0.1400	0.1335	0.1421	NaN
Entertainment	**0.2241**	0.2139	0.2240	0.1635	0.0649	0.1130	0.1271	0.1370
Health	0.2955	0.2567	**0.3007**	0.1790	0.1077	0.1404	0.1567	NaN
Reference	**0.1978**	0.1695	0.1942	0.1085	0.0673	0.1185	0.0907	NaN
Science	0.1407	0.0897	**0.1513**	0.0538	0.0179	0.0633	0.0408	0.0407
Social Science	**0.2227**	0.1526	0.2035	0.0950	0.0890	0.0513	0.1175	NaN
Society Culture	**0.1327**	0.1099	0.1317	0.0513	0.0673	0.0949	0.0714	0.0608
Average	0.2860	0.2473	0.2881	0.2151	0.2192	0.2141	0.2462	**0.3363**
Rank	3	4	2	7	6	8	5	1

Table 6.68: Performance of MLFLD-MAXP (TrTe) for Micro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Emotions	0.4356	0.4835	0.4911	0.5119	0.6104	0.6276	0.6278	**0.6483**
Scene	0.6132	0.5870	0.6185	0.6290	0.6380	0.5405	0.7156	**0.7502**
Image	0.6941	0.5459	0.6273	0.5856	0.7048	0.3361	0.7166	**0.7426**
Yeast	0.5461	0.5141	0.5414	0.4773	0.6193	0.6291	**0.6303**	0.6152
Arts Humanity	0.3131	0.2693	**0.3297**	0.1722	0.0580	0.1249	0.0480	0.2026
Business Eco.	0.6639	0.6244	0.6614	0.6486	0.6895	0.1736	**0.6990**	0.6747
Education	**0.3506**	0.2753	0.3504	0.2384	0.2336	0.1253	0.2541	0.2752
Entertainment	0.4023	0.3815	**0.4041**	0.3654	0.2020	0.1683	0.2696	0.2780
Health	0.5343	0.4909	**0.5353**	0.2374	0.1725	0.1604	0.4033	0.3921
Reference	**0.4780**	0.4057	0.4752	0.4102	0.0408	0.1267	0.1652	0.4388
Science	0.2876	0.2154	**0.2923**	0.1396	0.0599	0.0815	0.1063	0.2124
Social Science	**0.5424**	0.4693	0.5399	0.0943	0.2487	0.0000	0.3865	0.4822
Society Culture	0.3482	0.2975	0.3821	0.0521	0.2825	0.1235	0.2896	**0.3826**
Average	0.4776	0.4277	**0.4807**	0.3509	0.3508	0.2475	0.4086	0.4688
Rank	2	4	1	6	7	8	5	3

TABLE 6.69: Summary of MLFLD-MAXP (TrTe) performance using Hamming distance

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
HamLoss	0.0999	0.1183	0.1058	0.1024	0.0898	0.4077	**0.0769**	0.0855
RankLoss	0.2259	0.3938	0.2298	0.4146	0.1962	0.3214	**0.1081**	0.1242
OneError	0.4776	0.5704	0.4732	0.6103	0.5323	0.8181	**0.3930**	0.4578
Coverage	6.8052	11.7465	6.7022	13.2459	6.6902	9.4685	**3.4687**	4.1012
AvgPrec	0.6018	0.4793	0.6028	0.4259	0.5716	0.3137	**0.6933**	0.6394
Accuracy	0.3947	0.4065	0.4288	0.2802	0.2915	0.1663	0.3398	**0.4439**
SubAcc	0.2598	0.2997	0.3094	0.1875	0.2164	0.0516	0.2548	**0.3440**
Ex-F1	0.4442	0.4463	0.4726	0.3147	0.3179	0.2230	0.3685	**0.4790**
Macro-F1	0.2860	0.2473	0.2881	0.2151	0.2192	0.2141	0.2462	**0.3363**
Micro-F1	0.4776	0.4277	**0.4807**	0.3509	0.3508	0.2475	0.4086	0.4688
Avg Rank	3.8	5.0	3.0	6.9	5.0	7.6	3.0	**1.7**
#Wins	0	0	1	0	0	0	5	**4**

Observations: From Table 6.69, MLFLD-MAXP performance is improved in terms of average rank over all the measures though it shows only 4 wins among all the metrics. Though MLkNN shows 5 wins, its avg rank is almost twice than of MLFLD-MAXP. To summarize,

- Subset accuracy is the most improved metric by MLFLD-MAXP. It is increased for eight datasets among 13 with 11% and 35% rise w.r.t. CC and MLkNN resp.
- MLFLD-MAXP has topped for four metrics, whereas MLkNN has topped for five metrics.
- MLFLD-MAXP and MLkNN achieved rank 1 and 5 respectively, for two accuracy measures, Ex-F1 and macro-F1. They worked at positions 2 and 1, respectively, for parameters, namely, ham and rank loss, avg precision, coverage, and one err.
- Accuracy and ex-F1 have been raised by 30% w.r.t. MLkNN.
- MLFLD-MAXP got rank 3 for micro F1 for which MLkNN got position 5. . It raised micro-F by 14% over MLkNN. Macro-F could not be measured for five datasets.
- MLFLD-MAXP performed with rank two among eight algorithms though it is able to improve ham loss for Scene only. The difference between ham loss computed by

the proposed algorithm and MLkNN is minimal except for Arts, Education, and Entertainment.

- For Emotions, Scene, and Image, MLFLD-MAXP has improved one err, rank loss, coverage and avg precision. It stood second among all for rank.

6.3.3 Comparison of MLFLD and MLFLD-MAXP

Performance comparison of proposed algorithms is carried out in this section for Euclidean and Hamming distance shown in Table 6.70 to 6.79.

TABLE 6.70: Performance of MLFLD and MLFLD-MAXP (TrTe) for Hamming Loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3144	0.3226	0.3317	0.3053	0.2170	0.2467	**0.2162**	0.2195	0.2211
Scene	0.1364	0.1469	0.1377	0.1307	0.1080	0.2395	0.0962	**0.0863**	0.0886
Image	0.1390	0.2323	0.1683	0.1840	0.1153	0.2713	0.1147	**0.1127**	0.1160
Yeast	0.2766	0.2977	0.2898	0.2757	0.2029	0.2422	**0.2008**	0.2072	0.2072
Arts Humanity	0.0703	0.0891	0.0737	0.0677	0.0912	0.7743	**0.0612**	0.0628	0.0810
Business Eco.	0.0332	0.0383	0.0337	0.0309	0.0285	0.4181	**0.0269**	0.0285	0.0285
Education	0.0494	0.0633	0.0530	0.0481	0.0406	0.5215	**0.0387**	0.0465	0.0558
Entertainment	0.0692	0.0817	0.0713	0.0681	0.0887	0.5909	**0.0604**	0.0722	0.0847
Health	0.0425	0.0512	0.0439	0.0502	0.0936	0.3693	**0.0458**	0.0512	0.0519
Reference	0.0320	0.0416	0.0320	0.0314	0.0622	0.4103	**0.0314**	0.0354	0.0371
Science	0.0403	0.0554	0.0454	0.0387	0.0351	0.6759	**0.0325**	0.0358	0.0478
Social Science	0.0268	0.0340	0.0265	0.0335	0.0290	0.0331	**0.0218**	0.0287	0.0309
Society Culture	0.0682	0.0844	0.0678	0.0669	0.0555	0.5076	**0.0537**	0.0585	0.0614
Average	0.0999	0.1183	0.1058	0.1024	0.0898	0.4077	**0.0769**	0.0804	0.0855
Rank	5	8	7	6	4	9	1	2	3

Table 6.71: Performance of MLFLD and MLFLD-MAXP (TrTe) for Ranking Loss (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3650	0.4050	0.4086	0.2951	0.1694	0.1952	0.1781	**0.1570**	**0.1570**
Scene	0.2315	0.2171	0.2350	0.1591	0.1173	0.1740	0.0930	**0.0830**	**0.0830**
Image	0.1382	0.2240	0.1999	0.1769	0.0924	0.3337	0.1154	**0.0888**	**0.0888**
Yeast	0.3551	0.4311	0.3397	0.3888	0.1902	0.2011	**0.1766**	0.1839	0.1839
Arts Humanity	0.2645	0.3958	0.2481	0.4067	0.2670	0.4292	**0.1514**	0.1707	0.1707
Business Eco.	0.1150	0.2946	0.1239	0.2689	0.0729	0.1635	**0.0373**	0.0454	0.0454
Education	0.2270	0.5558	0.2138	0.4859	0.1744	0.3746	**0.0800**	0.1112	0.1112
Entertainment	0.2353	0.4822	0.2650	0.4707	0.2755	0.4254	**0.1151**	0.1735	0.1735
Health	0.1502	0.4289	0.1484	0.6860	0.3145	0.2459	**0.0605**	0.0788	0.0788
Reference	0.1831	0.4526	0.1787	0.4217	0.2656	0.2894	**0.0919**	0.1367	0.1367
Science	0.2485	0.4828	0.2653	0.5390	0.2719	0.4789	**0.1167**	0.1551	0.1551
Social Science	0.1511	0.3441	0.1440	0.6310	0.1299	0.4045	**0.0561**	0.0767	0.0767
Society Culture	0.2720	0.4048	0.2168	0.4602	0.2093	0.4622	**0.1338**	0.1543	0.1543
Average	0.2259	0.3938	0.2298	0.4146	0.1962	0.3214	**0.1081**	0.1242	0.1242
Rank	5	8	6	9	4	7	1	2	2

Table 6.72: Performance of MLFLD and MLFLD-MAXP (TrTe) for One Error (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.4356	0.5396	0.4901	0.4059	0.3069	0.3267	0.3218	**0.2970**	**0.2970**
Scene	0.4189	0.4047	0.3687	0.3470	0.3010	0.5510	0.2425	**0.2191**	**0.2191**
Image	0.2417	0.4100	0.3383	0.3350	0.2267	0.6483	0.2517	**0.2183**	**0.2183**
Yeast	0.3915	0.5703	0.3479	0.3631	0.2595	0.3217	**0.2519**	0.2835	0.2835
Arts Humanity	0.6413	0.7153	**0.6243**	0.7960	0.9043	0.9817	0.6330	0.7323	0.7323
Business Eco.	0.2653	0.3443	0.2270	0.1843	0.1273	0.9877	**0.1213**	0.1343	0.1343
Education	0.6317	0.7647	0.6313	0.7340	0.5983	0.9957	**0.5207**	0.6710	0.6710
Entertainment	0.5887	0.6213	0.5713	0.6387	0.7487	0.9640	**0.5300**	0.6897	0.6897
Health	**0.4027**	0.5200	0.4167	0.8090	0.7307	0.9937	0.4190	0.5070	0.5070
Reference	0.5110	0.5823	0.5230	0.5937	0.9520	0.9823	**0.4730**	0.5227	0.5227
Science	0.6827	0.7847	0.6870	0.8780	0.7507	0.9490	**0.5810**	0.7423	0.7423
Social Science	0.4047	0.4773	0.4040	0.9223	0.5580	0.9933	**0.3270**	0.4467	0.4467
Society Culture	0.5927	0.6803	0.5220	0.9267	0.4553	0.9403	**0.4357**	0.4870	0.4870
Average	0.4776	0.5704	0.4732	0.6103	0.5323	0.8181	**0.3930**	0.4578	0.4578
Rank	5	7	4	8	6	9	1	2	2

Table 6.73: Performance of MLFLD and MLFLD-MAXP (TrTe) for Coverage (↓) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	3.0050	3.1634	3.2030	2.6089	1.9158	2.0644	1.9356	**1.8119**	**1.8119**
Scene	1.2834	1.1982	1.3035	0.9013	0.6873	0.9724	0.5661	**0.5184**	**0.5184**
Image	0.7333	1.0250	0.9550	0.8583	0.5100	1.4883	0.6083	**0.5000**	**0.5000**
Yeast	9.8244	9.8571	9.2072	10.5125	6.7764	6.7481	**6.4318**	6.5540	6.5540
Arts Humanity	9.0557	12.3843	8.5843	12.6653	8.8693	12.3893	**5.4313**	5.9870	5.9870
Business Eco.	5.6803	12.0833	6.1823	12.5133	4.0303	6.3847	**2.1840**	2.4683	2.4683
Education	9.4910	20.0320	8.8017	18.0113	7.4220	13.1420	**3.4973**	4.5247	4.5247
Entertainment	6.0390	10.9297	6.6727	10.8017	6.7330	9.2197	**3.1467**	4.3117	4.3117
Health	7.2900	16.6443	7.1783	24.8783	13.1273	9.5870	**3.3043**	4.0317	4.0317
Reference	6.7697	15.7433	6.6327	14.6760	9.5627	9.8253	**3.5420**	5.0580	5.0580
Science	12.1370	21.2330	13.0420	23.5560	12.8283	20.5630	**6.0470**	7.6283	7.6283
Social Science	7.4227	15.3023	7.1950	25.7307	6.5350	16.7437	**3.0340**	3.9590	3.9590
Society Culture	9.7363	13.1083	8.1703	14.4827	7.9757	13.9627	**5.3653**	5.9630	5.9630
Average	6.8052	11.7465	6.7022	13.2459	6.6902	9.4685	**3.4687**	4.1012	4.1012
Rank	6	8	5	9	4	7	1	2	2

Table 6.74: Performance of MLFLD and MLFLD-MAXP (TrTe) for Average Precision (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.6540	0.6082	0.6270	0.6946	0.7916	0.7664	0.7810	**0.8024**	**0.8024**
Scene	0.7143	0.7247	0.7312	0.7751	0.8154	0.6810	0.8511	**0.8653**	**0.8653**
Image	0.8377	0.7342	0.7712	0.7862	0.8692	0.5899	0.8456	**0.8718**	**0.8718**
Yeast	0.5859	0.5399	0.6150	0.5836	0.7440	0.7155	**0.7505**	0.7396	0.7396
Arts Humanity	0.4635	0.3603	0.4780	0.2937	0.3250	0.1441	**0.5097**	0.4459	0.4459
Business Eco.	0.7596	0.6165	0.7760	0.6826	0.8606	0.2442	**0.8798**	0.8657	0.8657
Education	0.4848	0.2620	0.4856	0.2949	0.5086	0.1125	**0.5993**	0.4806	0.4806
Entertainment	0.5327	0.4024	0.5301	0.3951	0.4263	0.1493	**0.6013**	0.4652	0.4652
Health	0.6502	0.4639	0.6436	0.1905	0.3126	0.1993	**0.6817**	0.6055	0.6055
Reference	0.5816	0.4243	0.5720	0.4224	0.2899	0.1514	**0.6194**	0.5445	0.5445
Science	0.4203	0.2471	0.4142	0.1600	0.3647	0.0933	**0.5324**	0.4019	0.4019
Social Science	0.6641	0.5089	0.6652	0.1071	0.5584	0.0860	**0.7481**	0.6581	0.6581
Society Culture	0.4746	0.3386	0.5274	0.1513	0.5645	0.1451	**0.6128**	0.5651	0.5651
Average	0.6018	0.4793	0.6028	0.4259	0.5716	0.3137	**0.6933**	0.6394	0.6394
Rank	5	7	4	8	6	9	1	2	2

Table 6.75: Performance of MLFLD and MLFLD-MAXP (TrTe) for Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3173	0.3774	0.3859	0.3672	0.4612	0.4905	0.4818	0.5136	**0.5202**
Scene	0.5173	0.5787	0.5975	0.5596	0.5439	0.3871	0.6597	0.6749	**0.7389**
Image	0.6308	0.5794	0.6165	0.5444	0.6294	0.2992	0.6492	0.7008	**0.7292**
Yeast	0.3965	0.3714	0.4120	0.3296	0.4857	0.4976	**0.4998**	0.4802	0.4821
Arts Humanity	0.2332	0.2579	**0.2895**	0.1095	0.0564	0.0651	0.0331	0.0262	0.2058
Business Eco.	0.6292	0.6176	0.6310	0.6412	0.6811	0.0827	**0.6967**	0.6813	0.6825
Education	0.2561	0.2430	**0.2987**	0.1723	0.1397	0.0592	0.1560	0.0433	0.2636
Entertainment	0.3105	**0.3787**	0.3370	0.2870	0.2012	0.0836	0.1862	0.1340	0.2572
Health	0.4495	0.4725	**0.4828**	0.1362	0.1088	0.0629	0.3390	0.3533	0.3714
Reference	0.3968	0.4089	0.3979	0.3259	0.0397	0.0578	0.1032	0.0358	**0.4280**
Science	0.2122	0.2127	**0.2553**	0.0897	0.0397	0.0364	0.0695	0.0120	0.2141
Social Science	0.4924	0.4974	**0.5012**	0.0560	0.1718	0.0000	0.2996	0.3686	0.4938
Society Culture	0.2894	0.2888	0.3690	0.0235	0.2313	0.0402	0.2431	0.1770	**0.3833**
Average	0.3947	0.4065	0.4288	0.2802	0.2915	0.1663	0.3398	0.3232	**0.4439**
Rank	4	3	2	8	7	9	5	6	1

Table 6.76: Performance of MLFLD and MLFLD-MAXP (TrTe) for Subset Accuracy (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.1238	0.1485	0.1485	0.1089	0.2129	0.2129	0.2178	**0.2574**	**0.2574**
Scene	0.4080	0.5401	0.5376	0.4724	0.5167	0.0962	0.6012	0.6279	**0.6890**
Image	0.5150	0.5067	0.5533	0.4750	0.5900	0.2550	0.5983	0.6350	**0.6617**
Yeast	0.0371	0.0687	0.1047	0.0153	**0.1810**	0.1069	0.1647	**0.1810**	**0.1810**
Arts Humanity	0.1380	0.1867	**0.2040**	0.0703	0.0457	0.0000	0.0277	0.0223	0.1643
Business Eco.	0.4420	0.4407	0.4543	0.4830	0.5140	0.0000	0.5353	0.5357	**0.5363**
Education	0.1577	0.1737	0.2083	0.1160	0.1180	0.0000	0.1310	0.0293	**0.2117**
Entertainment	0.2130	**0.3153**	0.2490	0.2150	0.1797	0.0000	0.1687	0.1157	0.2097
Health	0.2997	**0.3580**	0.3560	0.0690	0.0327	0.0000	0.2403	0.2517	0.2637
Reference	0.3250	0.3590	0.3317	0.2943	0.0360	0.0000	0.0963	0.0313	**0.3820**
Science	0.1437	0.1663	**0.1880**	0.0637	0.0357	0.0000	0.0603	0.0110	0.1827
Social Science	0.3983	**0.4403**	0.4303	0.0470	0.1597	0.0000	0.2700	0.3313	0.4380
Society Culture	0.1763	0.1927	0.2563	0.0077	0.1917	0.0000	0.2010	0.1450	**0.2947**
Average	0.2598	0.2997	0.3094	0.1875	0.2164	0.0516	0.2548	0.2442	**0.3440**
Rank	4	3	2	8	7	9	5	6	1

TABLE 6.77: Performance of MLFLD and MLFLD-MAXP (TrTe) for Ex-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3936	0.4589	0.4749	0.4630	0.5416	0.5795	0.5662	0.5954	**0.6045**
Scene	0.5551	0.5917	0.6177	0.5893	0.5530	0.5029	0.6793	0.6906	**0.7556**
Image	0.6713	0.6055	0.6378	0.5683	0.6428	0.3139	0.6667	0.7233	**0.7522**
Yeast	0.5239	0.4845	0.5244	0.4510	0.5868	**0.6148**	0.6067	0.5805	0.5835
Arts Humanity	0.2707	0.2869	**0.3223**	0.1244	0.0608	0.1198	0.0352	0.0277	0.2226
Business Eco.	0.6951	0.6817	0.6932	0.7012	0.7407	0.1477	**0.7546**	0.7357	0.7370
Education	0.2924	0.2699	**0.3316**	0.1933	0.1472	0.1093	0.1647	0.0481	0.2828
Entertainment	0.3475	**0.4029**	0.3699	0.3143	0.2096	0.1491	0.1924	0.1398	0.2738
Health	0.5035	0.5157	**0.5289**	0.1609	0.1453	0.1128	0.3772	0.3923	0.4128
Reference	0.4224	0.4267	0.4214	0.3368	0.0410	0.1072	0.1055	0.0372	**0.4439**
Science	0.2386	0.2305	**0.2807**	0.0998	0.0411	0.0691	0.0728	0.0124	0.2264
Social Science	0.5262	0.5186	**0.5268**	0.0594	0.1761	0.0000	0.3100	0.3819	0.5139
Society Culture	0.3343	0.3281	0.4140	0.0300	0.2466	0.0732	0.2594	0.1896	**0.4184**
Average	0.4442	0.4463	0.4726	0.3147	0.3179	0.2230	0.3685	0.3503	**0.4790**
Rank	4	3	2	8	7	9	5	6	1

TABLE 6.78: Performance of MLFLD and MLFLD-MAXP (TrTe) for Macro-F1 (↑) using Hamming distance

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.4294	0.4563	0.4680	0.5063	0.5909	0.6090	0.5880	**0.6275**	0.6273
Scene	0.6209	0.5938	0.6280	0.6388	0.6285	0.5697	0.7156	0.7400	**0.7569**
Image	0.4930	0.4078	0.4721	0.4665	0.5666	0.2211	0.5904	0.6104	**0.6202**
Yeast	0.3645	0.3498	0.3832	0.2482	0.3605	**0.4274**	0.3444	0.3887	0.3891
Arts Humanity	0.1845	0.1358	**0.1853**	0.0706	0.0208	0.1044	0.0343	0.0176	0.0583
Business Eco.	**0.2263**	0.1448	0.2185	0.1575	0.1281	0.1365	0.1817	NaN	NaN
Education	**0.1855**	0.1348	0.1842	0.0574	0.1400	0.1335	0.1421	NaN	NaN
Entertainment	**0.2241**	0.2139	0.2240	0.1635	0.0649	0.1130	0.1271	0.1031	0.1370
Health	0.2955	0.2567	**0.3007**	0.1790	0.1077	0.1404	0.1567	NaN	NaN
Reference	**0.1978**	0.1695	0.1942	0.1085	0.0673	0.1185	0.0907	NaN	NaN
Science	0.1407	0.0897	**0.1513**	0.0538	0.0179	0.0633	0.0408	0.0072	0.0407
Social Science	**0.2227**	0.1526	0.2035	0.0950	0.0890	0.0513	0.1175	NaN	NaN
Society Culture	**0.1327**	0.1099	0.1317	0.0513	0.0673	0.0949	0.0714	0.0343	0.0608
Average	0.2860	0.2473	0.2881	0.2151	0.2192	0.2141	0.2462	0.3161	**0.3363**
Rank	4	5	3	8	7	9	6	2	1

TABLE 6.79: Performance of MLFLD and MLFLD-MAXP (TrTe) for Micro-F1 (↑) using Hamming distance

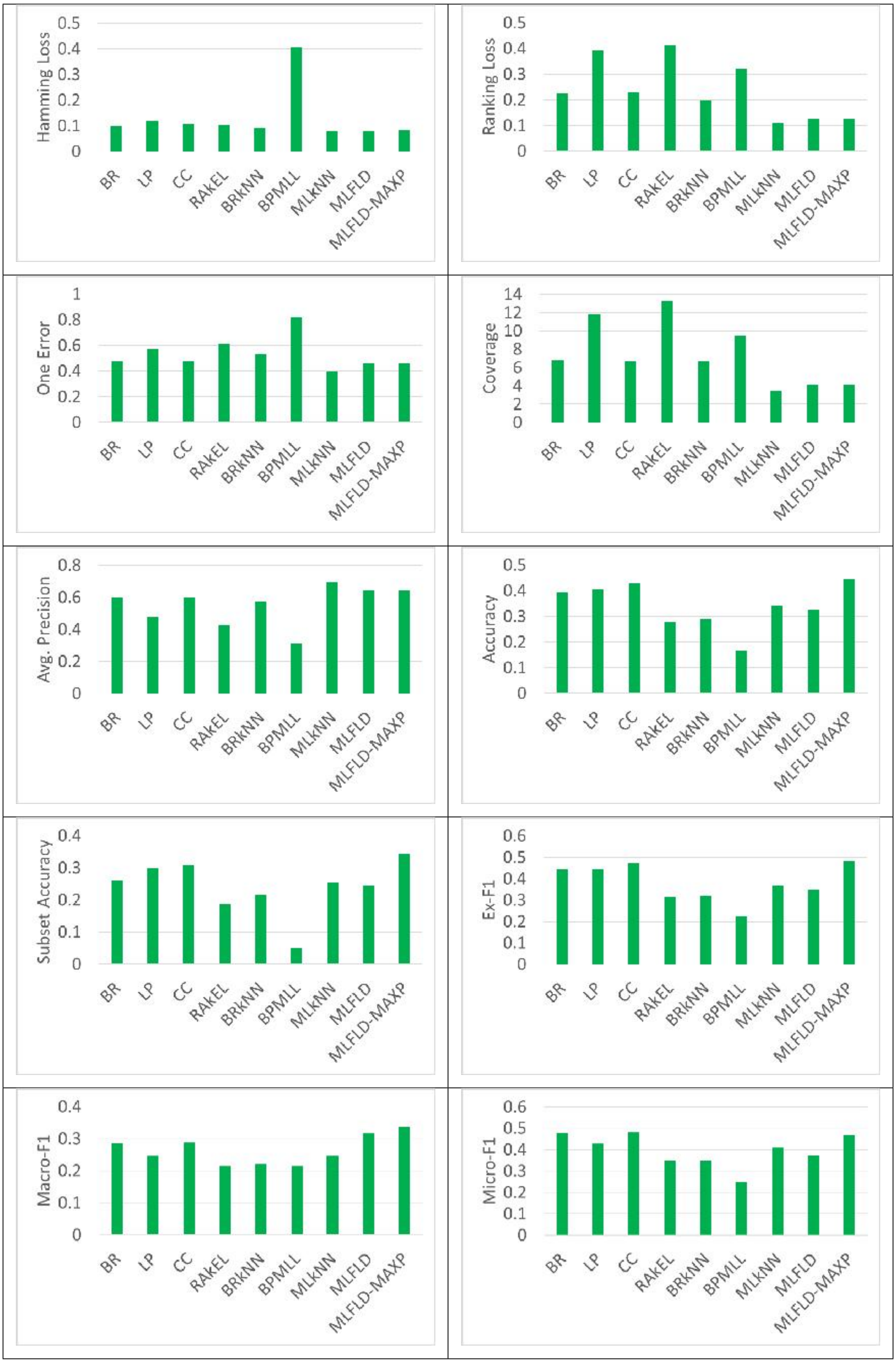
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.4356	0.4835	0.4911	0.5119	0.6104	0.6276	0.6278	0.6472	**0.6483**
Scene	0.6132	0.5870	0.6185	0.6290	0.6380	0.5405	0.7156	0.7387	**0.7502**
Image	0.6941	0.5459	0.6273	0.5856	0.7048	0.3361	0.7166	0.7412	**0.7426**
Yeast	0.5461	0.5141	0.5414	0.4773	0.6193	0.6291	**0.6303**	0.6144	0.6152
Arts Humanity	0.3131	0.2693	**0.3297**	0.1722	0.0580	0.1249	0.0480	0.0373	0.2026
Business Eco.	0.6639	0.6244	0.6614	0.6486	0.6895	0.1736	**0.6990**	0.6739	0.6747
Education	**0.3506**	0.2753	0.3504	0.2384	0.2336	0.1253	0.2541	0.0934	0.2752
Entertainment	0.4023	0.3815	**0.4041**	0.3654	0.2020	0.1683	0.2696	0.1929	0.2780
Health	0.5343	0.4909	**0.5353**	0.2374	0.1725	0.1604	0.4033	0.3834	0.3921
Reference	**0.4780**	0.4057	0.4752	0.4102	0.0408	0.1267	0.1652	0.0645	0.4388
Science	0.2876	0.2154	**0.2923**	0.1396	0.0599	0.0815	0.1063	0.0183	0.2124
Social Science	**0.5424**	0.4693	0.5399	0.0943	0.2487	0.0000	0.3865	0.4244	0.4822
Society Culture	0.3482	0.2975	0.3821	0.0521	0.2825	0.1235	0.2896	0.2207	**0.3826**
Average	0.4776	0.4277	**0.4807**	0.3509	0.3508	0.2475	0.4086	0.3731	0.4688
Rank	2	4	1	7	8	9	5	6	3

TABLE 6.80: Summary of MLFLD and MLFLD-MAXP performance (TrTe) using Hamming distance

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
HamLoss	0.0999	0.1183	0.1058	0.1024	0.0898	0.4077	**0.0769**	0.0804	0.0855
RankLoss	0.2259	0.3938	0.2298	0.4146	0.1962	0.3214	**0.1081**	0.1242	0.1242
OneError	0.4776	0.5704	0.4732	0.6103	0.5323	0.8181	**0.3930**	0.4578	0.4578
Coverage	6.8052	11.7465	6.7022	13.2459	6.6902	9.4685	**3.4687**	4.1012	4.1012
AvgPrec	0.6018	0.4793	0.6028	0.4259	0.5716	0.3137	**0.6933**	0.6394	0.6394
Accuracy	0.3947	0.4065	0.4288	0.2802	0.2915	0.1663	0.3398	0.3232	**0.4439**
SubAcc	0.2598	0.2997	0.3094	0.1875	0.2164	0.0516	0.2548	0.2442	**0.3440**
Ex-F1	0.4442	0.4463	0.4726	0.3147	0.3179	0.2230	0.3685	0.3503	**0.4790**
Macro-F1	0.2860	0.2473	0.2881	0.2151	0.2192	0.2141	0.2462	0.3161	**0.3363**
Micro-F1	0.4776	0.4277	**0.4807**	0.3509	0.3508	0.2475	0.4086	0.3731	0.4688
Avg Rank	4.4	5.6	3.6	7.9	6	8.6	3.1	3.6	**1.8**
#Wins	0	0	1	0	0	0	5	0	**4**

Observations: From Table 6.80 and Table 6.81, MLFLD-MAXP performance is improved in terms of average rank over all the measures though it shows only 4 wins over all the metrics. MLkNN got avg rank 3.1, which is much higher than that of MLFLD-MAXP though it shows 5 wins. Avg rank of MLFLD is twice than of MLFLD-MAXP. It shares avg rank with CC. To summarize,

Table 6.81: Comparison of MLFLD and MLFLD-MAXP Performance (train-test) with Hamming distance



- MLFLD-MAXP got rank 1 for subset accuracy with 11% and 35% improvement over CC and MLkNN, respectively, and improvement for 8 datasets. MLFLD got position 6 with an enhancement for two datasets individually.
- It outperformed for accuracy with 3% and 30% improvement w.r.t. CC and MLkNN, respectively showing improvement for five datasets.
- It outperformed for Ex-F1 with 1% and 30% improvement w.r.t. CC and MLkNN, respectively showing improvement for five datasets.
- MLFLD got rank 6 for accuracy, subset accuracy, micro-F1, and ex-F1.
- MLFLD-MAXP is better among others for macro-F than micro-F, indicating more influenced by rare labels as compared to MLFLD.
- Both proposed algorithms have shown fewer misclassifications than others except MLkNN.
- Both proposed algorithms are similar for one err, rank loss, avg precision, coverage, and defeated other algorithms except MLkNN.

Few observations noted for different behaviors of yahoo datasets:

- All Yahoo datasets have a minimal density between 0.03-0.1 approx.
- %outlier is comparatively more in Scene, Image, Business, Education, Reference, Society, except Emotions and Yeast, both having cardinality and density more than others. For these datasets, MLFLD-MAXP has shown maximum subset accuracy, better accuracy, and ex-F1.
- Scene and Image have less MLE but more outliers.
- Though Business has a more significant skew, it also has larger Ex/Label.
- %Skew (grey) line shows opposite behavior to that of %Ex/Label (orange) line. That is, for less skew, %Ex/label is more and vice-versa, as shown in Figure 5.3 of chapter 5.
- %Skew (grey) line shows similar behavior to that of %Ex/Label (orange) line. That is, for less skew, %Ex/label is also less and vice-versa, as shown in Figure 5.4 of chapter 5.

6.3.4 Effect of distance variation for feature similarity on MLFLD and MLFLD-MAXP using Hamming distance for label dissimilarity

How the performance of proposed algorithms gets affected by distance variation for feature similarity on train-test datasets, is examined in this section from Table 6.82 to 6.91.

TABLE 6.82: Effect of distance variation on Hamming Loss (↓) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	**0.2162**	0.2195	0.2351	0.2277	0.2211	0.2351	0.2277
Scene	0.0962	0.0863	**0.0858**	0.0868	0.0886	0.0907	0.0878
Image	0.1147	0.1127	0.1110	0.1070	0.1160	0.1160	**0.1057**
Yeast	**0.2008**	0.2072	0.2021	0.2109	0.2072	0.2021	0.2107
Arts Humanity	**0.0612**	0.0628	0.0630	0.0656	0.0810	0.0816	0.0818
Business Eco.	**0.0269**	0.0285	0.0289	0.0302	0.0285	0.0289	0.0287
Education	**0.0387**	0.0465	0.0463	0.0443	0.0558	0.0556	0.0552
Entertainment	**0.0604**	0.0722	0.0658	0.0650	0.0847	0.0843	0.0835
Health	**0.0458**	0.0512	0.0518	0.0505	0.0519	0.0524	0.0507
Reference	**0.0314**	0.0354	0.0353	0.0342	0.0371	0.0370	0.0500
Science	**0.0325**	0.0358	0.0358	0.0363	0.0478	0.0480	0.0494
Social Science	**0.0218**	0.0287	0.0289	0.0300	0.0309	0.0313	0.0327
Society Culture	**0.0537**	0.0585	0.0586	0.0597	0.0614	0.0613	0.0622
Average	**0.0769**	0.0804	0.0806	0.0806	0.0855	0.0865	0.0866
Rank	1	2	3	3	5	6	7

Table 6.83: Effect of distance variation on Ranking Loss (↓) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1781	**0.1570**	0.1677	0.1838	**0.1570**	0.1677	0.1838
Scene	0.0930	0.0830	0.0843	**0.0799**	0.0830	0.0843	**0.0799**
Image	0.1154	0.0888	0.0900	**0.0840**	0.0888	0.0900	**0.0840**
Yeast	**0.1766**	0.1839	0.1806	0.1823	0.1839	0.1806	0.1823
Arts Humanity	**0.1514**	0.1707	0.1704	0.1775	0.1707	0.1704	0.1775
Business Eco.	**0.0373**	0.0454	0.0458	0.0474	0.0454	0.0458	0.0474
Education	**0.0800**	0.1112	0.1085	0.1095	0.1112	0.1085	0.1095
Entertainment	**0.1151**	0.1735	0.1777	0.1462	0.1735	0.1777	0.1462
Health	**0.0605**	0.0788	0.0792	0.0776	0.0788	0.0792	0.0776
Reference	**0.0919**	0.1367	0.1378	0.1379	0.1367	0.1378	0.1379
Science	**0.1167**	0.1551	0.1532	0.1596	0.1551	0.1532	0.1596
Social Science	**0.0561**	0.0767	0.0767	0.0770	0.0767	0.0767	0.0770
Society Culture	**0.1338**	0.1543	0.1541	0.1574	0.1543	0.1541	0.1574
Average	**0.1081**	0.1242	0.1251	0.1246	0.1242	0.1251	0.1246
Rank	1	2	6	4	2	6	4

Table 6.84: Effect of distance variation on One Error (↓) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.3218	**0.2970**	0.3267	0.3267	**0.2970**	0.3267	0.3267
Scene	0.2425	**0.2191**	0.2283	**0.2191**	**0.2191**	0.2283	**0.2191**
Image	0.2517	0.2183	0.2167	**0.1950**	0.2183	0.2167	**0.1950**
Yeast	**0.2519**	0.2835	0.2748	0.2672	0.2835	0.2748	0.2672
Arts Humanity	**0.6330**	0.7323	0.7397	0.7403	0.7323	0.7397	0.7403
Business Eco.	**0.1213**	0.1343	0.1357	0.1370	0.1343	0.1357	0.1370
Education	**0.5207**	0.6710	0.6710	0.6807	0.6710	0.6710	0.6807
Entertainment	**0.5300**	0.6897	0.6890	0.6747	0.6897	0.6890	0.6747
Health	**0.4190**	0.5070	0.5147	0.4773	0.5070	0.5147	0.4773
Reference	**0.4730**	0.5227	0.5223	0.7373	0.5227	0.5223	0.7373
Science	**0.5810**	0.7423	0.7470	0.7750	0.7423	0.7470	0.7750
Social Science	**0.3270**	0.4467	0.4543	0.4930	0.4467	0.4543	0.4930
Society Culture	**0.4357**	0.4870	0.4857	0.4973	0.4870	0.4857	0.4973
Average	**0.3930**	0.4578	0.4620	0.4785	0.4578	0.4620	0.4785
Rank	1	2	4	6	2	4	6

Table 6.85: Effect of distance variation on Coverage (↓) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	1.9356	**1.8119**	1.8515	1.9653	**1.8119**	1.8515	1.9653
Scene	0.5661	0.5184	0.5268	**0.5033**	0.5184	0.5268	**0.5033**
Image	0.6083	0.5000	0.5067	**0.4833**	0.5000	0.5067	**0.4833**
Yeast	**6.4318**	6.5540	6.5213	6.5453	6.5540	6.5213	6.5453
Arts Humanity	**5.4313**	5.9870	5.9513	6.1230	5.9870	5.9513	6.1230
Business Eco.	**2.1840**	2.4683	2.4873	2.5450	2.4683	2.4873	2.5450
Education	**3.4973**	4.5247	4.4843	4.5013	4.5247	4.4843	4.5013
Entertainment	**3.1467**	4.3117	4.4050	3.7467	4.3117	4.4050	3.7467
Health	**3.3043**	4.0317	3.9680	3.9573	4.0317	3.9680	3.9573
Reference	**3.5420**	5.0580	5.1023	5.0667	5.0580	5.1023	5.0667
Science	**6.0470**	7.6283	7.5603	7.8150	7.6283	7.5603	7.8150
Social Science	**3.0340**	3.9590	3.9603	3.9847	3.9590	3.9603	3.9847
Society Culture	**5.3653**	5.9630	5.9663	6.0060	5.9630	5.9663	6.0060
Average	**3.4687**	4.1012	4.0993	4.0956	4.1012	4.0993	4.0956
Rank	1	6	4	2	6	4	2

Table 6.86: Effect of distance variation on Average Precision (↑) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.7810	**0.8024**	0.7887	0.7746	**0.8024**	0.7887	0.7746
Scene	0.8511	0.8653	0.8605	**0.8672**	0.8653	0.8605	**0.8672**
Image	0.8456	0.8718	0.8717	**0.8823**	0.8718	0.8717	**0.8823**
Yeast	**0.7505**	0.7396	0.7443	0.7431	0.7396	0.7443	0.7431
Arts Humanity	**0.5097**	0.4459	0.4450	0.4316	0.4459	0.4450	0.4316
Business Eco.	**0.8798**	0.8657	0.8642	0.8560	0.8657	0.8642	0.8560
Education	**0.5993**	0.4806	0.4846	0.4754	0.4806	0.4846	0.4754
Entertainment	**0.6013**	0.4652	0.4651	0.4987	0.4652	0.4651	0.4987
Health	**0.6817**	0.6055	0.6074	0.6264	0.6055	0.6074	0.6264
Reference	**0.6194**	0.5445	0.5444	0.4451	0.5445	0.5444	0.4451
Science	**0.5324**	0.4019	0.4053	0.3818	0.4019	0.4053	0.3818
Social Science	**0.7481**	0.6581	0.6554	0.6356	0.6581	0.6554	0.6356
Society Culture	**0.6128**	0.5651	0.5668	0.5548	0.5651	0.5668	0.5548
Average	**0.6933**	0.6394	0.6387	0.6287	0.6394	0.6387	0.6287
Rank	1	2	4	6	2	4	6

Table 6.87: Effect of distance variation on Accuracy (↑) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.4818	0.5136	0.4979	0.5173	0.5202	0.5054	**0.5289**
Scene	0.6597	0.6749	0.6732	0.6644	0.7389	0.7317	**0.7409**
Image	0.6492	0.7008	0.7044	0.6936	0.7292	0.7286	**0.7461**
Yeast	0.4998	0.4802	**0.5108**	0.4820	0.4821	**0.5108**	0.4843
Arts Humanity	0.0331	0.0262	0.0306	0.0673	**0.2058**	0.2007	0.1933
Business Eco.	**0.6967**	0.6813	0.6773	0.6289	0.6825	0.6773	0.6802
Education	0.1560	0.0433	0.0366	0.0221	0.2636	**0.2679**	0.2475
Entertainment	0.1862	0.1340	0.1014	0.0516	0.2572	0.2559	**0.2679**
Health	0.3390	0.3533	0.3515	0.2506	0.3714	0.3683	**0.3948**
Reference	0.1032	0.0358	0.0319	0.1954	0.4280	**0.4285**	0.2364
Science	0.0695	0.0120	0.0075	0.0115	**0.2141**	0.2104	0.1831
Social Science	0.2996	0.3686	0.3632	0.2299	**0.4938**	0.4839	0.4530
Society Culture	0.2431	0.1770	0.1622	0.1038	0.3833	**0.3849**	0.3744
Average	0.3398	0.3232	0.3191	0.3014	**0.4439**	0.4426	0.4254
Rank	4	5	6	7	1	2	3

Table 6.88: Effect of distance variation on Subset Accuracy (↑) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2178	0.2574	0.2228	0.2574	0.2574	0.2228	**0.2624**
Scene	0.6012	0.6279	0.6246	0.6187	0.6890	0.6798	**0.6898**
Image	0.5983	0.6350	0.6417	0.6350	0.6617	0.6633	**0.6800**
Yeast	0.1647	0.1810	0.1788	**0.1887**	0.1810	0.1788	**0.1887**
Arts Humanity	0.0277	0.0223	0.0263	0.0570	**0.1643**	0.1607	0.1470
Business Eco.	0.5353	0.5357	0.5213	0.4957	**0.5363**	0.5213	0.5360
Education	0.1310	0.0293	0.0247	0.0130	0.2117	**0.2197**	0.1927
Entertainment	0.1687	0.1157	0.0887	0.0447	0.2097	0.2103	**0.2227**
Health	0.2403	0.2517	0.2493	0.1913	0.2637	0.2610	**0.3013**
Reference	0.0963	0.0313	0.0280	0.1783	0.3820	**0.3827**	0.2120
Science	0.0603	0.0110	0.0070	0.0090	**0.1827**	0.1797	0.1530
Social Science	0.2700	0.3313	0.3257	0.2137	**0.4380**	0.4267	0.4080
Society Culture	0.2010	0.1450	0.1333	0.0893	0.2947	**0.2967**	0.2867
Average	0.2548	0.2442	0.2363	0.2301	**0.3440**	0.3387	0.3293
Rank	4	5	6	7	1	2	3

TABLE 6.89: Effect of distance variation on Ex-F1 (↑) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5662	0.5954	0.5843	0.5985	0.6045	0.5942	**0.6126**
Scene	0.6793	0.6906	0.6895	0.6798	0.7556	0.7492	**0.7581**
Image	0.6667	0.7233	0.7261	0.7133	0.7522	0.7511	**0.7683**
Yeast	0.6067	0.5805	**0.6168**	0.5806	0.5835	**0.6168**	0.5845
Arts Humanity	0.0352	0.0277	0.0322	0.0713	**0.2226**	0.2169	0.2118
Business Eco.	**0.7546**	0.7357	0.7344	0.6791	0.7370	0.7344	0.7345
Education	0.1647	0.0481	0.0403	0.0252	0.2828	**0.2858**	0.2683
Entertainment	0.1924	0.1398	0.1052	0.0538	0.2738	0.2720	**0.2844**
Health	0.3772	0.3923	0.3909	0.2735	0.4128	0.4097	**0.4316**
Reference	0.1055	0.0372	0.0332	0.2014	0.4439	**0.4443**	0.2450
Science	0.0728	0.0124	0.0077	0.0125	**0.2264**	0.2224	0.1948
Social Science	0.3100	0.3819	0.3766	0.2358	**0.5139**	0.5045	0.4693
Society Culture	0.2594	0.1896	0.1736	0.1096	0.4184	**0.4198**	0.4091
Average	0.3685	0.3503	0.3470	0.3257	**0.4790**	0.4785	0.4594
Rank	4	5	6	7	1	2	3

TABLE 6.90: Effect of distance variation on Macro-F1 (↑) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5880	0.6275	0.5954	0.6489	0.6273	0.5989	**0.6509**
Scene	0.7156	0.7400	0.7407	0.7341	0.7569	0.7530	**0.7601**
Image	0.5904	0.6104	**0.6248**	0.6048	0.6202	0.6243	0.6218
Yeast	0.3444	0.3887	**0.3987**	0.3922	0.3891	**0.3987**	0.3924
Arts Humanity	0.0343	0.0176	0.0175	0.0353	0.0583	0.0544	**0.0645**
Business Eco.	**0.1817**	NaN	NaN	NaN	NaN	NaN	NaN
Education	**0.1421**	NaN	NaN	NaN	NaN	NaN	NaN
Entertainment	0.1271	0.1031	0.0960	0.0698	0.1370	**0.1395**	0.1271
Health	**0.1567**	NaN	NaN	NaN	NaN	NaN	NaN
Reference	**0.0907**	NaN	NaN	NaN	NaN	NaN	NaN
Science	**0.0408**	0.0072	0.0046	0.0055	0.0407	0.0397	0.0326
Social Science	**0.1175**	NaN	NaN	NaN	NaN	NaN	NaN
Society Culture	**0.0714**	0.0343	0.0302	0.0210	0.0608	0.0488	0.0533
Average	0.2462	0.3161	0.3135	0.3140	0.3363	0.3322	**0.3378**
Rank	7	4	6	5	2	3	1

Table 6.91: Effect of distance variation on Micro-F1 (↑) using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6278	0.6472	0.6265	0.6480	0.6483	0.6294	**0.6515**
Scene	0.7156	0.7387	0.7394	0.7341	0.7502	0.7444	**0.7526**
Image	0.7166	0.7412	0.7429	0.7438	0.7426	0.7407	**0.7600**
Yeast	0.6303	0.6144	**0.6415**	0.6125	0.6152	**0.6415**	0.6137
Arts Humanity	0.0480	0.0373	0.0425	0.0906	**0.2026**	0.1971	0.1990
Business Eco.	**0.6990**	0.6739	0.6764	0.6415	0.6747	0.6764	0.6710
Education	0.2541	0.0934	0.0855	0.0506	0.2752	0.2775	0.2637
Entertainment	0.2696	0.1929	0.1705	0.1002	0.2780	0.2770	**0.2808**
Health	**0.4033**	0.3834	0.3796	0.2997	0.3921	0.3878	0.4000
Reference	0.1652	0.0645	0.0581	0.2754	0.4388	**0.4392**	0.2426
Science	0.1063	0.0183	0.0115	0.0198	**0.2124**	0.2086	0.1856
Social Science	0.3865	0.4244	0.4203	0.2994	**0.4822**	0.4752	0.4431
Society Culture	0.2896	0.2207	0.2055	0.1346	0.3826	**0.3836**	0.3746
Average	0.4086	0.3731	0.3692	0.3577	**0.4688**	0.4676	0.4491
Rank	4	5	6	7	1	2	3

Table 6.92: Summary of effect of distance variation on MLFLD and MLFLD-MAXP performance using Hamming distance and TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	**0.0769**	0.0804	0.0806	0.0806	0.0855	0.0865	0.0866
RankLoss	**0.1081**	0.1242	0.1251	0.1246	0.1242	0.1251	0.1246
OneError	**0.3930**	0.4578	0.4620	0.4785	0.4578	0.4620	0.4785
Coverage	**3.4687**	4.1012	4.0993	4.0956	4.1012	4.0993	4.0956
AvgPrec	**0.6933**	0.6394	0.6387	0.6287	0.6394	0.6387	0.6287
Accuracy	0.3398	0.3232	0.3191	0.3014	**0.4439**	0.4426	0.4254
SubAcc	0.2548	0.2442	0.2363	0.2301	**0.3440**	0.3387	0.3293
Ex-F1	0.3685	0.3503	0.3470	0.3257	**0.4790**	0.4785	0.4594
Macro-F1	0.2462	0.3161	0.3135	0.314	0.3363	0.3322	**0.3378**
Micro-F1	0.4086	0.3731	0.3692	0.3577	**0.4688**	0.4676	0.4491
ExecTime	6	28	31	107	28	28	102
Avg Rank	2.8	3.8	5.1	5.4	**2.3**	3.5	3.8
#Wins	5	0	0	0	**4**	0	1

Observations: From Table 6.92, average rank of MLFLD-MAXP using Euclidean and Hamming distances is found better among all the seven experiments, though it wins only four times as compared to MLkNN that wins five times. But it should be noted that the average rank of MLFLD-MAXP using Euclidean has exceeded that of MLkNN. To summarize,

- Performance improvement for both accuracy and Ex-F1 by 3 MLFLD-MAXP variations is approx. 25-37% while it is 10-15% for label-based measures.
- For two accuracy and three F measures, three distance variations of MLFLD-MAXP beat MLkNN while MLFLD could not. MLFLD-MAXP with Euclidean, Manhattan, and Minkowski functioned at rank 1, 2, and 3 respectively for these five measures with approx. 10-15% rise.
- The performance of MLFLD is very close to that of MLkNN. MLFLD-Euclidean pair have done minimal misclassification among six variations of proposed algorithms. (MLFLD, Manhattan) and (MLFLD-MAXP, Minkowski) pairs have performed better for Scene and Image respectively while both proposed algorithms could not exceed MLkNN for remaining datasets.
- The performance of MLFLD is the same as that of MLFLD-MAXP for coverage, one error, rank loss, and avg precision. Both functioned better for these metrics with Emotions, Image, and Scene, whereas they could not work well for remaining datasets. The reason may be that
  - The percentage of outliers is substantial for Scene (72) and Image (86) as compared to the remaining 11 datasets, as shown in Table 5.2 and 5.4.
  - No. of unique label sets is also more in all datasets except Scene and Image having 0.6% and 1% unique label sets, respectively, as shown in Table 5.1 and 5.3 from Chapter 5.
- Use of Minkowski distance requires more computation time among all variations.

6.4 Performance of proposed algorithms after outlier removal

An outlier is a value that lies away by threshold 3.0 ($\pm$ 1.5) from the mean. Such values affect the predictive performance of a classifier. They can be removed from datasets using Weka and Meka [73] [75].

Experimentation is performed on datasets after outlier removal, and performance is analyzed for cross-validation as well as train-test datasets. It is perceived that among all the contesting algorithms, MLkNN is the best contestant. Hence in this section performance of proposed algorithms is compared with only MLkNN.

6.4.1 Performance of proposed algorithms with cross-validation after outlier removal

TABLE 6.93: Effect of outlier removal on MLFLD and MLFLD-MAXP using cross-validation

(a) *Hamming loss* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1878	0.1115	**0.1104**
Scene	0.1052	0.0914	**0.0877**
Image	0.1919	**0.1444**	0.1474
Yeast	0.1967	**0.1522**	**0.1522**
CAL500	0.1394	**0.1324**	**0.1324**
Average	0.1642	0.1264	**0.1260**
Rank	3	2	1

(b) *Ranking loss* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1582	**0.0502**	**0.0502**
Scene	0.0946	**0.0669**	**0.0669**
Image	0.2089	**0.1537**	**0.1537**
Yeast	0.1638	**0.0971**	**0.0971**
CAL500	0.1837	**0.1696**	**0.1696**
Average	0.1618	**0.1075**	**0.1075**
Rank	3	1	1

(c) *One Error* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2599	**0.1042**	**0.1042**
Scene	0.2910	**0.2302**	**0.2302**
Image	0.3765	**0.2815**	**0.2815**
Yeast	0.2222	**0.1147**	**0.1147**
CAL500	0.1095	**0.0597**	**0.0597**
Average	0.2518	**0.1581**	**0.1581**
Rank	3	1	1

(d) *Coverage* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	1.7959	**1.1792**	**1.1792**
Scene	0.5612	**0.4154**	**0.4154**
Image	1.0545	**0.8259**	**0.8259**
Yeast	6.2599	**5.1735**	**5.1735**
CAL500	131.0571	**130.0358**	**130.0358**
Average	28.1457	**27.5260**	**27.5260**
Rank	3	1	1

Table 6.94: Effect of outlier removal on MLFLD and MLFLD-MAXP using cross-validation

(e) *Average Precision* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.8073	**0.9278**	**0.9278**
Scene	0.8301	**0.8700**	**0.8700**
Image	0.7568	**0.8201**	**0.8201**
Yeast	0.7696	**0.8634**	**0.8634**
CAL500	0.4946	**0.5369**	**0.5369**
Average	0.7317	**0.8036**	**0.8036**
Rank	3	1	1

(f) *Accuracy* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.5665	0.7276	**0.7380**
Scene	0.6060	0.6667	**0.7407**
Image	0.3937	0.5722	**0.6630**
Yeast	0.5058	0.6235	**0.6236**
CAL500	0.1936	**0.2385**	**0.2385**
Average	0.4531	0.5657	**0.6008**
Rank	3	2	1

(g) *Subset Accuracy* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.3223	0.5083	**0.5167**
Scene	0.5701	0.6189	**0.6907**
Image	0.3501	0.5148	**0.5963**
Yeast	0.1805	0.2806	**0.2806**
CAL500	0.0000	0.0000	0.0000
Average	0.2846	0.3845	**0.4169**
Rank	3	2	1

(h) *Ex-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6458	0.7948	**0.8059**
Scene	0.6179	0.6826	**0.7574**
Image	0.4084	0.5920	**0.6858**
Yeast	0.6111	0.7206	0.7209
CAL500	0.3186	**0.3781**	**0.3781**
Average	0.5204	0.6336	**0.6696**
Rank	3	2	1

(i) *Macro-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6404	0.8166	**0.8196**
Scene	0.6336	0.6998	**0.7397**
Image	0.4455	0.5961	**0.6153**
Yeast	**0.3858**	NaN	NaN
CAL500	**0.1957**	NaN	NaN
Average	0.4602	0.7042	**0.7249**
Rank	3	2	1

(j) *Micro-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6814	0.8220	**0.8247**
Scene	0.6715	0.7225	**0.7514**
Image	0.4768	0.6414	**0.6700**
Yeast	0.6396	0.7403	**0.7404**
CAL500	0.3147	**0.3831**	**0.3831**
Average	0.5568	0.6619	**0.6739**
Rank	3	2	1

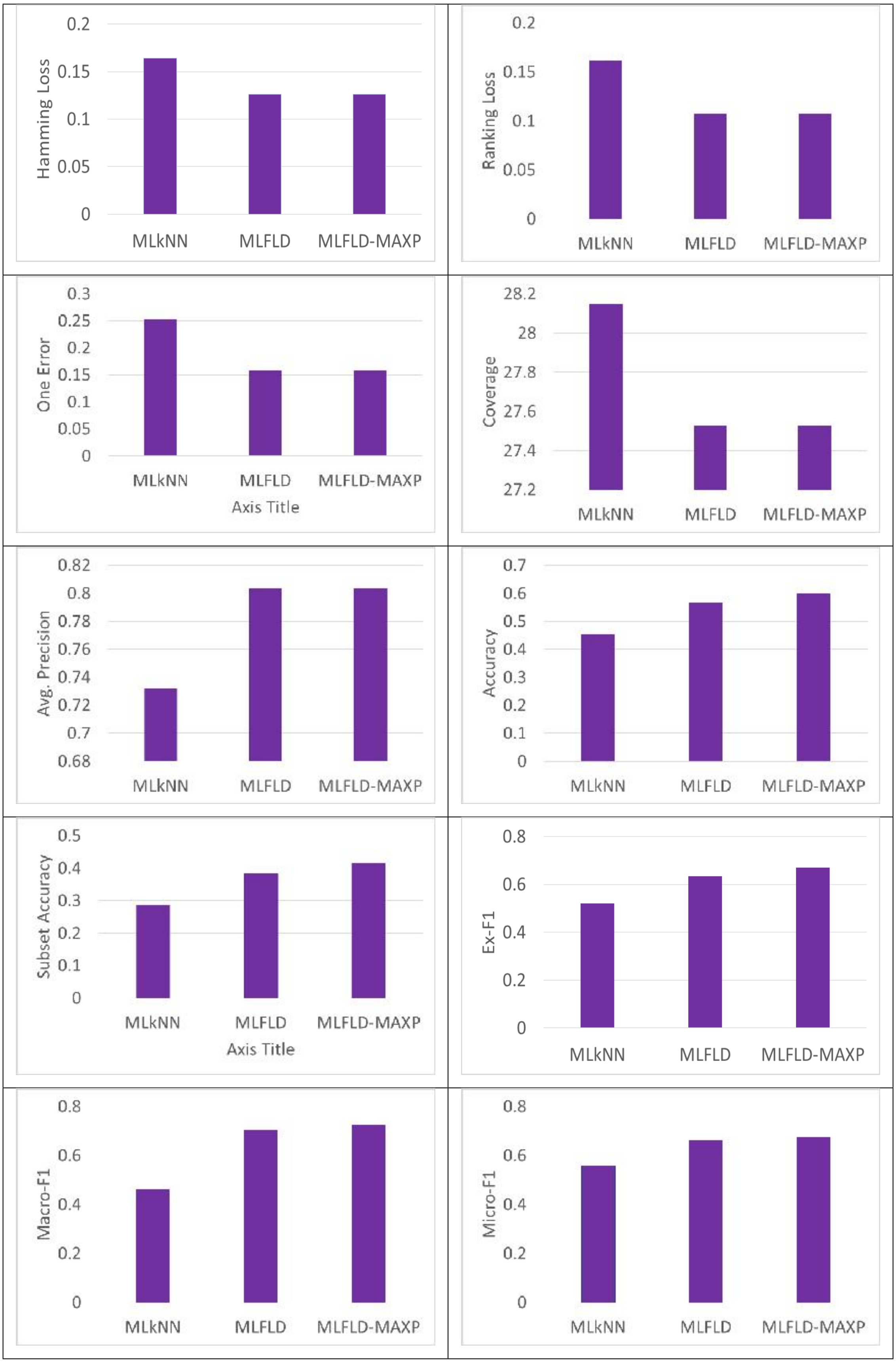
TABLE 6.95: Summary of MLFLD and MLFLD-MAXP (CV) performance for checking effect of outlier removal

Metric	MLkNN	MLFLD	MLFLD-MAXP
HamLoss	0.1642	0.1264	**0.1260**
RankLoss	0.1618	**0.1075**	**0.1075**
OneError	0.2518	**0.1581**	**0.1581**
Coverage	28.1457	**27.5260**	**27.5260**
AvgPrec	0.7317	0.8036	**0.8036**
Accuracy	0.4531	0.5657	**0.6008**
SubAcc	0.2846	0.3845	**0.4169**
Ex-F1	0.5204	0.6336	**0.6696**
Macro-F1	0.4602	0.7042	**0.7249**
Micro-F1	0.5568	0.6619	**0.6739**
ExecTime	6	8	8
Avg rank	3.0	1.6	**1.0**
#Wins	0	4	**10**

After removing outliers, datasets are fed to three algorithms to be evaluated. In this section, proposed algorithms are observed for Euclidean and Hamming distance for ten folds shown in Table 6.93 and 6.95.

Observations: Summary Table 6.95 shows that after removing outliers from datasets and applying cross-validation, both proposed algorithms have beaten competing algorithm. MLFLD-MAXP has shown more growth than MLFLD.

TABLE 6.96: Performance of proposed algorithms with cross-validation after Outlier removal



To summarize,

- Figures in Table 6.96(a)-(e) show that proposed algorithms have worked similarly for the first 5 metrics, whereas MLFLD-MAXP has exceeded MLFLD for the remaining 5 metrics, as in Table 6.96(g)-(j).
- Previous sections have shown that MLFLD has always proved itself better than MLFLD-MAXP for improvement in the hamming loss. Table 6.96(a) shows that after outlier removal, MLFLD-MAXP seems to behave better in terms of hamming loss.
- Both proposed algorithms have shown the same performance for one error, ranking loss, coverage, and avg precision with 37, 33, 10, and 2 percent improvement over MLkNN, respectively.
- Maximum improvement is seen for subset accuracy, that is 46% and 35%, whereas 32% and 24% for accuracy with MLFLD-MAXP and MLFLD, respectively.
- MLFLD-MAXP defeated MLFLD for ex-F1 and micro-F1 by (28, 21) and (21, 18) percent, respectively. Both algorithms have enhanced compared to MLkNN for 3 datasets, but could not compute macro-F1 for 2 datasets.
- The execution time of all experiments is comparable.

6.4.2 Performance of proposed algorithms with train-test splits after Outlier removal

In this section, evaluation carried out after outlier removal from train-test (TrTe) splits are monitored for three algorithms shown in Table 6.97-6.99.

TABLE 6.97: Effect of outlier removal on MLFLD and MLFLD-MAXP using TrTe

(a) *Hamming loss* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2246	**0.1341**	0.1382
Scene	0.1275	**0.1104**	0.1156
Image	0.1863	**0.1649**	0.1664
Yeast	0.1986	0.1549	**0.1548**
Arts Humanity	0.0580	0.0558	**0.0471**
Business Eco.	0.0291	**0.0159**	0.0171
Education	**0.0419**	0.0458	0.0458
Entertainment	0.0643	**0.0577**	0.0697
Health	0.0456	**0.0326**	0.0357
Reference	**0.0296**	0.0348	0.0363
Science	**0.0357**	**0.0357**	0.0493
Social Science	0.0295	**0.0263**	0.0268
Society Culture	0.0547	**0.0531**	0.0551
Average	0.0866	**0.0709**	0.0737
Rank	3	1	2

(b) *Ranking loss* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1857	**0.0844**	**0.0844**
Scene	0.1133	0.**0951**	**0.0951**
Image	0.2481	**0.1858**	**0.1858**
Yeast	0.1659	**0.1012**	**0.1012**
Arts Humanity	0.0987	**0.0543**	**0.0543**
Business Eco.	0.0469	**0.0438**	**0.0438**
Education	0.0900	**0.0716**	**0.0716**
Entertainment	0.1219	**0.1058**	**0.1058**
Health	0.0695	**0.0545**	**0.0545**
Reference	**0.0908**	0.1301	0.1301
Science	**0.1432**	0.1574	0.1574
Social Science	0.0714	**0.0666**	**0.0666**
Society Culture	0.1463	**0.1227**	**0.1227**
Average	0.1224	**0.0979**	**0.0979**
Rank	3	1	1

(c) *One Error* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2988	**0.1707**	**0.1707**
Scene	0.3003	**0.2972**	**0.2972**
Image	0.4427	**0.3664**	**0.3664**
Yeast	0.2456	**0.1360**	**0.1360**
Arts Humanity	0.4481	**0.2808**	**0.2808**
Business Eco.	0.1415	**0.0789**	**0.0789**
Education	0.5871	**0.5107**	**0.5107**
Entertainment	0.5837	**0.5244**	**0.5244**
Health	0.4580	**0.2546**	**0.2546**
Reference	**0.4924**	0.5178	0.5178
Science	**0.7236**	0.7696	0.7696
Social Science	0.4497	**0.3734**	**0.3734**
Society Culture	0.4645	**0.4041**	**0.4041**
Average	0.4335	**0.3604**	**0.3604**
Rank	3	1	1

(d) *Coverage* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	1.9756	1.4451	1.4451
Scene	0.6749	**0.5789**	**0.5789**
Image	1.1069	**0.8626**	**0.8626**
Yeast	6.3406	**5.2953**	**5.2953**
Arts Humanity	4.0149	**2.6072**	**2.6072**
Business Eco.	2.5865	**2.5272**	**2.5272**
Education	3.8607	**3.1843**	**3.1843**
Entertainment	3.2906	**2.8908**	**2.8908**
Health	3.6281	**3.1385**	**3.1385**
Reference	**3.4295**	4.7736	4.7736
Science	**7.1862**	7.7896	7.7896
Social Science	3.7175	**3.6077**	**3.6077**
Society Culture	5.7378	**5.0177**	**5.0177**
Average	3.6577	**3.3630**	**3.3630**
Rank	3	1	1

Table 6.98: Effect of outlier removal on MLFLD and MLFLD-MAXP using TrTe

(e) *Average Precision* (↑)			
Dataset	MLkNN	MLFLD	MAXP
Emotions	0.7874	**0.8823**	**0.8823**
Scene	0.8175	**0.8301**	**0.8301**
Image	0.7075	**0.7640**	**0.7640**
Yeast	0.7618	**0.8542**	**0.8542**
Arts Humanity	0.6421	**0.7710**	**0.7710**
Business Eco.	0.8625	**0.8940**	**0.8940**
Education	0.5492	**0.6056**	**0.6056**
Entertainment	0.5667	**0.6160**	**0.6160**
Health	0.6461	**0.7555**	**0.7555**
Reference	**0.6039**	0.5495	0.5495
Science	**0.4244**	0.3788	0.3788
Social Science	0.6701	**0.7175**	**0.7175**
Society Culture	0.5832	**0.6227**	**0.6227**
Average	0.6633	**0.7109**	**0.7109**
Rank	3	1	1

(f) *Accuracy* (↑)			
Dataset	MLkNN	MLFLD	MAXP
Emotions	0.5173	0.6657	**0.6768**
Scene	0.5341	0.5712	**0.6625**
Image	0.3677	0.4656	**0.5954**
Yeast	0.4992	0.6209	**0.6212**
Arts Humanity	0.1350	0.2427	**0.5299**
Business Eco.	0.6823	0.7993	**0.8189**
Education	0.0945	**0.3756**	**0.3756**
Entertainment	0.1268	0.1838	**0.4028**
Health	0.2637	0.4847	**0.5491**
Reference	0.2246	0.0417	0.4356
Science	0.0173	0.0136	0.1893
Social Science	0.2109	0.3215	**0.5565**
Society Culture	0.2805	0.3262	**0.4277**
Average	0.3041	0.3933	**0.5263**
Rank	3	2	1

(a) *Subset Accuracy* (↑)			
Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2927	0.4451	**0.4512**
Scene	0.4861	0.5263	**0.6130**
Image	0.3359	0.4275	**0.5496**
Yeast	0.1813	**0.2895**	**0.2895**
Arts Humanity	0.1016	0.1904	**0.3839**
Business Eco.	0.5326	0.7013	**0.7122**
Education	0.0725	**0.2865**	**0.2865**
Entertainment	0.1157	0.1530	**0.3485**
Health	0.1930	0.3817	**0.3915**
Reference	0.2134	0.0368	**0.3928**
Science	0.0147	0.0118	**0.1585**
Social Science	0.1930	0.2868	**0.4964**
Society Culture	0.2284	0.2350	**0.3093**
Average	0.2278	0.3055	**0.4141**
Rank	3	2	1

(b) *Ex-F1* (↑)			
Dataset	MLkNN	MLFLD	MAXP
Emotions	0.5935	0.7311	**0.7443**
Scene	0.5501	0.5862	**0.6791**
Image	0.3791	0.4796	**0.6120**
Yeast	0.6064	0.7174	**0.7180**
Arts Humanity	0.1473	0.2629	**0.5850**
Business Eco.	0.7370	0.8284	**0.8513**
Education	0.1024	**0.4091**	**0.4091**
Entertainment	0.1311	0.1956	**0.4234**
Health	0.2903	0.5243	0.6081
Reference	0.2286	0.0434	**0.4505**
Science	0.0182	0.0141	**0.2011**
Social Science	0.2172	0.3339	**0.5781**
Society Culture	0.3008	0.3621	**0.4743**
Average	0.3309	0.4222	**0.5642**
Rank	3	2	1

TABLE 6.99: Effect of outlier removal on MLFLD and MLFLD-MAXP using TrTe

(c) *Macro-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6261	**0.7875**	0.7855
Scene	0.5265	0.6174	**0.6740**
Image	0.5028	0.5897	**0.5950**
Yeast	0.3792	0.4961	**0.4963**
Arts Humanity	**0.1286**	NaN	NaN
Business Eco.	**0.1527**	NaN	NaN
Education	**0.2265**	NaN	NaN
Entertainment	0.0819	0.1403	**0.1788**
Health	**0.2614**	NaN	NaN
Reference	**0.2251**	NaN	NaN
Science	0.0285	0.0227	**0.0596**
Social Science	**0.1210**	NaN	NaN
Society Culture	0.0622	0.0915	**0.1023**
Average	0.2556	0.3922	**0.4131**
Rank	3	2	1

(d) *Micro-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6371	**0.7836**	0.7806
Scene	0.6048	0.6503	**0.6725**
Image	0.4649	0.5537	**0.6093**
Yeast	0.6317	0.7339	**0.7341**
Arts Humanity	0.2041	0.3103	**0.5444**
Business Eco.	0.6761	**0.8286**	0.8211
Education	0.1585	**0.4089**	**0.4089**
Entertainment	0.1779	0.2824	**0.3995**
Health	0.3552	0.5599	**0.5808**
Reference	0.3270	0.0757	**0.4464**
Science	0.0273	0.0226	**0.1901**
Social Science	0.2891	0.4202	**0.5488**
Society Culture	0.3312	0.3932	**0.4484**
Average	0.3758	0.4633	**0.5527**
Rank	3	2	1

Table 6.97, 6.98 and 6.99 show that

- The time required by the proposed algorithms is almost twice than of MLkNN due to label dissimilarity computation at the cost of performance enhancement for 9 metrics.
- For datasets with train-test splits, MLFLD has improved hamming loss with 18% than MLFLD-MAXP with 14% compared to MLkNN.
- Proposed algorithms performed equally well for rank loss, one error, coverage, and avg precision with 20, 16, 8, and 7 %improvement than MLkNN, respectively.
- More improvement is seen in subset accuracy and example-based accuracy by MLFLD-MAXP by 81% and 73% than 34% and 29% improvement of MLFLD, respectively.

TABLE 6.100: Summary of MLFLD and MLFLD-MAXP performance using TrTe for effect of outlier removal

Metric	MLkNN	MLFLD	MLFLD-MAXP
HamLoss	0.0866	**0.0709**	0.0737
RankLoss	0.1224	**0.0979**	**0.0979**
OneError	0.4335	**0.3604**	**0.3604**
Coverage	3.6577	**3.3630**	**3.3630**
AvgPrec	0.6633	**0.7109**	**0.7109**
Accuracy	0.3041	0.3933	**0.5263**
SubAcc	0.2278	0.3055	**0.4141**
Ex-F1	0.3309	0.4222	**0.5642**
Macro-F1	0.2556	0.3922	**0.4131**
Micro-F1	0.3758	0.4633	**0.5527**
ExecTime	5	11	11
Avg rank	3.0	1.5	**1.1**
#Wins	0	5	**9**

- MLFLD-MAXP has outperformed with all datasets for Ex-F1 and 11 datasets for micro-F1 with 70% and 47% resp. MLFLD resulted in 27% and 23% growth resp. Macro-F1 is increased for 6 out of 7 datasets while no computation for five datasets.

Observations: From Table 6.95 and 6.100, it is marked that proposed algorithms are sensitive to outlier data present in datasets. After removing outliers, MLFLD-MAXP has defeated contestant algorithm followed by MLFLD. From figures in Table 6.101, the behavior of proposed algorithms is noticed identical for the first 5 metrics while better for the last 5 metrics.

TABLE 6.101: Performance of proposed algorithms with train-test splits after outlier removal

6.5 Performance of proposed algorithms for large datasets

In all previous sections, comparatively smaller datasets are used. In this section, proposed algorithms are evaluated on two large datasets, namely Cbmi09-bow and Mediamill. As these datasets have 43907 examples, ten-fold cross-validation caused the system to hang with the current configuration. Hence train-test (TrTe) splits are used for experimentation.

From Table 5.4, both datasets show 89.6% MLE denoting large no. of examples associated with multiple labels. They are the only datasets in this work that show 3.9% ZLE, indicating zero label examples. That is the reason for NaN value for many measures. Denominator results in zero when there is no relevant label. For example-based measures, even if one example results in NaN, then corresponding measure results into NaN. Same for label-based metrics.

6.5.1 Performance of MLFLD for large datasets

In this section, the functioning of proposed algorithm MLFLD is studied.

Euclidean distance is used for the computation of the feature similarity and Hamming distance is used for the computation of the label dissimilarity.

The performance is shown is Table 6.102 and 6.103 for the ten metrics after evaluation.

Table 6.102: Performance of MLFLD for large datasets

(a) Hamming loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.0472	0.0544	0.0467	0.0434	**0.0331**	0.0638	0.0331	0.0336
Mediamill	0.0412	0.0494	0.0424	0.0409	0.0318	0.0645	**0.0316**	0.0317
Average	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0327
Rank	5	7	6	4	2	8	1	3

(b) Ranking loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.2939	0.3865	0.2684	0.3487	0.0967	0.0633	**0.0604**	NaN
Mediamill	0.2142	0.3668	0.2116	0.3465	0.0853	**0.0516**	0.0533	NaN
Average	0.2541	0.3767	0.2400	0.3476	0.0910	0.0575	**0.0569**	NaN
Rank	5	7	4	6	3	2	1	-

(c) One Error (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.5836	0.7697	0.3734	0.4300	**0.2020**	0.2375	0.2043	0.2032
Mediamill	0.5256	0.6951	0.3495	0.4236	0.1810	0.2205	**0.1804**	0.1809
Average	0.5546	0.7324	0.3615	0.4268	0.1915	0.2290	0.1924	**0.1921**
Rank	7	8	5	6	1	4	3	2

(d) Coverage (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	69.7428	66.8468	61.7521	67.2632	31.4247	21.2520	20.1887	**20.1426**
Mediamill	58.7190	64.4715	53.4878	67.5235	28.8667	**18.2272**	18.8066	18.8340
Average	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**
Rank	6	7	5	8	4	3	2	1

(e) Average Precision (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.4116	0.2169	0.4549	0.2725	0.6547	0.6366	**0.6738**	NaN
Mediamill	0.4880	0.2670	0.5156	0.2833	0.6850	0.6782	**0.7005**	NaN
Average	0.4498	0.2420	0.4853	0.2779	0.6699	0.6574	**0.6872**	NaN
Rank	5	7	4	6	2	3	1	-

Table 6.103: Performance of MLFLD for large datasets

				(f) Accuracy (↑)				
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.2981	0.2690	0.3032	0.1866	0.3899	0.3139	**0.4009**	NaN
Mediamill	0.3477	0.3140	0.3545	0.2078	0.4176	0.3374	**0.4200**	NaN
Average	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	NaN
Rank	5	6	3	7	2	4	1	-
				(g) Subset Accuracy (↑)				
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.0169	0.0302	0.0447	0.0098	**0.1032**	0.0000	0.1013	0.1026
Mediamill	0.0339	0.0500	0.0661	0.0126	**0.1125**	0.0077	0.1074	0.1111
Average	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.1069
Rank	6	5	4	7	1	8	3	2
				(h) Ex-F1 (↑)				
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.4193	0.3729	0.4079	0.2857	0.4933	0.4542	**0.5083**	NaN
Mediamill	0.4706	0.4231	0.4648	0.3139	0.5263	0.4766	**0.5317**	NaN
Average	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	NaN
Rank	4	6	5	7	2	3	1	-
				(i) Macro-F1 (↑)				
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.0742	0.0696	0.0705	0.0242	0.0618	0.0600	0.0939	**0.1146**
Mediamill	**0.1349**	0.1102	0.1192	0.0361	0.1056	0.0933	0.1063	0.1128
Average	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	**0.1137**
Rank	2	5	4	8	6	7	3	1
				(j) Micro-F1 (↑)				
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
Cbmi09-bow	0.4343	0.3848	0.4247	0.2960	0.5033	0.4715	**0.5184**	0.5084
Mediamill	0.4882	0.4352	0.4824	0.3259	0.5415	0.4935	**0.5442**	0.5433
Average	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5259
Rank	5	7	6	8	3	4	1	2

TABLE 6.104: Summary of MLFLD Performance for large datasets

(a) Summary of 10 metics

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
HamLoss	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0327
RankLoss	0.2541	0.3767	0.2400	0.3476	0.0910	0.0575	**0.0569**	NaN
OneError	0.5546	0.7324	0.3615	0.4268	0.1915	0.2290	0.1924	**0.1921**
Coverage	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**
AvgPrec	0.4498	0.2420	0.4853	0.2779	0.6699	0.6574	**0.6872**	NaN
Accuracy	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	NaN
SubAcc	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.1069
Ex-F1	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	NaN
Macro-F1	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	**0.1137**
Micro-F1	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5259
Avg Rank	5	6.5	4.6	6.7	2.6	4.6	**1.7**	1.8
#Wins	0	0	0	0	2	0	**6**	2

(b) Summary of 6 metrics without NaN

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD
HamLoss	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0327
OneError	0.5546	0.7324	0.3615	0.4268	0.1915	0.2290	0.1924	**0.1921**
Coverage	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**
SubAcc	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.1069
Macro F1	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	**0.1137**
Micro F1	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5259
Avg Rank	5.2	6.5	5.0	6.8	2.8	5.7	2.2	**1.8**
#Wins	0	0	0	0	2	0	2	**2**

Observations: From Table 6.104(a), a conclusion could not be drawn as MLFLD is not able to measure four metrics (shown by NaN). When compared using the remaining six parameters, MLFLD had resulted in the smallest average rank, as shown in Table 6.104(b). #Wins of all neighbour-based algorithms are the same. For macro-F and coverage, the overall achievement is enhanced by 13% and 0.05% compared to MLkNN resp. For subset accuracy, one error, and macro-F, MLFLD ranked second with a result near to MLkNN.

6.5.2 Performance of MLFLD-MAXP for large datasets

In this section, large datasets are used to evaluate MLFLD-MAXP shown in Table 6.105 and 6.106.

Table 6.105: Performance of MLFLD-MAXP for large datasets

(a) Hamming loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.0472	0.0544	0.0467	0.0434	0.0331	0.0638	**0.0331**	0.0337
Mediamill	0.0412	0.0494	0.0424	0.0409	0.0318	0.0645	**0.0316**	0.0318
Average	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0328
Rank	5	7	6	4	2	8	1	3

(b) Ranking loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.2939	0.3865	0.2684	0.3487	0.0967	0.0633	**0.0604**	NaN
Mediamill	0.2142	0.3668	0.2116	0.3465	0.0853	**0.0516**	0.0533	NaN
Average	0.2541	0.3767	0.2400	0.3476	0.0910	0.0575	**0.0569**	NaN
Rank	5	7	4	6	3	2	1	NaN

(c) One Error (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.5836	0.7697	0.3734	0.4300	**0.2020**	0.2375	0.2043	0.2032
Mediamill	0.5256	0.6951	0.3495	0.4236	0.1810	0.2205	**0.1804**	0.1809
Average	0.5546	0.7324	0.3615	0.4268	0.1915	0.2290	0.1924	**0.1921**
Rank	7	8	5	6	1	4	3	2

(d) Coverage (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	69.7428	66.8468	61.7521	67.2632	31.4247	21.2520	20.1887	**20.1426**
Mediamill	58.7190	64.4715	53.4878	67.5235	28.8667	**18.2272**	18.8066	18.8340
Average	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**
Rank	6	7	5	8	4	3	2	1

(e) Average Precision (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.4116	0.2169	0.4549	0.2725	0.6547	0.6366	**0.6738**	NaN
Mediamill	0.4880	0.2670	0.5156	0.2833	0.6850	0.6782	**0.7005**	NaN
Average	0.4498	0.2420	0.4853	0.2779	0.6699	0.6574	**0.6872**	NaN
Rank	5	7	4	6	2	3	1	-

(f) Accuracy (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.2981	0.2690	0.3032	0.1866	0.3899	0.3139	**0.4009**	0.3869
Mediamill	0.3477	0.3140	0.3545	0.2078	0.4176	0.3374	**0.4200**	0.4176
Average	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	0.4023
Rank	6	7	4	8	2	5	1	3

Table 6.106: Performance of MLFLD-MAXP for large datasets

(g) Subset Accuracy (↑)								
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.0169	0.0302	0.0447	0.0098	**0.1032**	0.0000	0.1013	0.0939
Mediamill	0.0339	0.0500	0.0661	0.0126	**0.1125**	0.0077	0.1074	0.1050
Average	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.0995
Rank	6	5	4	7	1	8	2	3
(h) Ex-F1 (↑)								
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.4193	0.3729	0.4079	0.2857	0.4933	0.4542	**0.5083**	0.4947
Mediamill	0.4706	0.4231	0.4648	0.3139	0.5263	0.4766	**0.5317**	0.5298
Average	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	0.5123
Rank	5	7	6	8	3	4	1	2
(i) Macro-F1 (↑)								
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.0742	0.0696	0.0705	0.0242	0.0618	0.0600	0.0939	**0.1155**
Mediamill	**0.1349**	0.1102	0.1192	0.0361	0.1056	0.0933	0.1063	0.1148
Average	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	**0.1152**
Rank	2	5	4	8	6	7	3	1
(j) Micro-F1 (↑)								
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
Cbmi09-bow	0.4343	0.3848	0.4247	0.2960	0.5033	0.4715	**0.5184**	0.5118
Mediamill	0.4882	0.4352	0.4824	0.3259	0.5415	0.4935	0.5442	**0.5450**
Average	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5284
Rank	5	7	6	8	3	4	1	2

Observations: Metrics getting NaN value are removed from Table 6.107(a). From the resulting Table 6.107(b), MLFLD-MAXP has functioned better than all comparing algorithms except MLkNN in terms of average rank. It stood at rank 2.1 among eight algorithms. The functioning of MLFLD-MAXP is better than that of MLkNN for three measures, namely macro-F1 with a 15% rise and coverage, one error with a small rise. It is comparable to the remaining measures. For micro, ex-F1, and one error, the algorithm is ranked second.

TABLE 6.107: Summary of MLFLD-MAXP Performance for large datasets1

(a) Summary for 10 metrics

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
HamLoss	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0328
RankLoss	0.2541	0.3767	0.2400	0.3476	0.0910	0.0575	**0.0569**	NaN
OneError	0.5546	0.7324	0.3615	0.4268	**0.1915**	0.2290	0.1924	0.1921
Coverage	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**
AvgPrec	0.4498	0.2420	0.4853	0.2779	0.6699	0.6574	**0.6872**	NaN
Accuracy	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	0.4023
SubAcc	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.0995
Ex-F1	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	0.5123
Macro-F1	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	**0.1152**
Micro-F1	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5284
Avg Rank	5.2	6.7	4.8	6.9	2.7	4.8	**1.6**	2.1
#Wins	0	0	0	0	2	0	**6**	2

(b) Summary for 8 metrics without NaN

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MAXP
HamLoss	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0328
OneError	0.5546	0.7324	0.3615	0.4268	**0.1915**	0.2290	0.1924	0.1921
Coverage	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**
Accuracy	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	0.4023
SubAcc	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.0995
Ex-F1	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	0.5123
Macro F1	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	**0.1152**
Micro F1	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5284
Avg Rank	5.3	6.6	5.0	7.1	2.8	5.4	**1.8**	2.1
#Wins	0	0	0	0	2	0	**4**	2

6.5.3 Performance of MLFLD and MLFLD-MAXP for large datasets

Effect of applying both the proposed algorithms on large datasets is analyzed in this section for ten parameters (Table 6.108 and 6.109). Cbmi09-bow and Mediamill datasets have all the characteristics similar to each other except no. of outliers as shown in Table 5.4. Prior has more outliers than later. Both have 43907 examples.

TABLE 6.108: Performance of MLFLD and MLFLD-MAXP for large datasets

(a) Hamming loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.0472	0.0544	0.0467	0.0434	**0.0331**	0.0638	0.0331	0.0336	0.0337
Mediamill	0.0412	0.0494	0.0424	0.0409	0.0318	0.0645	**0.0316**	0.0317	0.0318
Average	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0327	0.0328
Rank	6	8	7	5	2	9	1	3	4

(b) Ranking loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.2939	0.3865	0.2684	0.3487	0.0967	0.0633	**0.0604**	NaN	NaN
Mediamill	0.2142	0.3668	0.2116	0.3465	0.0853	0.0516	**0.0533**	NaN	NaN
Average	0.2541	0.3767	0.2400	0.3476	0.0910	0.0575	**0.0569**	NaN	NaN
Rank	5	7	4	6	3	2	1	NaN	NaN

(c) One Error (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.5836	0.7697	0.3734	0.4300	**0.2020**	0.2375	0.2043	0.2032	0.2032
Mediamill	0.5256	0.6951	0.3495	0.4236	0.1810	0.2205	**0.1804**	0.1809	0.1809
Average	0.5546	0.7324	0.3615	0.4268	0.1915	0.2290	0.1924	**0.1921**	**0.1921**
Rank	8	9	6	7	1	5	4	2	2

(d) Coverage (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	69.7428	66.8468	61.7521	67.2632	31.4247	21.2520	20.1887	**20.1426**	**20.1426**
Mediamill	58.7190	64.4715	53.4878	67.5235	28.8667	**18.2272**	18.8066	18.8340	18.8340
Average	64.2309	65.6592	57.6200	67.3934	30.1457	19.7396	19.4977	**19.4883**	**19.4883**
Rank	7	8	6	9	5	4	3	1	1

(e) Average Precision (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.4116	0.2169	0.4549	0.2725	0.6547	0.6366	**0.6738**	NaN	NaN
Mediamill	0.4880	0.2670	0.5156	0.2833	0.6850	0.6782	**0.7005**	NaN	NaN
Average	0.4498	0.2420	0.4853	0.2779	0.6699	0.6574	**0.6872**	NaN	NaN
Rank	5	7	4	6	2	3	1	NaN	NaN

(f) Accuracy (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.2981	0.2690	0.3032	0.1866	0.3899	0.3139	**0.4009**	NaN	0.3869
Mediamill	0.3477	0.3140	0.3545	0.2078	0.4176	0.3374	**0.4200**	NaN	0.4176
Average	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	NaN	0.4023
Rank	6	7	4	8	2	5	1	NaN	3

Table 6.109: Performance of MLFLD and MLFLD-MAXP for large datasets

				(g) Subset Accuracy (↑)					
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.0169	0.0302	0.0447	0.0098	**0.1032**	0.0000	0.1013	0.1026	0.0939
Mediamill	0.0339	0.0500	0.0661	0.0126	**0.1125**	0.0077	0.1074	0.1111	0.1050
Average	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.1069	0.0995
Rank	7	6	5	8	1	9	3	2	4
				(h) Ex-F1 (↑)					
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.4193	0.3729	0.4079	0.2857	0.4933	0.4542	**0.5083**	NaN	0.4947
Mediamill	0.4706	0.4231	0.4648	0.3139	0.5263	0.4766	**0.5317**	NaN	0.5298
Average	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	NaN	0.5123
Rank	5	7	6	8	3	4	1	NaN	2
				(i) Macro-F1 (↑)					
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.0742	0.0696	0.0705	0.0242	0.0618	0.0600	0.0939	0.1146	**0.1155**
Mediamill	**0.1349**	0.1102	0.1192	0.0361	0.1056	0.0933	0.1063	0.1128	0.1148
Average	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	0.1137	**0.1152**
Rank	3	6	5	9	7	8	4	2	1
				(j) Micro-F1 (↑)					
Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Cbmi09-bow	0.4343	0.3848	0.4247	0.2960	0.5033	0.4715	**0.5184**	0.5084	0.5118
Mediamill	0.4882	0.4352	0.4824	0.3259	0.5415	0.4935	0.5442	0.5433	**0.5450**
Average	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5259	0.5284
Rank	6	8	7	9	4	5	1	3	2

Observations: From Table 6.110(a), MLFLD and MLFLD-MAXP could not compute Ex-F1, avg precision, rank loss, and accuracy showed by NaN. When functioning for remaining metrics (not having NaN) is considered as shown in Table 6.110(b), then MLFLD outshined with the smallest average rank 2.2 among 9. MLFLD-MAXP also worked similarly.

Table 6.111 shows that no particular algorithm could improve all metrics. Proposed algorithms have enhanced coverage and macro-F1, while for 8 parameters, they functioned similarly to that of enhanced algorithms.

From Table 5.3 and 5.4, though there are 101 labels in both datasets, the cardinality of the dataset is only 4. Average and maximum no. of labels/example is only 4 and 17, respectively. The average no. of Ex/label is 4. It means that out of 43907, approx. 4 examples are associated with one label. It is reflected in Table 6.111(i) by the enhanced performance of MLFLD-MAXP and MLFLD by 15% and 13% resp. for macro-F1 that is more influenced by rare labels.

Compared to MLkNN, both algorithms worked better for one error and coverage, while MLFLD is better for subset accuracy.

Table 6.110: Summary of MLFLD and MLFLD-MAXP Performance for large datasets

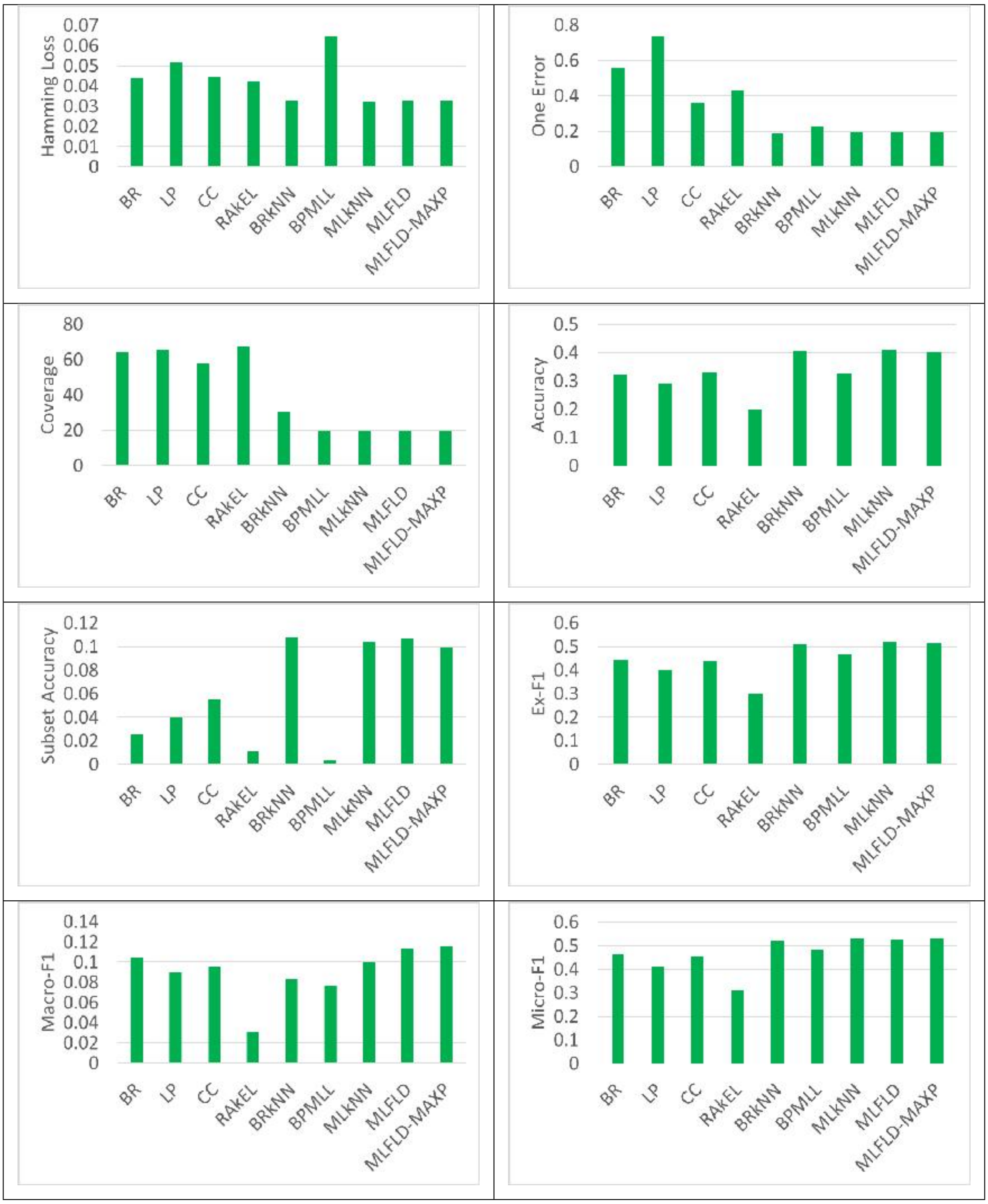
(a) Summary of 10 metrics

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
HamLoss	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0327	0.0328
RankLoss	0.2541	0.3767	0.2400	0.3476	0.0910	0.0575	**0.0569**	NaN	NaN
OneError	0.5546	0.7324	0.3615	0.4268	**0.1915**	0.2290	0.1924	0.1921	0.1921
Coverage	64.231	65.659	57.620	67.393	30.146	19.740	19.498	**19.488**	19.488
AvgPrec	0.4498	0.2420	0.4853	0.2779	0.6699	0.6574	**0.6872**	NaN	NaN
Accuracy	0.3229	0.2915	0.3289	0.1972	0.4038	0.3257	**0.4105**	NaN	0.4023
SubAcc	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.1069	0.0995
Ex-F1	0.4450	0.3980	0.4364	0.2998	0.5098	0.4654	**0.5200**	NaN	0.5123
Macro F1	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	0.1137	**0.1152**
Micro F1	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5259	0.5284
Avg Rank	5.8	7.3	5.4	7.5	3	5.4	**2**	2.2	2.4
#Wins	0	0	0	0	2	0	**6**	1	2

(b) Summary of 6 metrics without NaN

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
HamLoss	0.0442	0.0519	0.0446	0.0422	0.0325	0.0642	**0.0324**	0.0327	0.0328
OneError	0.5546	0.7324	0.3615	0.4268	**0.1915**	0.2290	0.1924	0.1921	0.1921
Coverage	64.231	65.659	57.620	67.393	30.146	19.740	19.498	**19.488**	**19.488**
SubAcc	0.0254	0.0401	0.0554	0.0112	**0.1079**	0.0039	0.1044	0.1069	0.0995
Macro F1	0.1046	0.0899	0.0949	0.0302	0.0837	0.0767	0.1001	0.1137	**0.1152**
Micro F1	0.4613	0.4100	0.4536	0.3110	0.5224	0.4825	**0.5313**	0.5259	0.5284
Avg Rank	6.2	7.5	6.0	7.8	3.3	6.7	2.7	**2.2**	**2.3**
#Wins	0	0	0	0	2	0	2	**1**	**2**

Table 6.111: Performance of proposed algorithms for large datasets



6.5.4 Effect of distance variation for feature similarities on the performance of proposed algorithms using Hamming distance for label dissimilarities for large datasets

By keeping Hamming distance measure the same for label dissimilarity, both algorithms are analyzed and compared with competing algorithm by varying feature similarity measures shown in Table 6.112 and 6.113.

TABLE 6.112: Effect of distance variation on MLFLD and MAXP for large datasets

(a) *Hamming loss* (↓)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	**0.0331**	0.0336	**0.0331**	0.0334	0.0337	0.0332	0.0335
Mediamill	**0.0316**	0.0317	0.0317	0.0317	0.0318	0.0317	0.0318
Average	**0.0324**	0.0327	0.0324	0.0326	0.0328	0.0325	0.0327
Rank	1	5	2	4	7	3	5

(b) *Ranking loss* (↓)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.0604	NaN	NaN	NaN	NaN	NaN	NaN
Mediamill	0.0533	NaN	NaN	NaN	NaN	NaN	NaN
Average	0.0569	NaN	NaN	NaN	NaN	NaN	NaN
Rank	1	-	-	-	-	-	-

(c) *One Error* (↓)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.2043	**0.2032**	0.2039	0.2084	**0.2032**	0.2039	0.2084
Mediamill	**0.1804**	0.1809	0.1805	0.1831	0.1809	0.1805	0.1831
Average	0.1924	**0.1921**	0.1922	0.1958	**0.1921**	0.1922	0.1958
Rank	5	1	3	6	1	3	6

(d) *Coverage* (↓)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	20.1887	20.1426	20.0584	**19.7938**	20.1426	20.0584	**19.7938**
Mediamill	**18.8066**	18.8340	18.8114	18.8683	18.8340	18.8114	18.8683
Average	19.4977	19.4883	19.4349	**19.3311**	19.4883	19.4349	**19.3311**
Rank	7	5	3	1	5	3	1

(e) *Average Precision* (↑)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.6738	NaN	NaN	NaN	NaN	NaN	NaN
Mediamill	0.7005	NaN	NaN	NaN	NaN	NaN	NaN
Average	0.6872	NaN	NaN	NaN	NaN	NaN	NaN

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.5184	0.5084	0.5174	0.5190	0.5118	0.5198	**0.5204**
Mediamill	0.5442	0.5433	0.5460	0.5416	0.5450	**0.5471**	0.5431
Average	0.5313	0.5259	0.5317	0.5303	0.5284	**0.5335**	0.5318
Rank	4	7	3	5	6	1	2

TABLE 6.113: Effect of distance variation on MLFLD and MAXP for large datasets

(f) Accuracy (↑)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.4009	NaN	NaN	NaN	0.3869	0.3969	0.3939
Mediamill	0.4200	NaN	NaN	NaN	0.4176	0.4194	0.4147
Average	0.4105	NaN	NaN	NaN	0.4023	0.4082	0.4043
Rank	1	-	-	-	4	2	3

(g) Subset Accuracy (↑)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.1013	0.1026	**0.1073**	0.1015	0.0939	0.0991	0.0933
Mediamill	0.1074	**0.1111**	0.1090	0.1070	0.1050	0.1042	0.1014
Average	0.1044	0.1069	**0.1082**	0.1043	0.0995	0.1017	**0.0974**
Rank	3	2	1	4	6	5	7

(h) Ex-F1 (↑)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	**0.5083**	NaN	NaN	NaN	0.4947	0.5045	0.5026
Mediamill	0.5317	NaN	NaN	NaN	0.5298	**0.5322**	0.5275
Average	**0.5200**	NaN	NaN	NaN	0.5123	0.5184	0.5151
Rank	1	-	-	-	4	2	3

(i) Macro-F1 (↑)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.0939	0.1146	0.1094	0.1231	0.1155	0.1098	**0.1236**
Mediamill	0.1063	0.1128	0.1150	0.1119	0.1148	**0.1165**	0.1132
Average	0.1001	0.1137	0.1122	0.1175	0.1152	0.1132	**0.1184**
Rank	7	4	6	2	3	5	1

(j) Micro-F1 (↑)

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Cbmi09-bow	0.5184	0.5084	0.5174	0.5190	0.5118	0.5198	**0.5204**
Mediamill	0.5442	0.5433	0.5460	0.5416	0.5450	**0.5471**	0.5431
Average	0.5313	0.5259	0.5317	0.5303	0.5284	**0.5335**	0.5318
Rank	4	7	3	5	6	1	2

TABLE 6.114: Summary of distance variation with proposed algorithms for large datasets

(a) Summary of 10 metrics

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	**0.0324**	0.0327	0.0324	0.0326	0.0328	0.0325	0.0327
RankLoss	**0.0569**	NaN	NaN	NaN	NaN	NaN	NaN
OneError	0.1924	**0.1921**	0.1922	0.1958	**0.1921**	0.1922	0.1958
Coverage	19.4977	19.4883	19.4349	**19.3311**	19.4883	19.4349	**19.3311**
AvgPrec	**0.6872**	NaN	NaN	NaN	NaN	NaN	NaN
Accuracy	**0.4105**	NaN	NaN	NaN	0.4023	0.4082	0.4043
SubAcc	0.1044	0.1069	**0.1082**	0.1043	0.0995	0.1017	0.0974
Ex-F1	**0.5200**	NaN	NaN	NaN	0.5123	0.5184	0.5151
Macro-F1	0.1001	0.1137	0.1122	0.1175	0.1152	0.1132	**0.1184**
Micro-F1	0.5313	0.5259	0.5317	0.5303	0.5284	**0.5335**	0.5318
ExecTime	638	1305	1242	1522	1291	1255	1502
Avg Rank	3.1	4	2.8	3.7	4.5	3	4
#Wins	5	1	2	1	1	1	2

(b) Summary of 6 metrics without NaN

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	**0.0324**	0.0327	0.0324	0.0326	0.0328	0.0325	0.0327
OneError	0.1924	**0.1921**	0.1922	0.1958	**0.1921**	0.1922	0.1958
Coverage	19.4977	19.4883	19.4349	**19.3311**	19.4883	19.4349	**19.3311**
SubAcc	0.1044	0.1069	**0.1082**	0.1043	0.0995	0.1017	0.0974
Macro-F1	0.1001	0.1137	0.1122	0.1175	0.1152	0.1132	**0.1184**
Micro-F1	0.5313	0.5259	0.5317	0.5303	0.5284	**0.5335**	0.5318
ExecTime	638	1305	1242	1522	1291	1255	1502
Avg Rank	4.5	4	2.8	3.7	4.7	3.3	4
#Wins	1	1	2	1	1	1	2

Observations: Table 6.114(a) shows that MLFLD using Manhattan distance for feature similarity has outperformed among all the six combinations and MLkNN. Even if only six measures (not showing NaN) are considered, then also the winner is the same for large datasets as shown in Table 6.114b). MLFLD could not compute 4 metrics shown by NaN, because of zero label instances.

All six variations of proposed algorithms have improved coverage and macro-F measures. Using Minkowski, better performance is obtained for coverage, subset accuracy, and macro-F1 than that with the remaining two distances. The use of Minkowski with

MLFLD-MAXP results in more improvement of macro-F1 than Euclidean. The time required by Manhattan, Euclidean and Minkowski experiments are twice, more than twice, and approx. 2.5 times as compared to that of MLkNN, respectively.

6.6 Effect of distance variation for label dissimilarity on the performance of proposed algorithms

In all previous sections, Hamming distance is used for label dissimilarity. In this section, the effect of using two other measures, namely Jaccard and SimIC distance, is observed.

6.6.1 Performance of proposed algorithms using Jaccard distance for label dissimilarity

Jaccard distance uses union and intersection operations for computation. It is used to compute label dissimilarity in this section.

6.6.1.1 Performance of MLFLD and MLFLD-MAXP (train test splits) using Jaccard distance for label dissimilarity to check the effect of distance variation for feature similarity

First, the performance of proposed algorithms using Jaccard distance is compared with that of a contesting algorithm. Proposed algorithms are evaluated using three distance measures for feature similarity in this section shown in Table 6.115 to 6.124.

Table 6.115: Effect of distance variation on Hamming Loss (↓) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	**0.2162**	0.2277	0.2186	0.2228	0.2285	0.2203	0.2236
Scene	0.0962	**0.0851**	0.0858	0.0892	0.0885	0.0917	0.0916
Image	0.1147	0.1127	0.1150	0.1140	0.1147	0.1177	**0.1103**
Yeast	0.2008	0.2068	**0.2001**	0.2078	0.2067	**0.2001**	0.2074
Arts Humanity	**0.0612**	0.0658	0.0658	0.0660	0.0817	0.0821	0.0826
Business Eco.	**0.0269**	0.0295	0.0298	0.0331	0.0296	0.0298	0.0326
Education	**0.0387**	0.0459	0.0455	0.0490	0.0595	0.0581	0.0620
Entertainment	**0.0604**	0.0689	0.0662	0.0755	0.0860	0.0850	0.0884
Health	**0.0458**	0.0549	0.0518	0.0561	0.0548	0.0518	0.0559
Reference	**0.0314**	0.0355	0.0355	0.0346	0.0370	0.0370	0.0389
Science	**0.0325**	0.0374	0.0368	0.0372	0.0512	0.0510	0.0509
Social Science	**0.0218**	0.0299	0.0303	0.0310	0.0329	0.0346	0.0349
Society Culture	**0.0537**	0.0592	0.0592	0.0626	0.0652	0.0624	0.0690
Average	**0.0769**	0.0815	0.0800	0.0830	0.0874	0.0863	0.0883
Rank	1	3	2	4	6	5	7

Table 6.116: Effect of distance variation on Ranking Loss (↓) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1781	**0.1664**	0.1683	0.1862	**0.1664**	0.1683	0.1862
Scene	0.0930	0.0826	0.0840	**0.0824**	0.0826	0.0840	**0.0824**
Image	0.1154	0.0926	0.0892	**0.0826**	0.0926	0.0892	**0.0826**
Yeast	**0.1766**	0.1822	0.1775	0.1791	0.1822	0.1775	0.1791
Arts Humanity	**0.1514**	0.1772	0.1759	0.1851	0.1772	0.1759	0.1851
Business Eco.	**0.0373**	0.0502	0.0481	0.0570	0.0502	0.0481	0.0570
Education	**0.0800**	0.1271	0.1190	0.1301	0.1271	0.1190	0.1301
Entertainment	**0.1151**	0.1696	0.1662	0.1666	0.1696	0.1662	0.1666
Health	**0.0605**	0.0835	0.0798	0.0894	0.0835	0.0798	0.0894
Reference	**0.0919**	0.1122	0.1120	0.1124	0.1122	0.1120	0.1124
Science	**0.1167**	0.1855	0.1812	0.1800	0.1855	0.1812	0.1800
Social Science	**0.0561**	0.0903	0.0849	0.0914	0.0903	0.0849	0.0914
Society Culture	**0.1338**	0.1654	0.1585	0.1705	0.1654	0.1585	0.1705
Average	**0.1081**	0.1296	0.1265	0.1318	0.1296	0.1265	0.1318
Rank	1	4	2	6	4	2	6

Table 6.117: Effect of distance variation on One Error (↓) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.3218	0.3218	0.3267	**0.3069**	0.3218	0.3267	**0.3069**
Scene	0.2425	0.2258	0.2316	**0.2241**	0.2258	0.2316	**0.2241**
Image	0.2517	0.2250	0.2150	**0.2017**	0.2250	0.2150	**0.2017**
Yeast	**0.2519**	0.2628	0.2650	0.2541	0.2628	0.2650	0.2541
Arts Humanity	**0.6330**	0.7387	0.7443	0.7443	0.7387	0.7443	0.7443
Business Eco.	**0.1213**	0.1470	0.1533	0.1777	0.1470	0.1533	0.1777
Education	**0.5207**	0.7490	0.7323	0.7743	0.7490	0.7323	0.7743
Entertainment	**0.5300**	0.6963	0.6967	0.7053	0.6963	0.6967	0.7053
Health	**0.4190**	0.4977	0.5080	0.5460	0.4977	0.5080	0.5460
Reference	**0.4730**	0.5220	0.5220	0.5507	0.5220	0.5220	0.5507
Science	**0.5810**	0.8010	0.8017	0.8017	0.8010	0.8017	0.8017
Social Science	**0.3270**	0.4947	0.5270	0.5343	0.4947	0.5270	0.5343
Society Culture	**0.4357**	0.5377	0.4993	0.5840	0.5377	0.4993	0.5840
Average	**0.3930**	0.4784	0.4787	0.4927	0.4784	0.4787	0.4927
Rank	1	2	4	6	2	4	6

Table 6.118: Effect of distance variation on Coverage (↓) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	1.9356	1.8416	**1.8366**	2.0050	1.8416	**1.8366**	2.0050
Scene	0.5661	**0.5151**	0.5234	0.5159	**0.5151**	0.5234	0.5159
Image	0.6083	0.5150	0.5033	**0.4750**	0.5150	0.5033	**0.4750**
Yeast	**6.4318**	6.5278	6.4482	6.4896	6.5278	6.4482	6.4896
Arts Humanity	**5.4313**	6.1297	6.0850	6.3130	6.1297	6.0850	6.3130
Business Eco.	**2.1840**	2.6870	2.5730	3.0040	2.6870	2.5730	3.0040
Education	**3.4973**	5.0587	4.8073	5.1657	5.0587	4.8073	5.1657
Entertainment	**3.1467**	4.3277	4.2607	4.2573	4.3277	4.2607	4.2573
Health	**3.3043**	4.1757	4.0703	4.3583	4.1757	4.0703	4.3583
Reference	**3.5420**	4.2063	4.2017	4.2207	4.2063	4.2017	4.2207
Science	**6.0470**	8.8140	8.6713	8.6090	8.8140	8.6713	8.6090
Social Science	**3.0340**	4.5370	4.2897	4.5687	4.5370	4.2897	4.5687
Society Culture	**5.3653**	6.2267	6.0667	6.3003	6.2267	6.0667	6.3003
Average	**3.4687**	4.2740	4.1798	4.3294	4.2740	4.1798	4.3294
Rank	1	4	2	6	4	2	6

TABLE 6.119: Effect of distance variation on Average Precision (↑) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.7810	**0.7911**	0.7892	0.7774	**0.7911**	0.7892	0.7774
Scene	0.8511	**0.8638**	0.8599	**0.8638**	**0.8638**	0.8599	**0.8638**
Image	0.8456	0.8676	0.8728	**0.8813**	0.8676	0.8728	**0.8813**
Yeast	**0.7505**	0.7445	0.7486	0.7484	0.7445	0.7486	0.7484
Arts Humanity	**0.5097**	0.4348	0.4370	0.4189	0.4348	0.4370	0.4189
Business Eco.	**0.8798**	0.8405	0.8460	0.8171	0.8405	0.8460	0.8171
Education	**0.5993**	0.4183	0.4336	0.3969	0.4183	0.4336	0.3969
Entertainment	**0.6013**	0.4643	0.4662	0.4596	0.4643	0.4662	0.4596
Health	**0.6817**	0.5913	0.6013	0.5641	0.5913	0.6013	0.5641
Reference	**0.6194**	0.5574	0.5581	0.5439	0.5574	0.5581	0.5439
Science	**0.5324**	0.3432	0.3471	0.3386	0.3432	0.3471	0.3386
Social Science	**0.7481**	0.6131	0.6023	0.5902	0.6131	0.6023	0.5902
Society Culture	**0.6128**	0.5370	0.5567	0.5062	0.5370	0.5567	0.5062
Average	**0.6933**	0.6205	0.6245	0.6082	0.6205	0.6245	0.6082
Rank	1	4	2	6	4	2	6

TABLE 6.120: Effect of distance variation on Accuracy (↑) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.4818	0.4827	0.5202	0.4917	0.5000	**0.5301**	0.5066
Scene	0.6597	0.6958	0.6950	0.6848	**0.7397**	0.7301	0.7312
Image	0.6492	0.7008	0.7042	0.7103	0.7325	0.7258	**0.7444**
Yeast	0.4998	0.4810	**0.5017**	0.4918	0.4832	**0.5017**	0.4943
Arts Humanity	0.0331	0.0634	0.0459	0.0545	**0.1993**	0.1942	0.1915
Business Eco.	**0.6967**	0.6704	0.6736	0.6198	0.6705	0.6736	0.6465
Education	0.1560	0.0555	0.0641	0.0884	0.1901	**0.2055**	0.1656
Entertainment	0.1862	0.1091	0.1079	0.1401	**0.2528**	0.2478	0.2360
Health	0.3390	0.3845	0.3539	0.1669	**0.3876**	0.3755	0.3193
Reference	0.1032	0.0389	0.0346	0.1970	0.4285	**0.4290**	0.4040
Science	0.0695	0.0462	0.0420	0.0331	**0.1655**	0.1626	0.1516
Social Science	0.2996	0.2186	0.2191	0.1614	**0.4542**	0.4265	0.4155
Society Culture	0.2431	0.1804	0.1703	0.1234	0.3466	**0.3739**	0.3089
Average	0.3398	0.3175	0.3179	0.3049	0.4270	**0.4289**	0.4089
Rank	4	6	5	7	2	1	3

Table 6.121: Effect of distance variation on Subset Accuracy (↑) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2178	0.2426	0.2921	0.2426	0.2475	**0.2970**	0.2426
Scene	0.6012	0.6463	0.6430	0.6321	**0.6873**	0.6756	0.6756
Image	0.5983	0.6350	0.6333	0.6333	**0.6650**	0.6533	0.6650
Yeast	0.1647	0.1788	0.1788	**0.1941**	0.1788	0.1788	**0.1941**
Arts Humanity	0.0277	0.0510	0.0377	0.0420	**0.1543**	0.1523	0.1447
Business Eco.	**0.5353**	0.5060	0.5127	0.4680	0.5060	0.5127	0.4883
Education	0.1310	0.0350	0.0500	0.0557	0.1360	**0.1540**	0.1083
Entertainment	0.1687	0.0830	0.0963	0.1097	0.2033	**0.2043**	0.1787
Health	0.2403	0.2700	0.2503	0.0887	**0.2717**	0.2670	0.2060
Reference	0.0963	0.0353	0.0313	0.1783	0.3827	**0.3833**	0.3617
Science	0.0603	0.0367	0.0350	0.0277	**0.1353**	0.1337	0.1173
Social Science	0.2700	0.2000	0.1993	0.1490	**0.4097**	0.3837	0.3733
Society Culture	0.2010	0.1457	0.1387	0.1017	0.2663	**0.2867**	0.2317
Average	4	6	5	7	2	1	3
Rank	4	6	5	7	2	1	3

Table 6.122: Effect of distance variation on Ex-F1 (↑) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5662	0.5569	0.5917	0.5691	0.5792	**0.6033**	0.5897
Scene	0.6793	0.7124	0.7124	0.7025	**0.7572**	0.7483	0.7499
Image	0.6667	0.7233	0.7283	0.7369	0.7555	0.7505	**0.7719**
Yeast	**0.6067**	0.5816	0.6060	0.5912	0.5848	0.6060	0.5952
Arts Humanity	0.0352	0.0683	0.0491	0.0592	**0.2173**	0.2111	0.2097
Business Eco.	**0.7546**	0.7298	0.7324	0.6754	0.7300	0.7325	0.7046
Education	0.1647	0.0627	0.0692	0.1004	0.2101	**0.2246**	0.1868
Entertainment	0.1924	0.1185	0.1122	0.1511	**0.2709**	0.2640	0.2571
Health	0.3772	0.4291	0.3937	0.1978	**0.4327**	0.4172	0.3640
Reference	0.1055	0.0401	0.0357	0.2035	0.4443	**0.4448**	0.4188
Science	0.0728	0.0498	0.0446	0.0350	**0.1769**	0.1736	0.1650
Social Science	0.3100	0.2253	0.2263	0.1658	**0.4703**	0.4420	0.4308
Society Culture	0.2594	0.1941	0.1827	0.1321	0.3782	**0.4083**	0.3394
Average	0.3685	0.3455	0.3449	0.3323	0.4621	**0.4636**	0.4448
Rank	4	5	6	7	2	1	3

Table 6.123: Effect of distance variation on Macro-F1 (↑) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5880	0.6099	0.6424	0.6306	0.6152	**0.6425**	0.6377
Scene	0.7156	0.7518	0.7509	0.7426	**0.7590**	0.7505	0.7540
Image	0.5904	0.6104	0.6268	0.6280	0.6209	0.6321	**0.6347**
Yeast	0.3444	0.3878	0.3789	0.3968	0.3884	0.3789	**0.3973**
Arts Humanity	0.0343	0.0293	0.0226	0.0405	0.0674	0.0623	**0.0802**
Business Eco.	0.1817	NaN	NaN	NaN	NaN	NaN	NaN
Education	0.1421	NaN	NaN	NaN	NaN	NaN	NaN
Entertainment	0.1271	0.1131	0.1016	0.1130	**0.1421**	0.1372	0.1387
Health	0.1567	NaN	NaN	NaN	NaN	NaN	NaN
Reference	0.0907	NaN	NaN	NaN	NaN	NaN	NaN
Science	0.0408	0.0179	0.0159	0.0148	**0.0470**	0.0414	0.0459
Social Science	0.1175	NaN	NaN	NaN	NaN	NaN	NaN
Society Culture	0.0714	0.0408	0.0375	0.0443	**0.0730**	0.0595	0.0722
Average	0.2462	0.3201	0.3221	0.3263	0.3391	0.3381	**0.3451**
Rank	7	6	5	4	2	3	1

Table 6.124: Effect of distance variation on Micro-F1 (↑) with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6278	0.6209	0.6414	0.6322	0.6272	**0.6426**	0.6382
Scene	0.7156	0.7474	0.7461	0.7373	**0.7515**	0.7428	0.7445
Image	0.7166	0.7412	0.7392	0.7463	0.7456	0.7403	**0.7603**
Yeast	0.6303	0.6154	0.6340	0.6207	0.6163	**0.6340**	0.6223
Arts Humanity	0.0480	0.0871	0.0621	0.0847	**0.2005**	0.1937	0.1986
Business Eco.	**0.6990**	0.6766	0.6736	0.6333	0.6761	0.6737	0.6461
Education	**0.2541**	0.1158	0.1214	0.1561	0.2205	0.2338	0.2013
Entertainment	0.2696	0.1984	0.1815	0.2097	**0.2815**	0.2732	0.2689
Health	0.4033	0.4088	0.3827	0.2501	**0.4108**	0.3954	0.3593
Reference	0.1652	0.0679	0.0609	0.2763	0.4393	**0.4397**	0.4129
Science	0.1063	0.0727	0.0647	0.0542	**0.1666**	0.1645	0.1656
Social Science	0.3865	0.2940	0.2920	0.2279	**0.4423**	0.4147	0.4080
Society Culture	0.2896	0.2244	0.2131	0.1570	0.3453	**0.3735**	0.3120
Average	0.4086	0.3747	0.3702	0.3681	**0.4557**	0.4555	0.4414
Rank	4	5	6	7	1	2	3

Table 6.125: Summary of effect of distance variation on MLFLD and MLFLD-MAXP performance with Jaccard distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	**0.0769**	0.0815	0.0800	0.0830	0.0874	0.0863	0.0883
RankLoss	**0.1081**	0.1296	0.1265	0.1318	0.1296	0.1265	0.1318
OneError	**0.3930**	0.4784	0.4787	0.4927	0.4784	0.4787	0.4927
Coverage	**3.4687**	4.2740	4.1798	4.3294	4.2740	4.1798	4.3294
AvgPrec	**0.6933**	0.6205	0.6245	0.6082	0.6205	0.6245	0.6082
Accuracy	0.3398	0.3175	0.3179	0.3049	0.4270	**0.4289**	0.4089
SubAcc	0.2548	0.2358	0.2383	0.2248	0.3265	**0.3294**	0.3067
Ex-F1	0.3685	0.3455	0.3449	0.3323	0.4621	**0.4636**	0.4448
Macro-F1	0.2462	0.3201	0.3221	0.3263	0.3391	0.3381	**0.3451**
Micro-F1	0.4086	0.3747	0.3702	0.3681	**0.4557**	0.4555	0.4414
ExecTime	6	32	35	112	32	31	115
Avg Rank	2.8	4.5	3.9	6	2.9	**2.3**	4.4
#Wins	5	0	0	0	1	**3**	1

Observations: Table 6.125 for train-test splits of datasets shows that MLFLD-MAXP with Manhattan distance for feature similarity has outperformed among six experimentations carried out using proposed algorithms with Jaccard distance for label dissimilarity and competing algorithm MLkNN. It has shown a minimum average rank among all. MLFLD-MAXP has enhanced for the last five metrics, but it could not be better in the first five metrics. Calibration of a threshold for individual datasets may help to improve the performance of these measures.

MLFLD-MAXP with Manhattan has raised performance by 29% for subset accuracy, 25% for accuracy and ex-F1, 11% for micro-F, and 7% for macro-F compared to MLkNN. Both algorithms could not compute macro-F for 5 datasets.

All MLFLD variations are noticed to function better than all MLFLD-MAXP variations to reduce misclassifications, but could not beat MLkNN.

For one error, coverage, rank loss, and avg precision, both proposed algorithms have similar performance when the same distance is used for feature similarity.

6.6.1.2 Effect of distance variation for feature similarity on MLFLD and MLFLD-MAXP Performance (cross-validation) using Jaccard distance for label dissimilarity

In this section, keeping label dissimilarity distance the same, feature similarity measures are varied to observe the performance of MLFLD and MLFLD-MAXP and compared with MLkNN shown in Table 6.126 to 6.135.

TABLE 6.126: Effect of distance variation on Hamming Loss (↓) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1959	0.1989	**0.1898**	0.1912	0.1952	0.1904	0.1918
Image	0.1690	0.1632	**0.1612**	0.1629	0.1661	0.1661	0.1663
Scene	0.0861	0.0795	**0.0786**	0.0796	0.0811	0.0801	0.0810
Yeast	0.1940	0.1967	0.1939	0.1975	0.1961	**0.1938**	0.1976
CAL500	0.1388	0.1393	0.1395	**0.1387**	0.1393	0.1395	**0.1387**
Average	0.1568	0.1555	**0.1526**	0.1540	0.1556	0.1540	0.1551
Rank	7	5	1	2	6	2	4

TABLE 6.127: Effect of distance variation on Ranking Loss (↓) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1594	0.1547	**0.1488**	0.1551	0.1547	**0.1488**	0.1551
Image	0.1680	0.1569	**0.1567**	0.1586	0.1569	**0.1567**	0.1586
Scene	0.0775	0.0689	0.0679	**0.0664**	0.0689	0.0679	**0.0664**
Yeast	0.1670	0.1688	**0.1662**	0.1732	0.1688	**0.1662**	0.1732
CAL500	**0.1828**	0.1836	0.1835	0.1834	0.1836	0.1835	0.1834
Average	0.1509	0.1466	**0.1446**	0.1473	0.1466	**0.1446**	0.1473
Rank	7	3	1	5	3	1	5

Table 6.128: Effect of distance variation on One Error (↓) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2699	0.2508	0.2576	**0.2441**	0.2508	0.2576	**0.2441**
Image	0.3000	0.2916	**0.2866**	0.2876	0.2916	**0.2866**	0.2876
Scene	0.2256	0.2050	**0.2017**	**0.2017**	0.2050	**0.2017**	**0.2017**
Yeast	**0.2300**	0.2311	0.2320	0.2440	0.2311	0.2320	0.2440
CAL500	0.1176	**0.1140**	0.1160	0.1260	**0.1140**	0.1160	0.1260
Average	0.2286	**0.2185**	0.2188	0.2207	**0.2185**	0.2188	0.2207
Rank	7	1	3	5	1	3	5

Table 6.129: Effect of distance variation on Coverage (↓) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	1.7764	1.7542	**1.7136**	1.7559	1.7542	**1.7136**	1.7559
Image	0.9390	**0.8964**	0.8979	0.9030	**0.8964**	0.8979	0.9030
Scene	0.4753	0.4288	0.4242	**0.4146**	0.4288	0.4242	**0.4146**
Yeast	6.2750	6.3183	**6.2631**	6.3432	6.3183	**6.2631**	6.3432
CAL500	130.564	130.5120	**130.370**	130.5020	130.5120	**130.370**	130.502
Average	28.006	27.9819	**27.9338**	27.9837	27.9819	**27.9338**	27.9837
Rank	7	3	1	5	3	1	5

Table 6.130: Effect of distance variation on Average Precision (↑) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.8034	0.8094	**0.8136**	0.8129	0.8094	**0.8136**	0.8129
Image	0.8030	0.8106	**0.8120**	0.8115	0.8106	**0.8120**	0.8115
Scene	0.8652	0.8785	0.8804	**0.8819**	0.8785	0.8804	**0.8819**
Yeast	0.7650	0.7663	**0.7682**	0.7611	0.7663	**0.7682**	0.7611
CAL500	**0.4942**	0.4927	0.4918	0.4914	0.4927	0.4918	0.4914
Average	0.7462	0.7515	**0.7532**	0.7518	0.7515	**0.7532**	0.7518
Rank	7	5	1	3	5	1	3

Table 6.131: Effect of distance variation on Accuracy (↑) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5340	0.5158	0.5494	0.5516	0.5463	**0.5694**	0.5648
Image	0.4937	0.5709	0.5735	0.5837	**0.6187**	0.6157	0.6174
Scene	0.6635	0.7194	0.7122	0.7276	0.7604	**0.7631**	0.7628
Yeast	0.5162	0.5172	0.5166	0.5129	**0.5195**	0.5178	0.5141
CAL500	**0.1972**	0.1951	0.1960	0.1967	0.1951	0.1960	0.1967
Average	0.4809	0.5037	0.5095	0.5145	0.5280	**0.5324**	0.5312
Rank	7	6	5	4	3	1	2

Table 6.132: Effect of distance variation on Subset Accuracy (↑) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2934	0.2915	0.3119	0.2949	0.3017	**0.3237**	0.3000
Image	0.4090	0.4657	0.4703	0.4642	**0.5063**	0.5038	0.4943
Scene	0.6248	0.6758	0.6721	0.6742	0.7150	**0.7196**	0.7079
Yeast	0.1874	0.2033	0.2012	0.1975	**0.2037**	0.2017	0.1975
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3029	0.3273	0.3311	0.3262	0.3453	**0.3498**	0.3399
Rank	7	5	4	6	2	1	3

Table 6.133: Effect of distance variation on Ex-F1 (↑) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6141	0.5901	0.6263	0.6344	0.6279	0.6491	**0.6505**
Image	0.5223	0.6070	0.6089	0.6243	0.6572	0.6540	**0.6593**
Scene	0.6764	0.7340	0.7257	0.7456	0.7756	0.7776	**0.7813**
Yeast	**0.6204**	0.6165	0.6171	0.6121	0.6201	0.6187	0.6139
CAL500	**0.3240**	0.3212	0.3225	0.3237	0.3212	0.3225	0.3237
Average	0.5514	0.5738	0.5801	0.5880	0.6004	0.6044	**0.6057**
Rank	7	6	5	4	3	2	1

Table 6.134: Effect of distance variation on Macro-F1 (↑) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6226	0.6399	0.6613	0.6680	0.6534	0.6642	**0.6710**
Image	0.5815	0.6358	0.6396	0.6483	0.6507	0.6498	**0.6554**
Scene	0.7364	0.7718	0.7711	0.7759	0.7789	0.7815	**0.7837**
Yeast	0.3853	NaN	NaN	NaN	NaN	NaN	NaN
CAL500	0.1714	NaN	NaN	NaN	NaN	NaN	NaN
Average	0.4994	0.6825	0.6907	0.6974	0.6943	0.6985	**0.7034**
Rank	7	6	5	3	4	2	1

Table 6.135: Effect of distance variation on Micro-F1 (↑) with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6610	0.6476	0.6732	0.6787	0.6633	0.6784	**0.6821**
Image	0.5842	0.6346	0.6368	0.6453	0.6483	0.6463	**0.6521**
Scene	0.7332	0.7641	0.7634	0.7687	0.7702	0.7723	**0.7736**
Yeast	0.6471	0.6477	0.6479	0.6428	**0.6492**	0.6484	0.6432
CAL500	**0.3209**	0.3182	0.3197	0.3200	0.3182	0.3197	0.3200
Average	0.5893	0.6024	0.6082	0.6111	0.6098	0.6130	**0.6142**
Rank	7	6	5	3	4	2	1

Table 6.136: Summary of effect of distance variation on MLFLD and MLFLD-MAXP performance with Jaccard distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	0.1568	0.1555	**0.1526**	0.1540	0.1556	0.1540	0.1551
RankLoss	0.1509	0.1466	**0.1446**	0.1473	0.1466	**0.1446**	0.1473
OneError	0.2286	**0.2185**	0.2188	0.2207	**0.2185**	0.2188	0.2207
Coverage	28.006	27.9819	**27.9338**	27.9837	27.9819	**27.9338**	27.9837
AvgPrec	0.7462	0.7515	**0.7532**	0.7518	0.7515	**0.7532**	0.7518
Accuracy	0.4809	0.5037	0.5095	0.5145	0.5280	**0.5324**	0.5312
SubAcc	0.3029	0.3273	0.3311	0.3262	0.3453	**0.3498**	0.3399
Ex-F1	0.5514	0.5738	0.5801	0.5880	0.6004	0.6044	**0.6057**
Macro-F1	0.4994	0.6825	0.6907	0.6974	0.6943	0.6985	**0.7034**
Micro-F1	0.5893	0.6024	0.6082	0.6111	0.6098	0.6130	**0.6142**
ExecTime	17	62	64	72	52	58	81
Avg Rank	7	4.6	3.1	4	3.4	**1.6**	3
#Wins	0	1	4	0	1	**5**	3

Observations: From Table 6.136, the use of Manhattan distance for feature similarity has elevated the performance of MLFLD-MAXP when Jaccard distance is used for label dissimilarity. Also, for accuracy, subset accuracy, example-based F measure (Ex-F1), macro-F1, and micro-F1, MLFLD-MAXP performance is improved than MLFLD, whereas Manhattan with MLFLD-MAXP has outperformed among all. It is noticed that MLFLD-MAXP with Minkowski and Jaccard distance works better on all harmonic mean measures. Performance of both proposed algorithms is the same for one error, coverage, rank loss, and avg precision with 1-5% improvement, except ham loss for which MLFLD seems better with approx. 1% improvement. A rise in both accuracies, ex-F1 and macro-F1 are between 8-15% while it is 3-4% for micro-F1.

6.6.2 Performance of proposed algorithms using SimIC distance for label dissimilarity

The Similarity of Information Content (SimIC) is the proposed distance measure that is inspired by SimGIC distance. It is described in chapter 4. It is used to measure label dissimilarity, and its performance is compared with that of Jaccard and Hamming distance.

6.6.2.1 Performance of MLFLD and MLFLD-MAXP (train-test splits) using SimIC distance for label dissimilarity to check the effect of distance variation for feature similarity

This section has analyzed the performance of using SimIC distance with MLFLD and MLFLD-MAXP on train-test (TrTe) splits by varying feature similarity distance measures shown in Table 6.137 to 6.146.

TABLE 6.137: Effect of distance variation on Hamming Loss (↓) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	**0.2162**	0.2376	0.2285	0.2277	0.2417	0.2277	0.2244
Scene	0.0962	**0.0851**	0.0858	0.0892	0.0888	0.0911	0.0913
Image	0.1147	0.1147	0.1150	0.1140	0.1157	0.1177	**0.1103**
Yeast	**0.2008**	0.2134	0.2091	0.2119	0.2135	0.2084	0.2118
Arts Humanity	**0.0612**	0.0661	0.0661	0.0675	0.0816	0.0822	0.0845
Business Eco.	**0.0269**	0.0317	0.0312	0.0332	0.0307	0.0305	0.0346
Education	**0.0387**	0.0483	0.0481	0.0509	0.0608	0.0598	0.0619
Entertainment	**0.0604**	0.0778	0.0730	0.0775	0.0869	0.0854	0.0912
Health	**0.0458**	0.0548	0.0524	0.0574	0.0547	0.0522	0.0572
Reference	**0.0314**	0.0354	0.0353	0.0356	0.0584	0.0594	0.0506
Science	**0.0325**	0.0374	0.0372	0.0386	0.0517	0.0519	0.0525
Social Science	**0.0218**	0.0329	0.0303	0.0309	0.0366	0.0347	0.0370
Society Culture	**0.0537**	0.0630	0.0596	0.0655	0.0712	0.0689	0.0766
Average	**0.0769**	0.0845	0.0824	0.0846	0.0917	0.0900	0.0911
Rank	1	3	2	4	7	5	6

TABLE 6.138: Effect of distance variation on Ranking Loss (↓) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1781	0.1737	**0.1722**	0.1815	0.1737	**0.1722**	0.1815
Scene	0.0930	0.0849	0.0855	**0.0815**	0.0849	0.0855	**0.0815**
Image	0.1154	0.0882	0.0889	0.0831	0.0882	0.0889	0.0831
Yeast	**0.1766**	0.1892	0.1818	0.1893	0.1892	0.1818	0.1893
Arts Humanity	**0.1514**	0.1904	0.1837	0.2056	0.1904	0.1837	0.2056
Business Eco.	**0.0373**	0.0516	0.0498	0.0608	0.0516	0.0498	0.0608
Education	**0.0800**	0.1451	0.1289	0.1364	0.1451	0.1289	0.1364
Entertainment	**0.1151**	0.2345	0.2297	0.1953	0.2345	0.2297	0.1953
Health	**0.0605**	0.0851	0.0821	0.1007	0.0851	0.0821	0.1007
Reference	**0.0919**	0.3026	0.2918	0.2123	0.3026	0.2918	0.2123
Science	**0.1167**	0.2406	0.2093	0.2418	0.2406	0.2093	0.2418
Social Science	**0.0561**	0.1106	0.0937	0.1126	0.1106	0.0937	0.1126
Society Culture	**0.1338**	0.1978	0.1760	0.2112	0.1978	0.1760	0.2112
Average	**0.1081**	0.1611	0.1518	0.1548	0.1611	0.1518	0.1548
Rank	1	6	2	4	6	2	4

TABLE 6.139: Effect of distance variation on One Error (↓) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.3218	0.3317	0.3267	**0.3020**	0.3317	0.3267	**0.3020**
Scene	0.2425	0.2283	0.2316	**0.2232**	0.2283	0.2316	**0.2232**
Image	0.2517	0.2167	0.2167	**0.1950**	0.2167	0.2167	**0.1950**
Yeast	**0.2519**	0.2781	0.2683	0.2694	0.2781	0.2683	0.2694
Arts Humanity	**0.6330**	0.7370	0.7453	0.7647	0.7370	0.7453	0.7647
Business Eco.	**0.1213**	0.1550	0.1617	0.2043	0.1550	0.1617	0.2043
Education	**0.5207**	0.7553	0.7417	0.7573	0.7553	0.7417	0.7573
Entertainment	0.5300	0.6893	0.6980	0.7223	0.6893	0.6980	0.7223
Health	**0.4190**	0.4983	0.5087	0.5473	0.4983	0.5087	0.5473
Reference	**0.4730**	0.8740	0.8917	0.7417	0.8740	0.8917	0.7417
Science	**0.5810**	0.8120	0.8147	0.8293	0.8120	0.8147	0.8293
Social Science	**0.3270**	0.5527	0.5270	0.5720	0.5527	0.5270	0.5720
Society Culture	**0.4357**	0.6090	0.5857	0.6737	0.6090	0.5857	0.6737
Average	**0.3930**	0.5183	0.5168	0.5232	0.5183	0.5168	0.5232
Rank	1	4	2	6	4	2	6

TABLE 6.140: Effect of distance variation on Coverage (↓) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	1.9356	**1.8762**	**1.8762**	1.9752	**1.8762**	**1.8762**	1.9752
Scene	0.5661	0.5268	0.5309	**0.5100**	0.5268	0.5309	**0.5100**
Image	0.6083	0.4967	0.5017	**0.4783**	0.4967	0.5017	**0.4783**
Yeast	**6.4318**	6.5758	6.5278	6.6467	6.5758	6.5278	6.6467
Arts Humanity	**5.4313**	6.5490	6.3027	6.9090	6.5490	6.3027	6.9090
Business Eco.	**2.1840**	2.7227	2.6203	3.0327	2.7227	2.6203	3.0327
Education	**3.4973**	5.8527	5.2007	5.4610	5.8527	5.2007	5.4610
Entertainment	**3.1467**	5.5913	5.4767	4.8280	5.5913	5.4767	4.8280
Health	**3.3043**	4.2040	4.1017	4.7567	4.2040	4.1017	4.7567
Reference	**3.5420**	10.4083	10.0433	7.4393	10.4083	10.0433	7.4393
Science	**6.0470**	11.1503	9.7957	11.0490	11.1503	9.7957	11.0490
Social Science	**3.0340**	5.3880	4.7050	5.5003	5.3880	4.7050	5.5003
Society Culture	**5.3653**	7.0357	6.5067	7.3437	7.0357	6.5067	7.3437
Average	**3.4687**	5.2598	4.9376	5.0715	5.2598	4.9376	5.0715
Rank	1	6	2	4	6	2	4

Table 6.141: Effect of distance variation on Average Precision (↑) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.7810	**0.7853**	0.7828	0.7824	**0.7853**	0.7828	0.7824
Scene	0.8511	0.8615	0.8592	**0.8647**	0.8615	0.8592	**0.8647**
Image	0.8456	0.8730	0.8722	**0.8836**	0.8730	0.8722	**0.8836**
Yeast	**0.7505**	0.7338	0.7451	0.7369	0.7338	0.7451	0.7369
Arts Humanity	**0.5097**	0.4108	0.4196	0.3960	0.4108	0.4196	0.3960
Business Eco.	**0.8798**	0.8350	0.8420	0.7946	0.8350	0.8420	0.7946
Education	**0.5993**	0.4015	0.4177	0.4025	0.4015	0.4177	0.4025
Entertainment	**0.6013**	0.4294	0.4256	0.4365	0.4294	0.4256	0.4365
Health	**0.6817**	0.5852	0.5884	0.5419	0.5852	0.5884	0.5419
Reference	**0.6194**	0.2334	0.2274	0.3629	0.2334	0.2274	0.3629
Science	**0.5324**	0.3083	0.3208	0.2844	0.3083	0.3208	0.2844
Social Science	**0.7481**	0.5563	0.5896	0.5501	0.5563	0.5896	0.5501
Society Culture	**0.6128**	0.4677	0.5001	0.4282	0.4677	0.5001	0.4282
Average	**0.6933**	0.5755	0.5839	0.5742	0.5755	0.5839	0.5742
Rank	1	4	2	6	4	2	6

Table 6.142: Effect of distance variation on Accuracy (↑) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.4818	0.4909	0.4967	0.4707	0.4934	**0.5165**	0.5087
Scene	0.6597	0.6958	0.6950	0.6848	**0.7389**	0.7317	0.7320
Image	0.6492	0.6928	0.7042	0.7103	0.7303	0.7258	**0.7444**
Yeast	**0.4998**	0.4778	0.4709	0.4747	0.4789	0.4738	0.4782
Arts Humanity	0.0331	0.0610	0.0481	0.0493	**0.2008**	0.1936	0.1708
Business Eco.	**0.6967**	0.6335	0.6397	0.6196	0.6627	0.6657	0.6232
Education	0.1560	0.0742	0.0748	0.1043	0.1880	**0.1998**	0.1834
Entertainment	0.1862	0.1626	0.1446	0.1652	**0.2556**	0.2476	0.2346
Health	0.3390	0.3850	0.3608	0.1985	**0.3884**	0.3725	0.3191
Reference	0.1032	0.0374	0.0326	0.1174	0.1108	0.0951	**0.2312**
Science	0.0695	0.0462	0.0414	0.0400	**0.1523**	0.1463	0.1353
Social Science	0.2996	0.2215	0.2193	0.2451	0.3990	**0.4258**	0.3813
Society Culture	0.2431	0.1918	0.1707	0.1296	0.2892	**0.3082**	0.2385
Average	0.3398	0.3208	0.3153	0.3084	0.3914	**0.3925**	0.3831
Rank	4	5	6	7	2	1	3

TABLE 6.143: Effect of distance variation on Subset Accuracy (↑) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2178	0.2376	0.2624	0.2376	0.2376	**0.2673**	0.2624
Scene	0.6012	0.6463	0.6430	0.6321	**0.6865**	0.6773	0.6764
Image	0.5983	0.6317	0.6333	0.6333	**0.6667**	0.6533	0.6650
Yeast	0.1647	0.1788	0.1723	**0.1799**	0.1788	0.1723	0.1799
Arts Humanity	0.0277	0.0490	0.0397	0.0353	**0.1553**	0.1517	0.1240
Business Eco.	**0.5353**	0.4763	0.4927	0.4673	0.4973	0.5113	0.4690
Education	0.1310	0.0467	0.0467	0.0667	0.1300	**0.1423**	0.1243
Entertainment	0.1687	0.1267	0.1267	0.1277	0.2013	**0.2027**	0.1823
Health	0.2403	0.2693	0.2527	0.0997	**0.2713**	0.2603	0.1933
Reference	0.0963	0.0330	0.0287	0.1053	0.0967	0.0827	**0.2077**
Science	0.0603	0.0367	0.0333	0.0323	**0.1197**	0.1140	0.1063
Social Science	0.2700	0.1920	0.1993	0.2240	0.3533	**0.3830**	0.3413
Society Culture	0.2010	0.1497	0.1370	0.1007	0.2153	**0.2347**	0.1727
Average	0.2548	0.2364	0.2360	0.2263	0.2931	**0.2964**	0.2850
Rank	4	5	6	7	2	1	3

TABLE 6.144: Effect of distance variation on Ex-F1 (↑) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5662	0.5729	0.5701	0.5464	0.5762	**0.5949**	0.5893
Scene	0.6793	0.7124	0.7124	0.7025	**0.7564**	0.7500	0.7507
Image	0.6667	0.7136	0.7283	0.7369	0.7519	0.7505	**0.7719**
Yeast	**0.6067**	0.5771	0.5702	0.5731	0.5790	0.5749	0.5782
Arts Humanity	0.0352	0.0657	0.0514	0.0549	**0.2190**	0.2104	0.1894
Business Eco.	**0.7546**	0.6900	0.6928	0.6754	0.7223	0.7216	0.6798
Education	0.1647	0.0838	0.0847	0.1178	0.2091	**0.2208**	0.2047
Entertainment	0.1924	0.1757	0.1512	0.1795	**0.2756**	0.2642	0.2545
Health	0.3772	0.4297	0.4023	0.2375	**0.4337**	0.4154	0.3687
Reference	0.1055	0.0389	0.0339	0.1216	0.1158	0.0994	**0.2396**
Science	0.0728	0.0498	0.0444	0.0428	**0.1646**	0.1585	0.1462
Social Science	0.3100	0.2321	0.2265	0.2527	0.4154	**0.4413**	0.3957
Society Culture	0.2594	0.2084	0.1839	0.1410	0.3184	**0.3373**	0.2642
Average	0.3685	0.3500	0.3425	0.3371	0.4260	**0.4261**	0.4179
Rank	4	5	6	7	2	1	3

Table 6.145: Effect of distance variation on Macro-F1 (↑) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5880	0.6187	0.6116	0.6015	0.6171	0.6194	**0.6245**
Scene	0.7156	0.7518	0.7509	0.7426	**0.7578**	0.7525	0.7548
Image	0.5904	0.5970	0.6268	0.6280	0.6132	0.6321	**0.6345**
Yeast	0.3444	0.3991	0.3844	0.3864	**0.3996**	0.3854	0.3875
Arts Humanity	0.0343	0.0273	0.0232	0.0375	0.0708	0.0630	**0.0768**
Business Eco.	0.1817	NaN	NaN	NaN	NaN	NaN	NaN
Education	0.1421	NaN	NaN	NaN	NaN	NaN	NaN
Entertainment	0.1271	0.1312	0.1092	0.1236	**0.1456**	0.1377	0.1409
Health	0.1567	NaN	NaN	NaN	NaN	NaN	NaN
Reference	0.0907	NaN	NaN	NaN	NaN	NaN	NaN
Science	0.0408	0.0179	0.0160	0.0185	0.0481	0.0428	**0.0521**
Social Science	0.1175	NaN	NaN	NaN	NaN	NaN	NaN
Society Culture	0.0714	0.0550	0.0395	0.0566	0.0812	0.0748	**0.0813**
Average	0.2462	0.3248	0.3202	0.3243	0.3417	0.3385	**0.3441**
Rank	7	4	6	5	2	3	1

Table 6.146: Effect of distance variation on Micro-F1 (↑) with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6278	0.6250	0.6302	0.6177	0.6219	**0.6378**	0.6324
Scene	0.7156	0.7474	0.7461	0.7373	**0.7507**	0.7443	0.7452
Image	0.7166	0.7329	0.7392	0.7463	0.7409	0.7403	**0.7603**
Yeast	**0.6303**	0.6104	0.6072	0.6083	0.6108	0.6093	0.6095
Arts Humanity	0.0480	0.0839	0.0650	0.0774	**0.2019**	0.1931	0.1814
Business Eco.	**0.6990**	0.6506	0.6493	0.6327	0.6675	0.6634	0.6247
Education	**0.2541**	0.1495	0.1443	0.1741	0.2266	0.2315	0.2164
Entertainment	0.2696	0.2341	0.2067	0.2374	**0.2870**	0.2732	0.2638
Health	0.4033	0.4092	0.3861	0.2877	**0.4116**	0.3946	0.3663
Reference	0.1652	0.0670	0.0592	0.1827	0.1165	0.1003	**0.2367**
Science	0.1063	0.0727	0.0649	0.0639	**0.1584**	0.1536	0.1431
Social Science	0.3865	0.2884	0.2923	0.3123	0.3903	**0.4141**	0.3749
Society Culture	0.2896	0.2323	0.2144	0.1659	0.2946	**0.3095**	0.2457
Average	0.4086	0.3772	0.3696	0.3726	**0.4214**	0.4204	0.4154
Rank	4	5	7	6	1	2	3

TABLE 6.147: Summary of effect of distance variation on MLFLD and MLFLD-MAXP performance with SimIC distance using TrTe

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	**0.0769**	0.0845	0.0824	0.0846	0.0917	0.0900	0.0911
RankLoss	**0.1081**	0.1611	0.1518	0.1548	0.1611	0.1518	0.1548
OneError	**0.3930**	0.5183	0.5168	0.5232	0.5183	0.5168	0.5232
Coverage	**3.4687**	5.2598	4.9376	5.0715	5.2598	4.9376	5.0715
AvgPrec	**0.6933**	0.5755	0.5839	0.5742	0.5755	0.5839	0.5742
Accuracy	0.3398	0.3208	0.3153	0.3084	0.3914	**0.3925**	0.3831
SubAcc	0.2548	0.2364	0.2360	0.2263	0.2931	**0.2964**	0.2850
Ex-F1	0.3685	0.3500	0.3425	0.3371	0.4260	**0.4261**	0.4179
Macro-F1	0.2462	0.3248	0.3202	0.3243	0.3417	0.3385	**0.3441**
Micro-F1	0.4086	0.3772	0.3696	0.3726	**0.4214**	0.4204	0.4154
ExecTime	6	32	29	111	33	36	146
Avg Rank	2.8	4.7	4.1	5.6	3.6	**2.1**	3.9
#Wins	5	0	0	0	1	**3**	1

Observations: Table 6.147 show that MLFLD-MAXP with Manhattan distance for feature similarity has topped for avg rank among seven experimentations when SimIC distance is used for label dissimilarity. To summarize,

- All MLFLD-MAXP variations defeated remaining experiments with Manhattan surpassing the remaining two distances. They raised accuracy, subset accuracy, and ex-F1 by 12-16%, and macro and micro-F1 approx. 8% and 2% w.r.t. MLkNN resp.
- Like other experiments, here also MLFLD proved itself better, showing fewer misclassifications, but could not exceed MLkNN for the same.
- For coverage, one error, avg precision, and rank loss, proposed algorithms revealed the equal performance and Manhattan experiments stood at rank 2 among 6 distance variants.

6.6.2.2 Performance of MLFLD and MLFLD-MAXP (cross-validation) using SimIC distance for label dissimilarity to check the effect of distance variation for feature similarity

This section has compared the functioning of proposed algorithms for ten folds by varying measures for feature similarity while keeping SimIC distance for label dissimilarity shown in Table 6.148 to 6.157.

TABLE 6.148: Effect of distance variation on Hamming Loss (↓) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1959	0.1952	**0.1929**	0.1986	0.1944	0.1935	0.1997
Image	0.1690	0.1620	**0.1601**	0.1628	0.1657	0.1630	0.1651
Scene	0.0861	0.0792	**0.0789**	0.0799	0.0807	0.0800	0.0806
Yeast	**0.1940**	0.2036	0.1983	0.2078	0.2041	0.1980	0.2080
CAL500	**0.1388**	0.1409	0.1401	0.1415	0.1409	0.1401	0.1415
Average	0.1568	0.1562	**0.1541**	0.1581	0.1572	0.1549	0.1590
Rank	4	3	1	6	5	2	7

TABLE 6.149: Effect of distance variation on Ranking Loss (↓) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.1594	0.1574	0.1508	**0.1580**	0.1574	**0.1508**	0.1580
Image	0.1680	0.1576	**0.1557**	0.1576	0.1576	**0.1557**	0.1576
Scene	0.0775	0.0693	0.0680	**0.0668**	0.0693	0.0680	**0.0668**
Yeast	**0.1670**	0.1772	0.1688	0.1840	0.1772	0.1688	0.1840
CAL500	**0.1828**	0.1856	0.1839	0.1866	0.1856	0.1839	0.1866
Average	0.1509	0.1494	**0.1454**	0.1506	0.1494	**0.1454**	0.1506
Rank	7	3	1	5	3	1	5

TABLE 6.150: Effect of distance variation on One Error (↓) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2699	0.2610	**0.2492**	0.2678	0.2610	**0.2492**	0.2678
Image	0.3000	0.2901	0.2866	**0.2886**	0.2901	0.2866	**0.2886**
Scene	0.2256	0.2046	0.2017	**0.1992**	0.2046	0.2017	**0.1992**
Yeast	**0.2300**	0.2506	0.2386	0.2602	0.2506	0.2386	0.2602
CAL500	**0.1176**	0.1240	0.1220	0.1260	0.1240	0.1220	0.1260
Average	0.2286	0.2261	**0.2196**	0.2284	0.2261	**0.2196**	0.2284
Rank	7	3	1	5	3	1	5

TABLE 6.151: Effect of distance variation on Coverage (↓) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	1.7764	1.7576	**1.7339**	1.7576	1.7576	**1.7339**	1.7576
Image	0.9390	0.8999	**0.8929**	0.8969	0.8999	**0.8929**	0.8969
Scene	0.4753	0.4304	0.4246	**0.4167**	0.4304	0.4246	**0.4167**
Yeast	**6.2750**	6.3697	6.2763	6.4390	6.3697	6.2763	6.4390
CAL500	130.56	130.6520	**130.2760**	131.2640	130.6520	**130.2760**	131.2640
Average	28.006	28.0219	**27.9207**	28.1548	28.0219	**27.9207**	28.1548
Rank	3	4	1	6	4	1	6

TABLE 6.152: Effect of distance variation on Average Precision (↑) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.8034	0.8061	**0.8146**	0.8038	0.8061	**0.8146**	0.8038
Image	0.8030	0.8104	**0.8123**	0.8116	0.8104	**0.8123**	0.8116
Scene	0.8652	0.8785	0.8805	**0.8826**	0.8785	0.8805	**0.8826**
Yeast	0.7650	0.7550	**0.7651**	0.7481	0.7550	**0.7651**	0.7481
CAL500	**0.4942**	0.4871	0.4903	0.4845	0.4871	0.4903	0.4845
Average	0.7462	0.7474	**0.7526**	0.7461	0.7474	**0.7526**	0.7461
Rank	5	3	1	6	3	1	6

TABLE 6.153: Effect of distance variation on Accuracy (↑) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.5340	0.5401	0.5513	0.5182	0.5619	**0.5671**	0.5422
Image	0.4937	0.5702	0.5600	0.5713	0.6179	**0.6215**	0.6188
Scene	0.6635	0.7110	0.7116	0.7235	0.7615	0.7637	**0.7641**
Yeast	**0.5162**	0.4862	0.4996	0.4840	0.4899	0.5019	0.4875
CAL500	0.1972	0.2077	0.2028	**0.2106**	0.2077	0.2028	**0.2106**
Average	0.4809	0.5030	0.5051	0.5015	0.5278	**0.5314**	0.5246
Rank	7	5	4	6	2	1	3

Table 6.154: Effect of distance variation on Subset Accuracy (↑) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.2934	0.3068	0.3119	0.2729	0.3169	**0.3220**	0.2814
Image	0.4090	0.4702	0.4692	0.4622	0.5093	**0.5188**	0.5043
Scene	0.6248	0.6696	0.6717	0.6687	0.7171	**0.7204**	0.7079
Yeast	0.1874	0.1954	**0.1983**	0.1925	0.1959	**0.1983**	0.1925
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3029	0.3284	0.3302	0.3193	0.3478	**0.3519**	0.3372
Rank	7	5	4	6	2	1	3

Table 6.155: Effect of distance variation on Ex-F1 (↑) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6141	0.6155	0.6307	0.5975	0.6415	**0.6485**	0.6269
Image	0.5223	0.6044	0.5911	0.6085	0.6551	0.6568	**0.6578**
Scene	0.6764	0.7249	0.7250	0.7419	0.7763	0.7782	**0.7830**
Yeast	**0.6204**	0.5819	0.5973	0.5801	0.5875	0.6011	0.5855
CAL500	0.3240	0.3377	0.3315	**0.3415**	0.3377	0.3315	0.3415
Average	0.5514	0.5729	0.5751	0.5739	0.5996	**0.6032**	0.5989
Rank	7	6	4	5	2	1	3

Table 6.156: Effect of distance variation on Macro-F1 (↑) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.6226	0.6596	0.6646	0.6459	0.6667	**0.6674**	0.6541
Image	0.5815	0.6358	0.6296	0.6399	0.6496	0.6508	**0.6541**
Scene	0.7364	0.7696	0.7701	0.7742	0.7793	0.7819	**0.7845**
Yeast	0.3853	NaN	NaN	NaN	NaN	NaN	NaN
CAL500	0.1714	NaN	NaN	NaN	NaN	NaN	NaN
Average	0.4994	0.6883	0.6881	0.6867	0.6985	**0.7000**	0.6976
Rank	7	4	5	6	2	1	3

TABLE 6.157: Effect of distance variation on Micro-F1 (↑) with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
Emotions	0.661	0.6665	0.6756	0.6520	0.6745	**0.6792**	0.6594
Image	0.5842	0.6328	0.6274	0.6372	0.6461	0.6478	**0.6505**
Scene	0.7332	0.7621	0.7626	0.7670	0.7709	0.7728	**0.7746**
Yeast	**0.6471**	0.6218	0.6323	0.6183	0.6227	0.6337	0.6194
CAL500	0.3209	0.3377	0.3303	**0.3416**	0.3377	0.3303	**0.3416**
Average	0.5893	0.6042	0.6056	0.6032	0.6104	**0.6128**	0.6091
Rank	7	5	4	6	2	1	3

TABLE 6.158: Summary of effect of distance variation on MLFLD and MLFLD-MAXP performance with SimIC distance using cross-validation

Dataset	MLkNN	MLFLD			MLFLD-MAXP		
		Euclidean	Manhattan	Minkowski	Euclidean	Manhattan	Minkowski
HamLoss	0.1568	0.1562	**0.1541**	0.1581	0.1572	0.1549	0.1590
RankLoss	0.1509	0.1494	**0.1454**	0.1506	0.1494	**0.1454**	0.1506
OneError	0.2286	0.2261	**0.2196**	0.2284	0.2261	**0.2196**	0.2284
Coverage	28.0060	28.0219	**27.9207**	28.1548	28.0219	**27.9207**	28.1548
AvgPrec	0.7462	0.7474	**0.7526**	0.7461	0.7474	**0.7526**	0.7461
Accuracy	0.4809	0.5030	0.5051	0.5015	0.5278	**0.5314**	0.5246
SubAcc	0.3029	0.3284	0.3302	0.3193	0.3478	**0.3519**	0.3372
Ex-F1	0.5514	0.5729	0.5751	0.5739	0.5996	**0.6032**	0.5989
Macro-F1	0.4994	0.6883	0.6881	0.6867	0.6985	**0.7000**	0.6976
Micro-F1	0.5893	0.6042	0.6056	0.6032	0.6104	**0.6128**	0.6091
ExecTime	17	65	65	61	55	56	64
Avg Rank	6.1	4.1	2.6	5.7	2.8	**1.1**	4.4
#Wins	0	0	5	0	0	**9**	0

Observations: From Table 6.158, when SimIC is used to measure label dissimilarity, then MLFLD-MAXP and Manhattan pair has outshined among seven experimentations. MLFLD-Manhattan pair functioned next to that of MLFLD-MAXP with Manhattan. To summarize,

- Among MLFLD-MAXP variations, Manhattan, Euclidean, and Minkowski achieved rank 1, 2, 3, respectively, for two accuracies and three F measures.
- All these variations elevated accuracy, ex-F1, and macro-F1 performance by approx. 8-10% compared to MLkNN. They raised subset accuracy and micro-F1 by approx. 11-16% and 3-4%, respectively.

- Pattern noted for one error, coverage, avg precision, and rank loss is the same for Hamming, Jaccard, and SimIC. The performance of MLFLD variants seems the same as that of corresponding MLFLD-MAXP variations.
- All six experimentations have improved rank loss and one error than MLkNN, while Minkowski variations are not able to improve avg precision and coverage.

6.6.3 Comparison of distances used for label dissimilarity

Throughout the experimentations, the main focus is to examine how the use of label dissimilarity measure affects the performance of MLFLD and MLFLD-MAXP. Initially, only Hamming distance is used for label dissimilarity, and feature similarity measure is taken as Euclidean, Manhattan, and Minkowski one by one. It resulted in three variants of MLFLD and MLFLD-MAXP each. These six variants are compared with MLkNN. Later Jaccard and SimIC distance measures are also used for label dissimilarity resulting in twelve new variants. In this section, all these variants are assessed together. Overall 2 X 3 X 3 = 18 variants are compared with each other and MLkNN.

6.6.3.1 Comparison of distance measures used for label dissimilarity with cross-validation

The performance of using ten folds is examined in this section, by varying measures of feature similarity and label dissimilarity. A summary is given in Table 6.159, and the individual metric is visualized in Figure 6.1 to 6.11.

TABLE 6.159: Comparison of distance measures used for label dissimilarity with cross-validation

Label Dissimilarity Measure	Algorithm	Feature Similarity Measure	Avg Rank	Execution Time
-	MLkNN	-	17.9	17
Hamming	MLFLD	Euclidean	10.1	60
		Manhattan	9.6	57
		Minkowski	12.2	70
	MLFLD-MAXP	Euclidean	6	58
		Manhattan	5.3	54
		Minkowski	7.1	65
Jaccard	MLFLD	Euclidean	12.5	62
		Manhattan	6.5	64
		Minkowski	9.6	72
	MLFLD-MAXP	Euclidean	8.5	52
		Manhattan	**2.9**	58
		Minkowski	7.3	81
SimIC	MLFLD	Euclidean	15.1	65
		Manhattan	9.4	65
		Minkowski	17.2	61
	MLFLD-MAXP	Euclidean	11	55
		Manhattan	4.9	56
		Minkowski	13.2	64

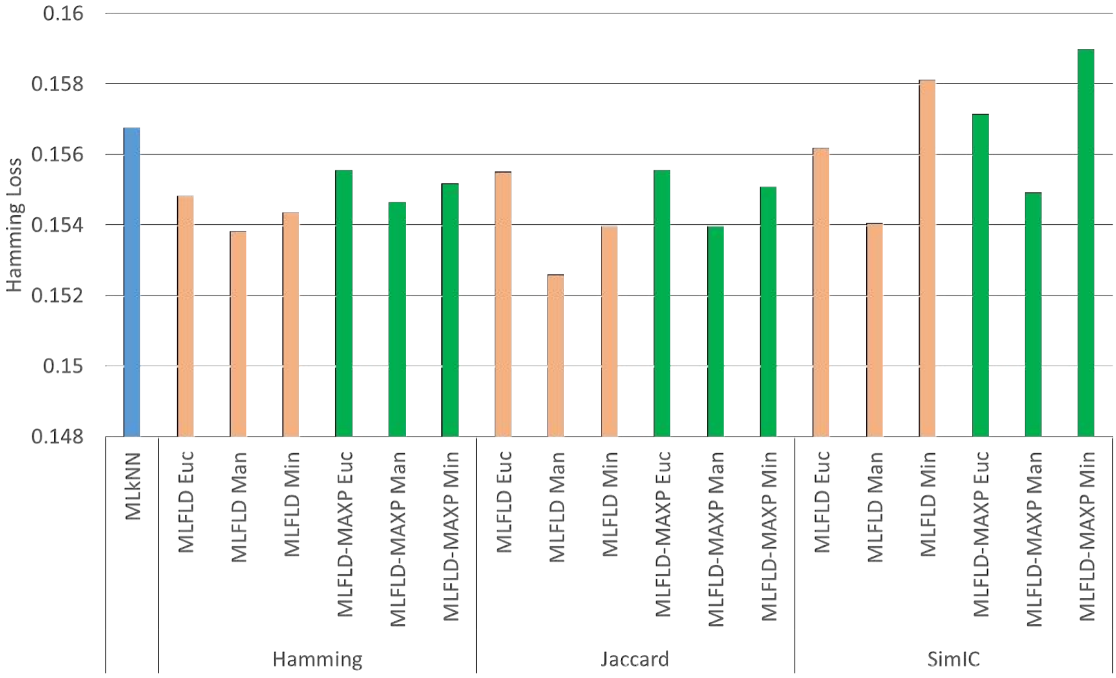


FIGURE 6.1: Comparison of distance measures used for label dissimilarity with cross-validation for Hamming Loss(↓)

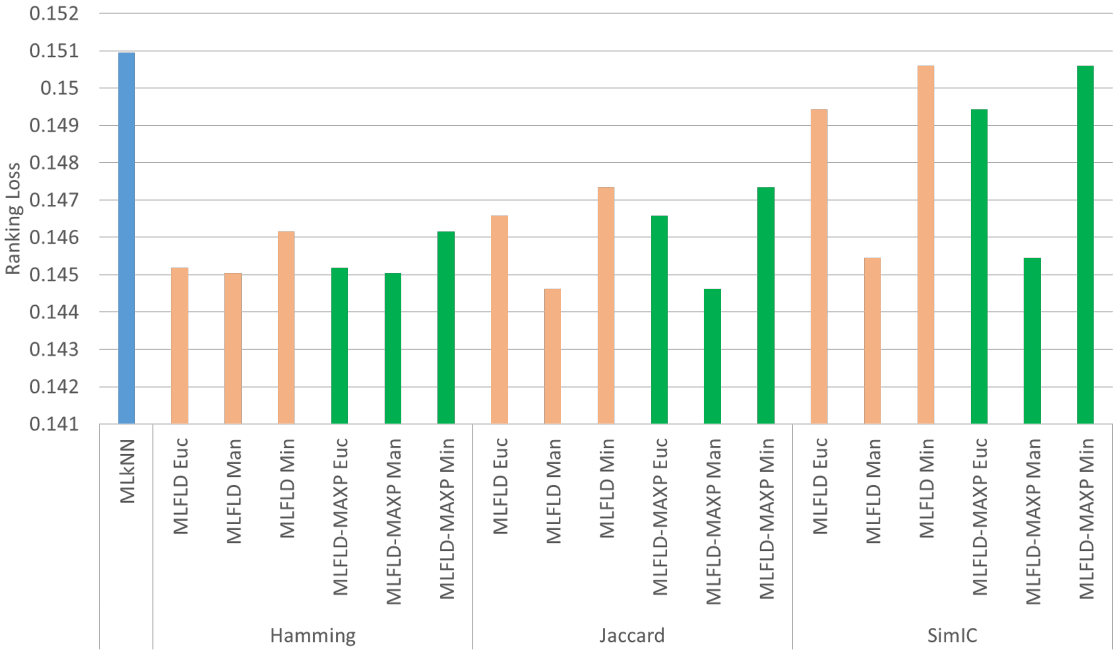


FIGURE 6.2: Comparison of distance measures used for label dissimilarity with cross-validation for Ranking Loss(↓)

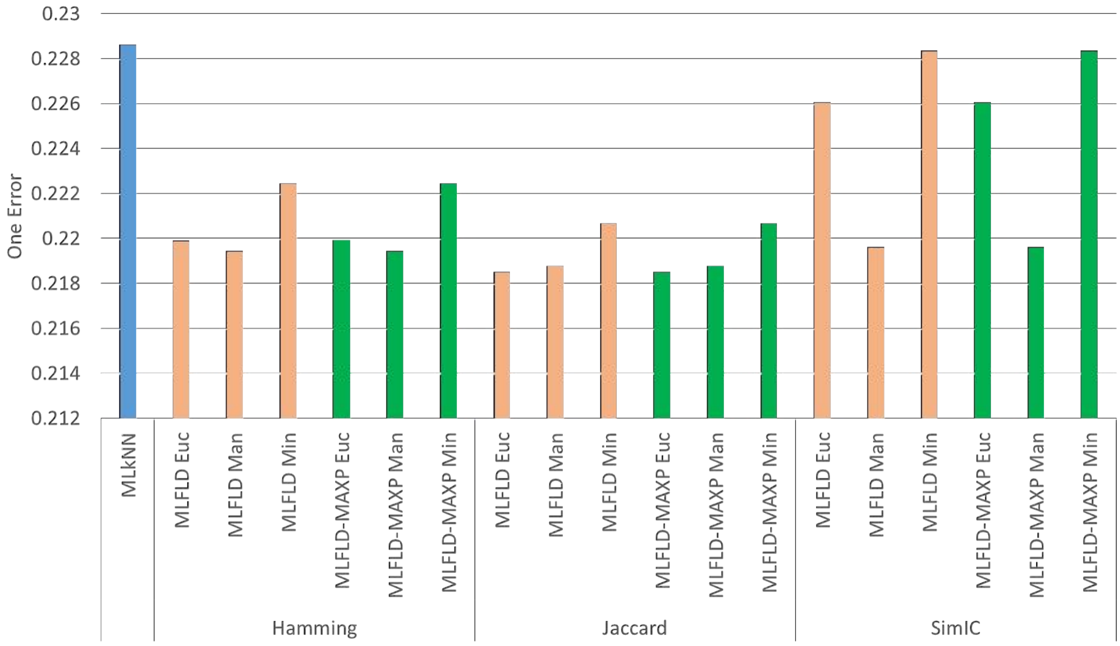


FIGURE 6.3: Comparison of distance measures used for label dissimilarity with cross-validation for One Error(↓)

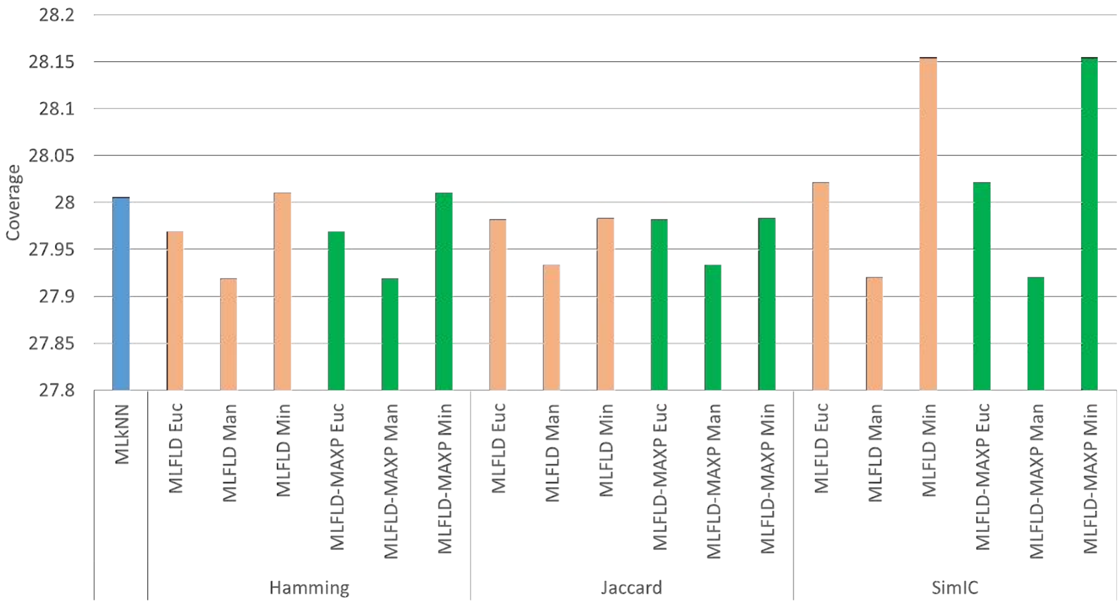


FIGURE 6.4: Comparison of distance measures used for label dissimilarity with cross-validation for Coverage(↓)

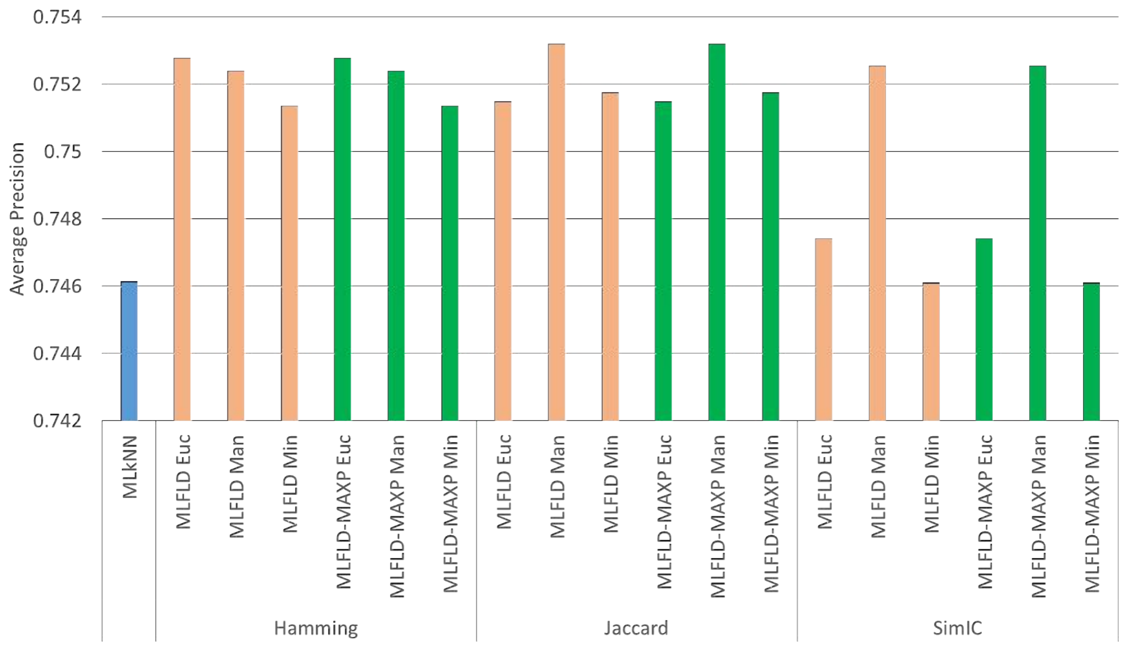


FIGURE 6.5: Comparison of distance measures used for label dissimilarity with cross-validation for Average Precision(↑)

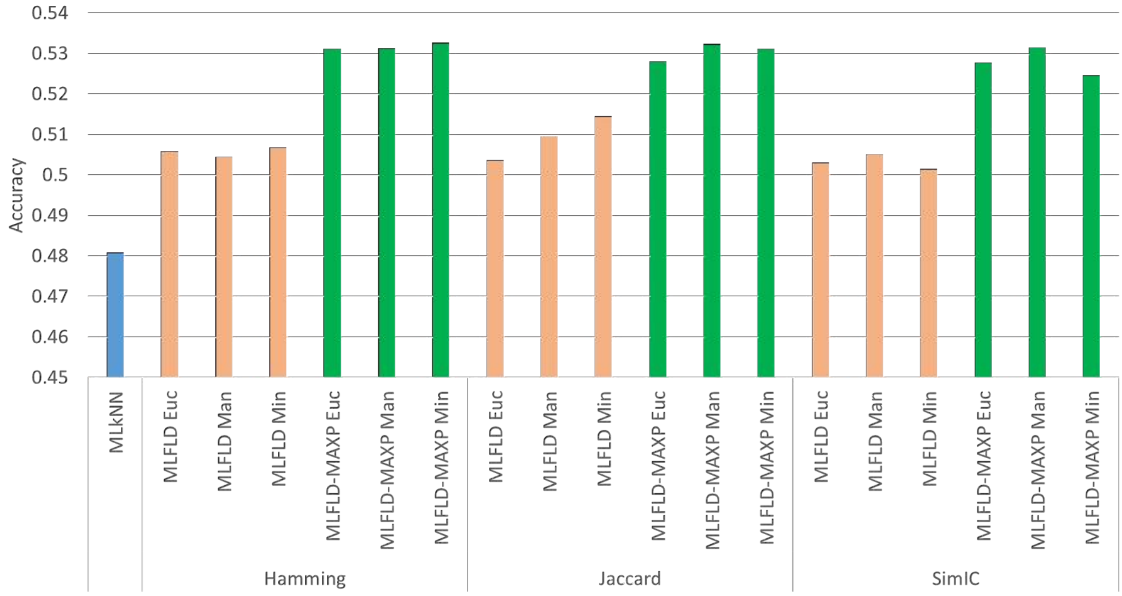


FIGURE 6.6: Comparison of distance measures used for label dissimilarity with cross-validation for Accuracy(↑)

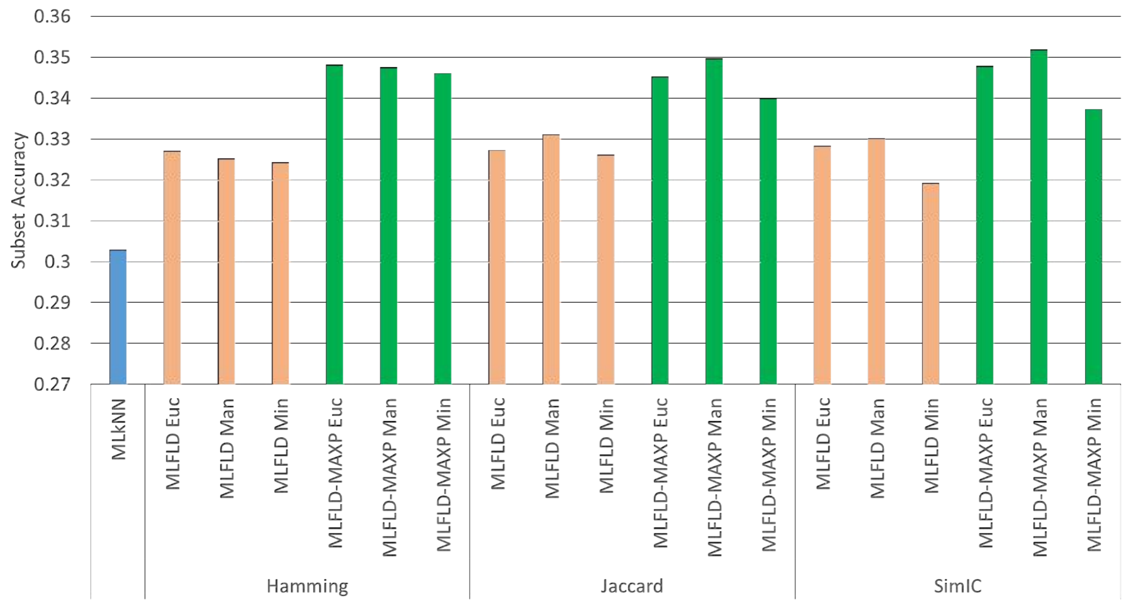


FIGURE 6.7: Comparison of distance measures used for label dissimilarity with cross-validation for Subset Accuracy(↑)

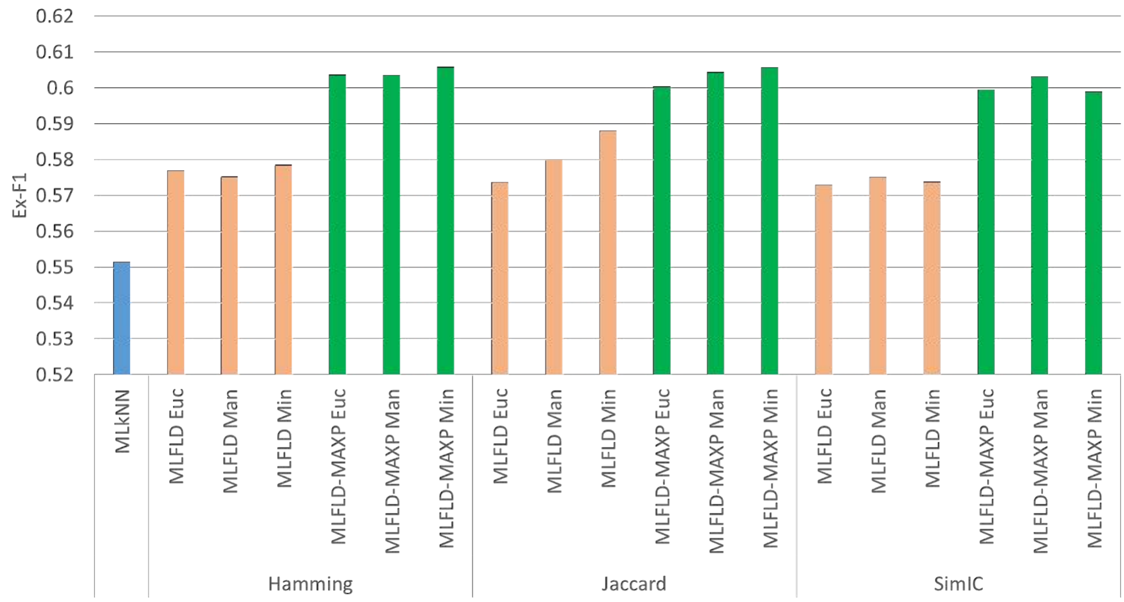


FIGURE 6.8: Comparison of distance measures used for label dissimilarity with cross-validation for Ex-F1(↑)

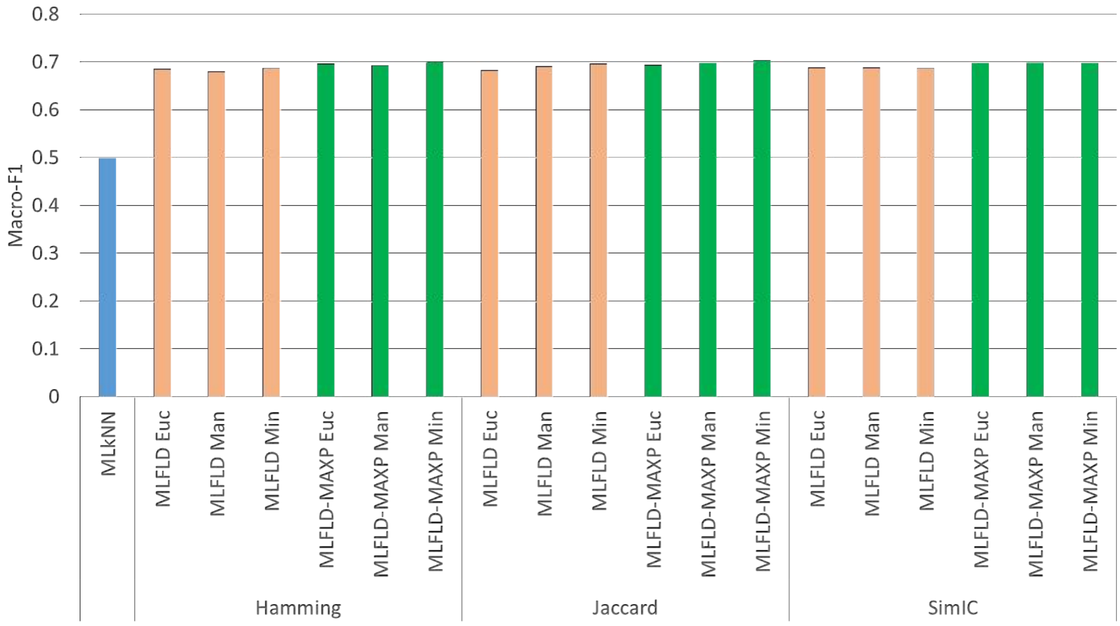


FIGURE 6.9: Comparison of distance measures used for label dissimilarity with cross-validation for Macro-F1(↑)

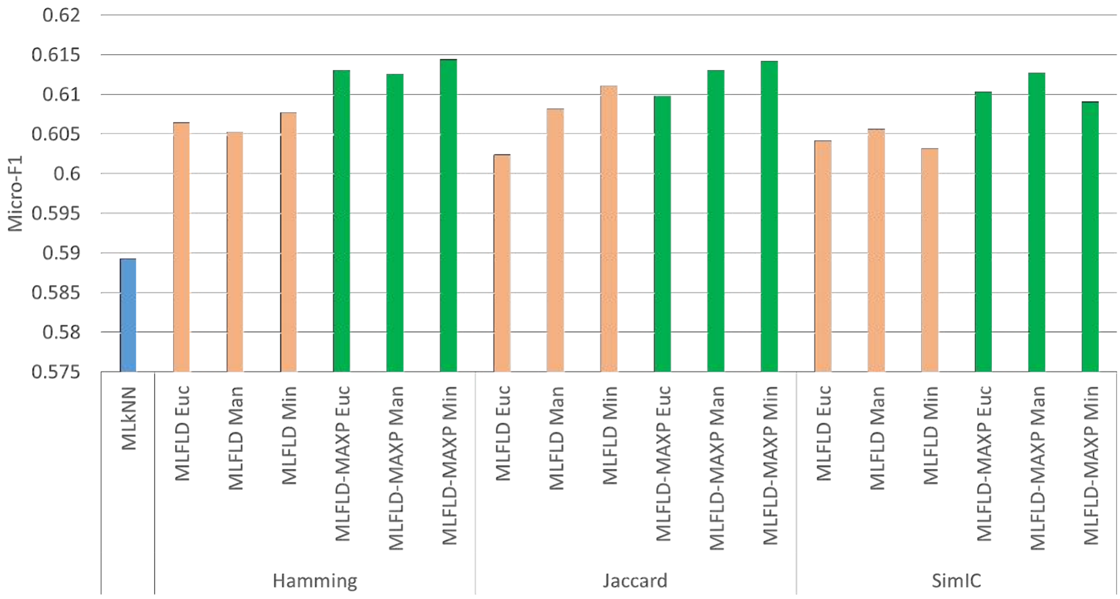


FIGURE 6.10: Comparison of distance measures used for label dissimilarity with cross-validation for Micro-F1(↑)

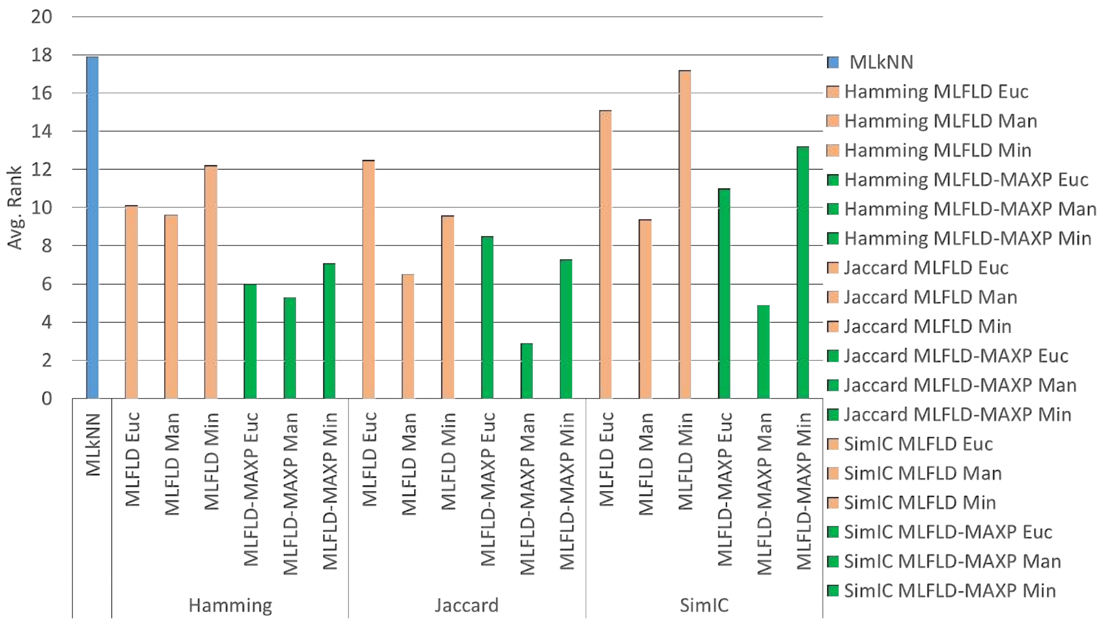


FIGURE 6.11: Comparison of distance measures used for label dissimilarity with cross-validation

From Figure 6.11 for ten folds experiments, (MLFLD-MAXP, Jaccard, Manhattan) triplet topped among 19 experiments with avg rank 2.9. To summarize,

- All variants of proposed algorithms defeated MLkNN in avg rank. It got a 17.9 avg rank.
- For all metrics, the proposed algorithm variants exceeded MLkNN except avg precision, hamming loss, and coverage showing 17, 16, and 13 ranks among 19.
- For the first five parameters, the same behavior of MLFLD and MLFLD-MAXP is seen for the same measures of feature similarity and label dissimilarity, as shown in Figure 6.1 - 6.5).
- For MLFLD-MAXP, variations with Hamming and Jaccard seemed to behave similarly, and both are viewed to be better than SimIC variants. This pattern is seen for micro-F1 (Figure 6.10), ex-F1 (Figure 6.8) and avg precision (Figure 6.5) also.
- For macro-F1, all variations are increased in functionality than MLkNN (Figure 6.9).
- Among feature similarity distance measures, Manhattan always exceeded the remaining two for both proposed algorithms and all label dissimilarity measures (Figure 6.11).

6.6.3.2 Comparison of distance measures used for label dissimilarity with train-test splits

The performance of using different combinations of feature and label dissimilarity measures is examined while using them with proposed algorithms. A summary is given in Table 6.160, and the individual metric is visualized in Figure 6.12 to 6.22 when train-test splits are used for experiments.

TABLE 6.160: Comparison of distance measures used for label dissimilarity with train-test splits

Label Dissimilarity Measure	Algorithm	Feature Similarity Measure	Avg Rank	Execution Time
-	MLkNN	-	6.4	6
Hamming	MLFLD	Euclidean	7.7	28
		Manhattan	9.9	31
		Minkowski	11.5	107
	MLFLD-MAXP	**Euclidean**	**3.5**	28
		Manhattan	4.8	28
		Minkowski	6	102
Jaccard	MLFLD	Euclidean	11.4	32
		Manhattan	10.6	35
		Minkowski	13.3	112
	MLFLD-MAXP	Euclidean	7.1	32
		Manhattan	6.5	31
		Minkowski	9.3	115
SimIC	MLFLD	Euclidean	13.6	32
		Manhattan	14.1	29
		Minkowski	15.9	111
	MLFLD-MAXP	Euclidean	12.1	33
		Manhattan	10.7	36
		Minkowski	12.4	146

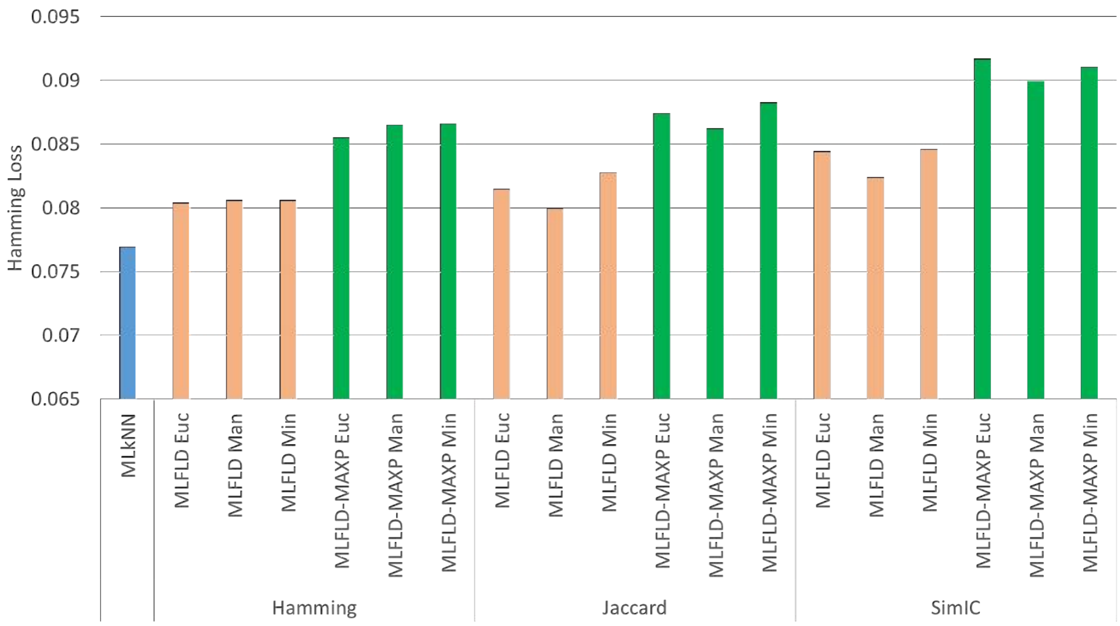


FIGURE 6.12: Comparison of distance measures used for label dissimilarity with train-test splits for Hamming Loss (↓)

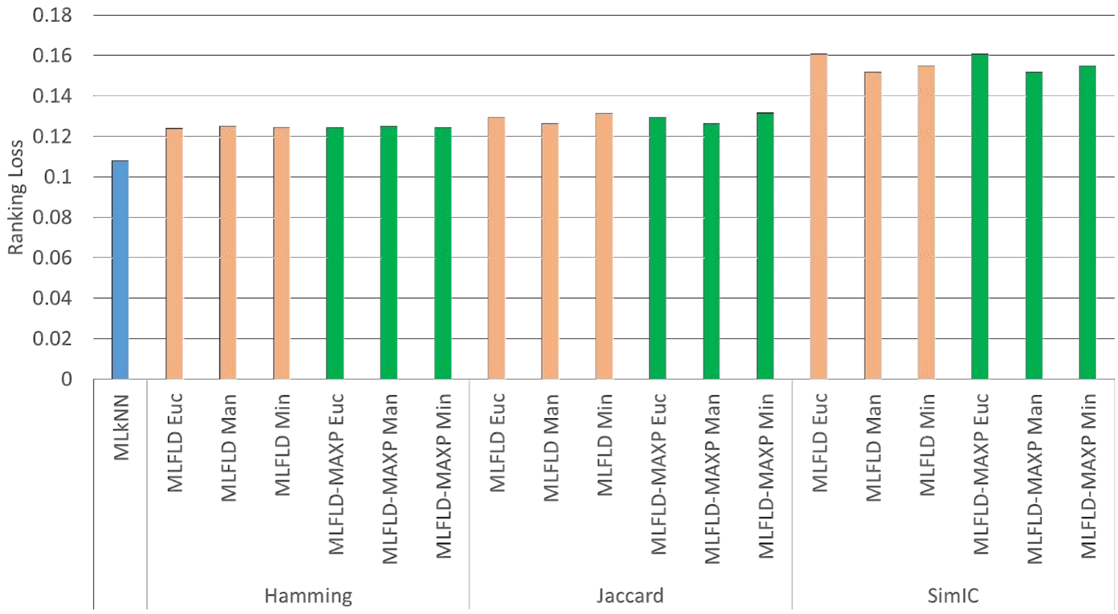


FIGURE 6.13: Comparison of distance measures used for label dissimilarity with train-test splits for Ranking Loss (↓)

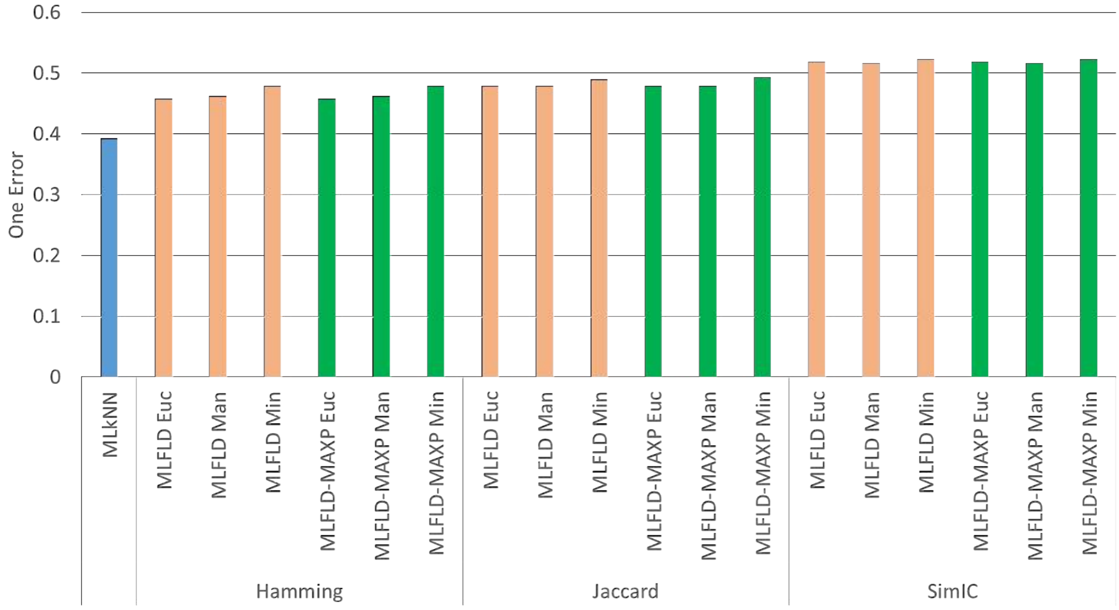


FIGURE 6.14: Comparison of distance measures used for label dissimilarity with train-test splits for One Error (↓)

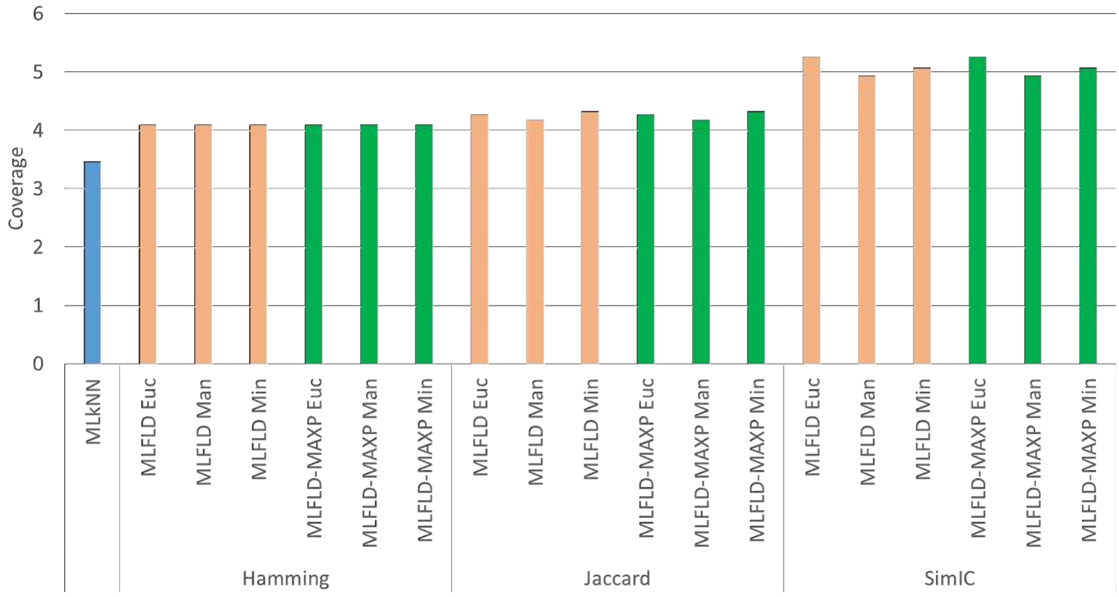


FIGURE 6.15: Comparison of distance measures used for label dissimilarity with train-test splits for Coverage (↓)

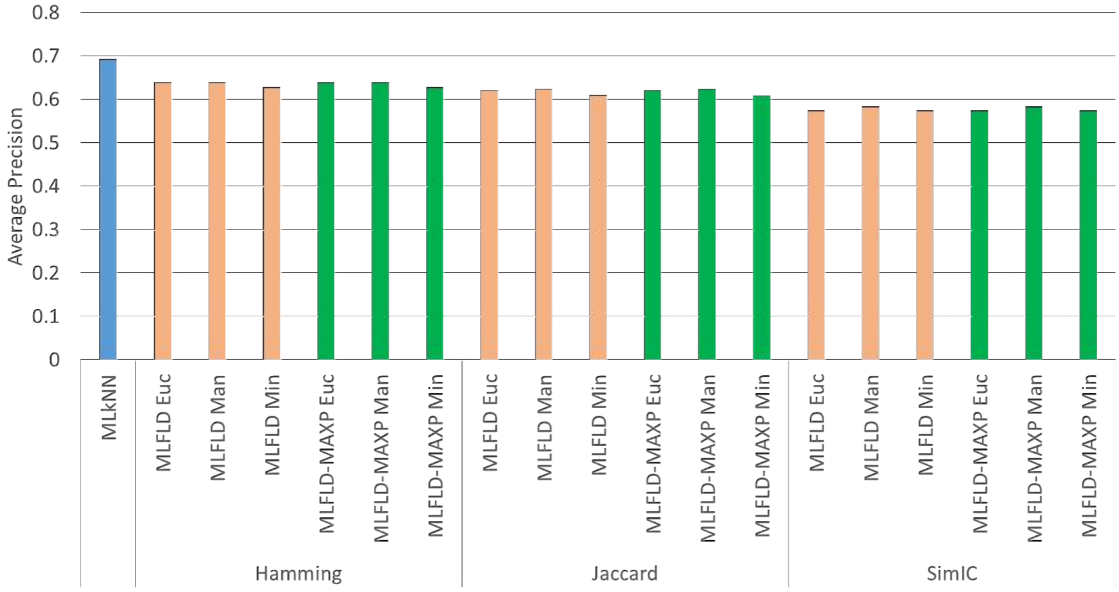


FIGURE 6.16: Comparison of distance measures used for label dissimilarity with train-test splits for Average Precision (↑)

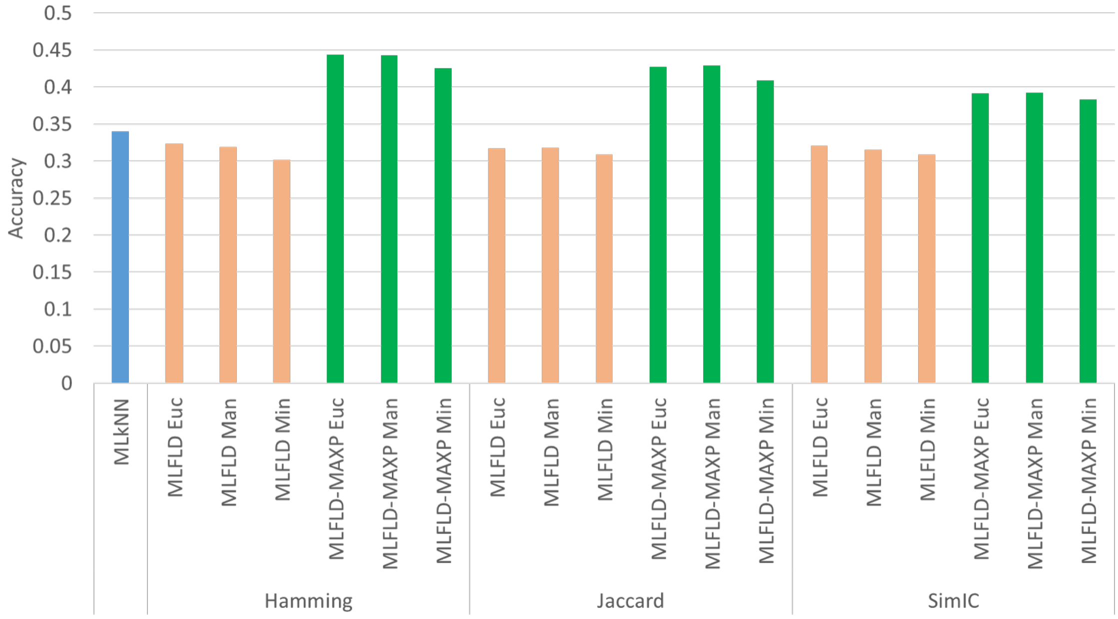


FIGURE 6.17: Comparison of distance measures used for label dissimilarity with train-test splits for Accuracy (↑)

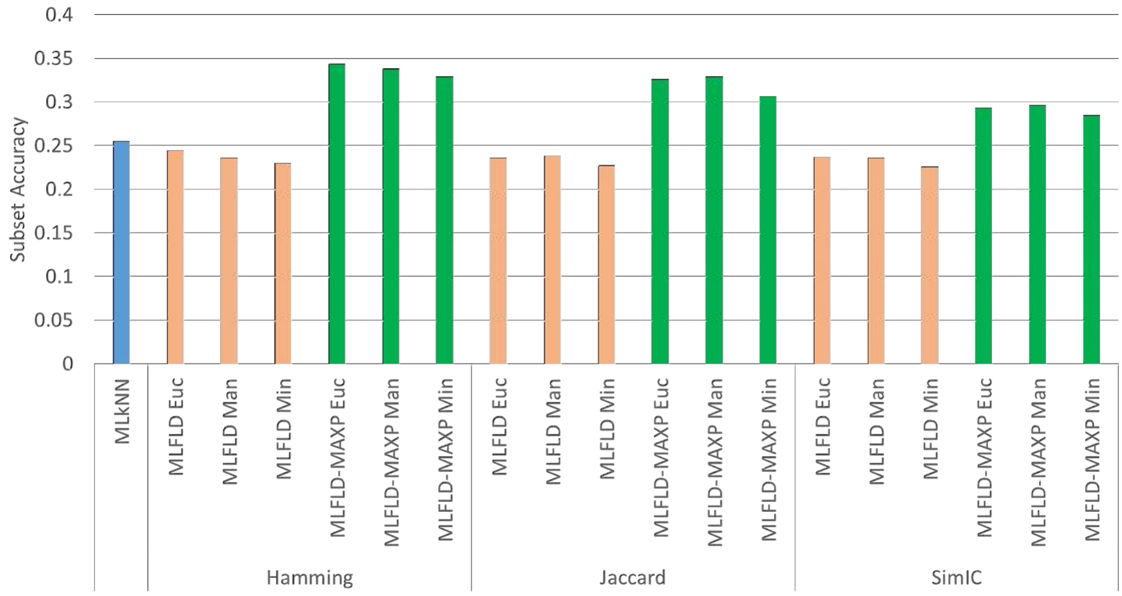


FIGURE 6.18: Comparison of distance measures used for label dissimilarity with train-test splits for Subset Accuracy (↑)

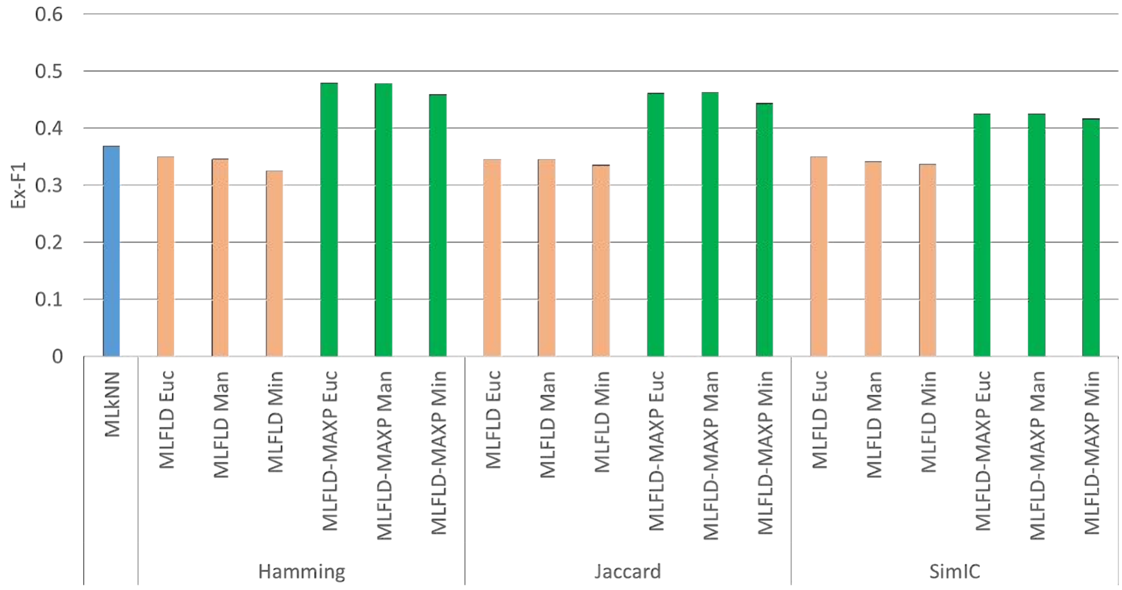


FIGURE 6.19: Comparison of distance measures used for label dissimilarity with train-test splits for Ex-F1 (↑)

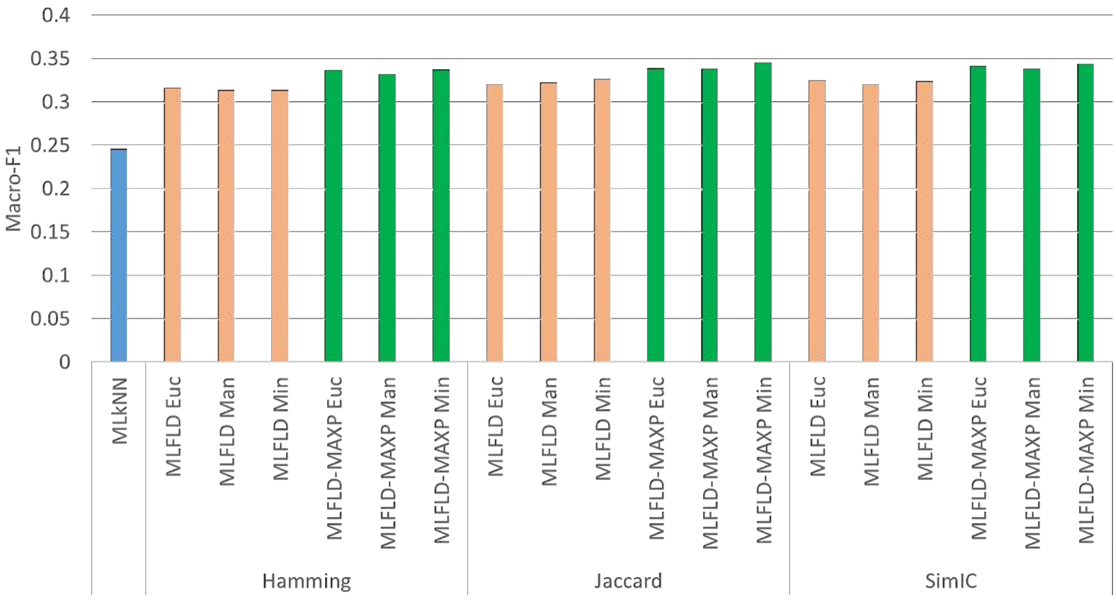


FIGURE 6.20: Comparison of distance measures used for label dissimilarity with train-test splits for Macro-F1 (↑)

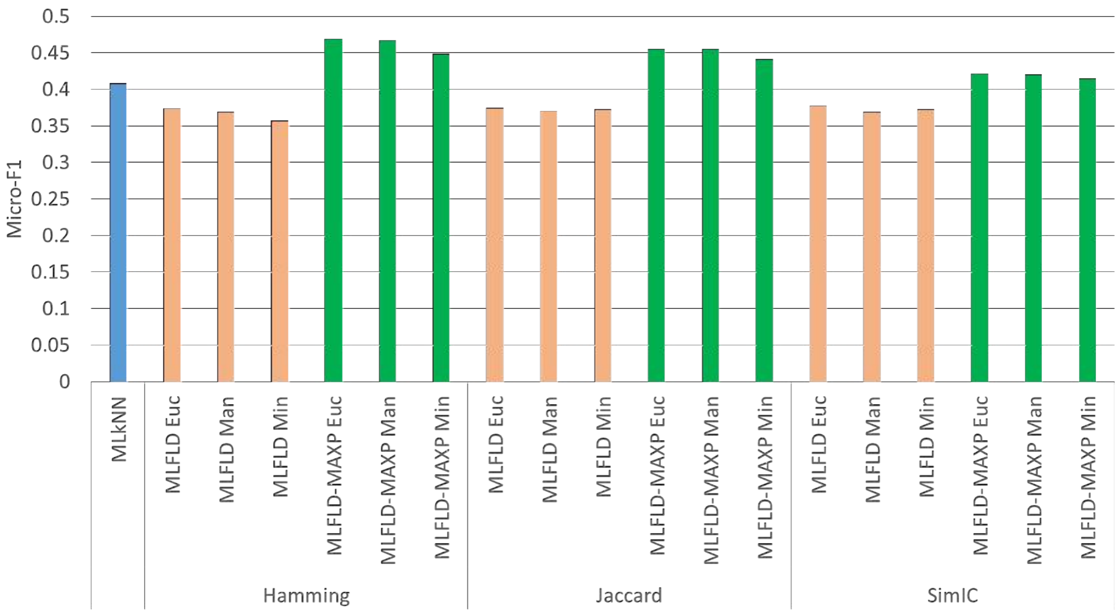


FIGURE 6.21: Comparison of distance measures used for label dissimilarity with train-test splits for Micro-F1 (↑)

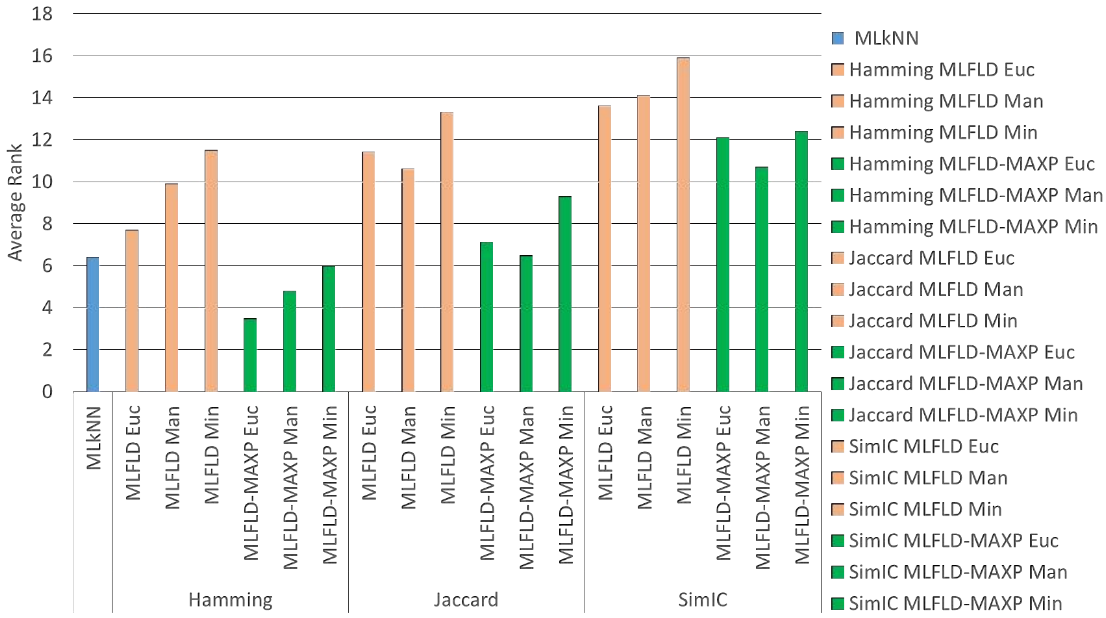


FIGURE 6.22: Comparison of distance measures used for label dissimilarity with train-test splits

From Table 6.160, in train-test experiments, (MLFLD-MAXP, Hamming, Euclidean) triplet topped among 19 experiments with rank 3.5. MLFLD-MAXP and hamming distance with Euclidean, Manhattan, Minkowski distances exceeded MLkNN in avg rank. They got avg rank 3.5, 4.8, 6, and 6.4, respectively. The remaining variants could not defeat MLkNN. It is followed by MLFLD-MAXP and Jaccard variants in avg rank as 7.1, 6.5, and 9.3. MLFLD-SimIC variations seemed not performing well. To summarize,

- MLFLD-MAXP and Hamming variants increased subset accuracy, accuracy, ex-F1, and micro-F1, followed by MLFLD-MAXP and Jaccard variants. Its Hamming variations increased subset accuracy by 29-35%, accuracy and ex-F1 by 25-30%, and micro-F1 by 10-15%, whereas Jaccard variants increased these parameters by 20-28%, 20-25% and 8-11% resp. MLFLD-MAXP and SimIC variants stood third, followed by MLkNN. MLFLD could not exceed MLkNN for these parameters.
- The exact opposite pattern is revealed for hamming loss. MLFLD-Hamming experiments are comparatively better for this parameter (Figure 6.12).
- For macro-F1, MLkNN is defeated by almost all variations of proposed algorithms showing up to 10% rise (Figure 6.20).

- MLkNN is seemed better for one error, ranking loss, coverage and average precision. Hamming and Jaccard variations exhibited little better than SimIC variations for these parameters(Figure 6.13-6.16).

6.7 Effect of feature selection on proposed algorithms

Attribute selection is proven to be beneficial to reduce the computational complexity of classifiers. There are different ways of attribute selection for multi-label data, as described in section 3.6 of chapter 3. The method used in this work is defined as an algorithm MLFS in chapter 4. Experiments are carried out, and a comparison of proposed algorithms with and without feature selection is made.

6.7.1 Effect of feature selection on MLFLD

In this section, the performance of MLFLD is examined when applied to datasets preprocessed by MLFS shown in Table 6.161 and 6.162.

Table 6.161: Effect of feature selection on for MLFLD

Hamming loss (↓)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**0.1938**	0.1969
Image	0.1631	**0.1608**
Scene	**0.0797**	0.0801
Yeast	**0.1981**	0.2039
CAL500	**0.1394**	0.1395
Average	**0.1548**	0.1562
Rank	1	2

Ranking loss (↓)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**0.1483**	0.1623
Image	**0.1570**	0.1604
Scene	**0.0682**	0.0707
Yeast	**0.1689**	0.1720
CAL500	0.1835	**0.1831**
Average	**0.1452**	0.1497
Rank	1	2

One Error (↓)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**0.2492**	0.2576
Image	0.2916	**0.2836**
Scene	**0.2050**	0.2112
Yeast	0.2378	**0.2369**
CAL500	**0.1160**	0.1240
Average	**0.2199**	0.2227
Rank	1	2

Coverage (↓)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**1.7102**	1.7644
Image	**0.8964**	0.9134
Scene	**0.4258**	0.4392
Yeast	6.2905	**6.2793**
CAL500	130.5240	**130.2020**
Average	27.9694	**27.9197**
Rank	2	1

Average Precision (↑)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**0.8183**	0.8063
Image	0.8105	**0.8121**
Scene	**0.8785**	0.8745
Yeast	**0.7648**	0.7643
CAL500	**0.4918**	0.4916
Average	**0.7528**	0.7498
Rank	1	2

Accuracy (↑)

Dataset	MLFLD	MLFS + MLFLD
Emotions	0.5483	**0.5507**
Image	0.5588	**0.5668**
Scene	0.7083	**0.7113**
Yeast	**0.5116**	0.5036
CAL500	0.2023	**0.2029**
Average	0.5059	**0.5071**
Rank	2	1

Table 6.162: Effect of feature selection on for MLFLD

Subset Accuracy (↑)

Dataset	MLFLD	MLFS + MLFLD
Emotions	0.3051	**0.3102**
Image	0.4632	**0.4723**
Scene	0.6629	**0.6658**
Yeast	0.2046	**0.2100**
CAL500	0.0000	0.0000
Average	0.3272	**0.3317**
Rank	2	1

Ex-F1 (↑)

Dataset	MLFLD	MLFS + MLFLD
Emotions	0.6274	**0.6292**
Image	0.5916	**0.5992**
Scene	0.7235	**0.7265**
Yeast	**0.6109**	0.6006
CAL500	0.3311	**0.3323**
Average	0.5769	**0.5776**
Rank	2	1

Macro-F1 (↑)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**0.6584**	0.6581
Image	0.6287	**0.6320**
Scene	**0.7683**	0.7681
Yeast	NaN	NaN
CAL500	NaN	NaN
Average	0.6851	**0.6861**
Rank	2	1

Micro-F1 (↑)

Dataset	MLFLD	MLFS + MLFLD
Emotions	**0.6727**	0.6698
Image	0.6259	**0.6312**
Scene	**0.7617**	0.7611
Yeast	**0.6426**	0.6325
CAL500	0.3294	**0.3306**
Average	**0.6757**	0.6737
Rank	1	2

Observations: From Table 6.163, MLFLD is noticed to perform almost similar before and after feature selection resulting in the same average rank 1.5 and 5 wins each. MLFLD with/without feature selection has outperformed MLkNN.

- For coverage, accuracy, subset accuracy, and ex-F1, feature selection proved to be beneficial, while for remaining parameters, it is not.

TABLE 6.163: Summary of effect of feature selection on MLFLD performance

Metric	MLFLD	MLFS + MLFLD
HamLoss	**0.1548**	0.1562
RankLoss	**0.1452**	0.1497
OneError	**0.2199**	0.2227
Coverage	27.9694	**27.9197**
AvgPrec	**0.7528**	0.7498
Accuracy	0.5059	**0.5071**
SubAcc	0.3272	**0.3317**
Ex-F1	0.5769	**0.5776**
Macro-F1	0.6851	**0.6861**
Micro-F1	**0.6757**	0.6737
ExecTime	60	37
Avg Rank	**1.5**	**1.5**
#Wins	**5**	**5**

- For avg precision, hamming loss, and micro-F1, performance is decreased slightly. But for one error and rank loss, the difference in fall is 1.2% and 3.1% resp.

6.7.2 Effect of feature selection on MLFLD-MAXP

In this section, the performance of MLFLD-MAXP with cross-validation is monitored and analyzed after feature selection shown in Table 6.164 and 6.165.

Table 6.164: Effect of feature selection on MLFLD-MAXP

Hamming loss (↓)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.1938**	0.1986
Image	0.1656	**0.1613**
Scene	**0.0812**	0.0828
Yeast	**0.1977**	0.2036
CAL500	**0.1394**	0.1395
Average	**0.1555**	0.1572
Rank	1	2

Ranking loss (↓)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.1483**	0.1623
Image	**0.1570**	0.1604
Scene	**0.0682**	0.0707
Yeast	**0.1689**	0.1720
CAL500	0.1835	**0.1831**
Average	**0.1452**	0.1497
Rank	1	2

One Error (↓)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.2492**	0.2576
Image	0.2916	**0.2836**
Scene	**0.2050**	0.2112
Yeast	0.2378	**0.2369**
CAL500	**0.1160**	0.1240
Average	**0.2199**	0.2227
Rank	1	2

Coverage (↓)

Dataset	MAXP	MLFS + MAXP
Emotions	**1.7102**	1.7644
Image	**0.8964**	0.9134
Scene	**0.4258**	0.4392
Yeast	6.2905	**6.2793**
CAL500	130.5240	**130.2020**
Average	27.9694	**27.9197**
Rank	2	1

Average Precision (↑)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.8183**	0.8063
Image	0.8105	**0.8121**
Scene	**0.8785**	0.8745
Yeast	**0.7648**	0.7643
CAL500	**0.4918**	0.4916
Average	**0.7528**	0.7498
Rank	1	2

Accuracy (↑)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.5627**	0.5617
Image	0.6169	**0.6260**
Scene	**0.7599**	0.7548
Yeast	**0.5140**	0.5061
CAL500	0.2023	**0.2029**
Average	**0.5312**	0.5303
Rank	1	2

Table 6.165: Effect of feature selection on MLFLD-MAXP

Subset Accuracy (↑)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.3136**	**0.3136**
Image	0.5108	**0.5198**
Scene	**0.7117**	0.7079
Yeast	0.2046	**0.2100**
CAL500	0.0000	0.0000
Average	0.3481	**0.3503**
Rank	2	1

Ex-F1 (↑)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.6441**	0.6431
Image	0.6532	**0.6623**
Scene	**0.7761**	0.7705
Yeast	**0.6145**	0.6046
CAL500	0.3311	0.3323
Average	**0.6038**	0.6026
Rank	1	2

Macro-F1 (↑)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.6609**	0.6593
Image	0.6482	**0.6566**
Scene	**0.7795**	0.7746
Yeast	NaN	NaN
CAL500	NaN	NaN
Average	0.6962	**0.6968**
Rank	2	1

Micro-F1 (↑)

Dataset	MAXP	MLFS + MAXP
Emotions	**0.6766**	0.6716
Image	0.6449	**0.6537**
Scene	**0.7706**	0.7653
Yeast	**0.6439**	0.6338
CAL500	0.3294	**0.3306**
Average	**0.6131**	0.6110
Rank	1	2

Observations: Table 6.166 has shown that MLFLD-MAXP is better for 7 metrics while it has enhanced only 3 parameters slightly after applying feature selection on datasets. Enhancement in subset accuracy, coverage and macro-F1 is 0.61%, 0.18% and 0.09% respectively. Most of the increase is noted for Image dataset. The reason may be that no. of features reduced for Image is remarkable compared to other datasets shown in Table 6.7.4(a).

TABLE 6.166: Summary of effect of feature selection on MLFLD-MAXP performance

Metric	MLFLD-MAXP	MLFS + MLFLD-MAXP
HamLoss	**0.1555**	0.1572
RankLoss	**0.1452**	0.1497
OneError	**0.2199**	0.2227
Coverage	27.9694	**27.9197**
AvgPrec	**0.7528**	0.7498
Accuracy	**0.5312**	0.5303
SubAcc	0.3481	**0.3503**
Ex-F1	**0.6038**	0.6026
Macro-F1	0.6962	**0.6968**
Micro-F1	**0.6131**	0.6110
ExecTime	58	36
Avg Rank	**1.3**	1.7
#Wins	**7**	3

6.7.3 Comparison of MLFLD and MLFLD-MAXP performance to check the effect of feature selection

How feature selection has affected working of both the proposed algorithms MLFLD and MLFLD-MAXP is monitored in this section.

First datasets are processed with algorithm MLFS. Then obtained datasets with reduced number of features and same number of labels are used further to evaluate

Table 6.167: sumaary of MLFLD and MLFLD-MAXP performance to check effect of feature selection

Metric	MLFLD	MAXP	MLFS followed by	
			MLFLD	MAXP
HamLoss	**0.1548**	0.1555	0.1562	0.1572
RankLoss	**0.1452**	**0.1452**	0.1497	0.1497
OneError	**0.2199**	**0.2199**	0.2227	0.2227
Coverage	27.9694	27.9694	**27.9197**	**27.9197**
AvgPrec	**0.7528**	**0.7528**	0.7498	0.7498
Accuracy	0.5059	**0.5312**	0.5071	0.5303
SubAcc	0.3272	0.3481	0.3317	**0.3503**
Ex-F1	0.5769	**0.6038**	0.5776	0.6026
Macro-F1	0.6851	0.6962	0.6861	**0.6968**
Micro-F1	**0.6757**	0.6131	0.6737	0.6110
ExecTime	60	58	36	36
Avg Rank	2.4	**1.7**	2.7	2.5
#Wins	5	**5**	1	3

Observations: From Table 6.167, MLFLD-MAXP has functioned well among four experiments. Feature selection has not enhanced the overall performance of the proposed algorithms.

6.7.4 Feature selection: Comparison with competing algorithms

First datasets are preprocessed with MLFS algorithm for attribute selection and then fed to all multi-label algorithms used for evaluation. When MLFS is run with threshold 1, those features selected for at least 1 label by selection criteria, are retained. Table 6.168(a) shows a number of features for all datasets. It shows that attributes of Image are relevant to more number of class labels among all datasets. When MLFS is executed with a threshold of 25%, it retained features selected for at least 25% labels. Similarly, for 50 and 75 % is also obtained, as shown in Table 6.169 and Figure 6.23.

Table 6.168: Number of features selected for datasets

Dataset	% Features Selected
Emotions	72
Image	84
Scene	68
Yeast	77
CAL500	75

Table 6.169: Percentage of features related to labels

Datasets	%Features retained related to at least			
	25% labels	50% labels	75% labels	1 label
Emotions	72	28	11	72
Image	68	27	9	84
Scene	84	12	2	68
Yeast	45	8	1	77
CAL500	1	1	0	75

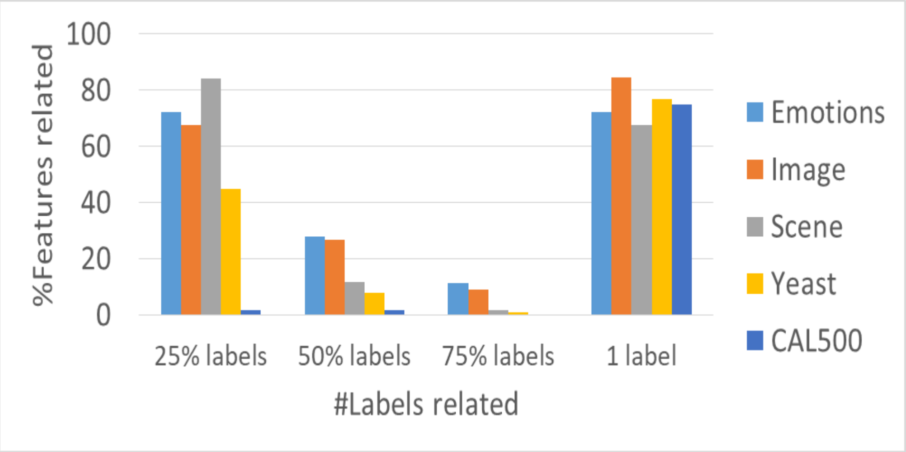
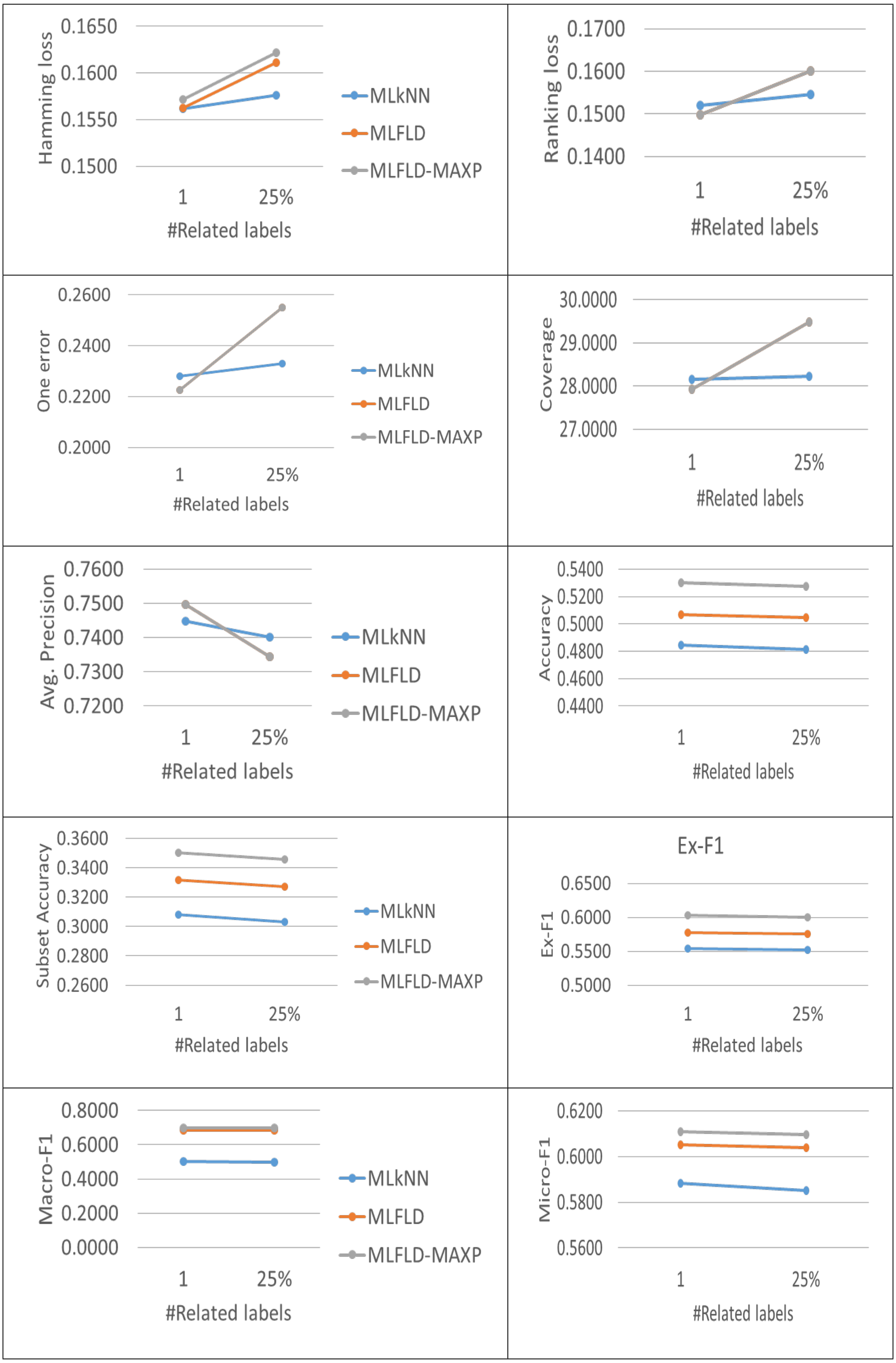


Figure 6.23: Percentage of features related to labels

When features related to at least 25% of labels are used, then no growth is seen in the performance of MLFLD-MAXP and MLFLD, as shown in Table 6.170. Hence the remaining thresholds are not used further. Only ML datasets obtained using thresholds 1 and 25% are used further.

Table 6.170: Effect of feature selection with different threshold



Proposed and competing methods are executed with ML datasets generated by the MLFS algorithm. Performance comparison is shown in Table 6.171 to 6.180.

TABLE 6.171: Performance of ML methods on selected features for Hamming loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.2515	0.2603	0.2576	0.2498	0.1925	0.2093	**0.1907**	0.1969	0.1986
Image	0.2237	0.2407	0.2267	0.2005	0.1740	0.4537	0.1698	**0.1608**	0.1613
Scene	0.1333	0.1480	0.1426	0.1180	0.0960	0.2249	0.0886	**0.0801**	0.0828
Yeast	0.2441	0.2823	0.2697	0.2505	0.1945	0.2242	**0.1918**	0.2039	0.2036
CAL500	0.1537	0.2046	0.1747	0.1530	0.1422	0.2600	0.1400	**0.1395**	**0.1395**
Average	0.2013	0.2272	0.2143	0.1944	0.1598	0.2744	**0.1562**	**0.1562**	0.1572
Rank	6	8	7	5	4	9	1	1	3

TABLE 6.172: Performance of ML methods on selected features for Ranking loss (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.3067	0.3233	0.3209	0.2491	0.1694	0.1663	**0.1607**	0.1623	0.1623
Image	0.2947	0.3175	0.2955	0.2145	0.1836	0.3833	0.1743	**0.1604**	**0.1604**
Scene	0.2340	0.2196	0.2352	0.1345	0.0927	0.1409	0.0782	**0.0707**	**0.0707**
Yeast	0.3059	0.4013	0.3267	0.3566	0.1750	0.1757	**0.1630**	0.1720	0.1720
CAL500	0.2610	0.6509	0.3564	0.6115	0.2318	**0.1770**	0.1837	0.1831	0.1831
Average	0.2805	0.3825	0.3069	0.3132	0.1705	0.2086	0.1520	**0.1497**	**0.1497**
Rank	6	9	7	8	4	5	3	1	1

TABLE 6.173: Performance of ML methods on selected features for One error (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.4201	0.4269	0.3981	0.3576	0.2596	0.2919	**0.2545**	0.2576	0.2576
Image	0.4545	0.4745	0.4405	0.3565	0.3350	0.6315	0.3240	**0.2836**	**0.2836**
Scene	0.3997	0.4101	0.3835	0.3033	0.2651	0.4442	0.2289	**0.2112**	**0.2112**
Yeast	0.3914	0.5226	0.3599	0.3144	0.2209	0.2334	**0.2135**	0.2369	0.2369
CAL500	0.6454	0.9880	0.7071	0.7747	0.1913	0.1355	**0.1195**	0.1240	0.1240
Average	0.4622	0.5644	0.4578	0.4213	0.2544	0.3473	0.2281	**0.2227**	**0.2227**
Rank	8	9	7	6	4	5	3	1	1

Table 6.174: Performance of ML methods on selected features for Coverage (↓)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	2.6019	2.6051	2.6775	2.2965	1.8341	1.7818	1.7882	**1.7644**	**1.7644**
Image	1.4470	1.5145	1.4480	1.1180	0.9925	1.7610	0.9650	**0.9134**	**0.9134**
Scene	1.2755	1.1953	1.2813	0.7661	0.5476	0.7910	0.4790	**0.4392**	**0.4392**
Yeast	9.2926	9.3328	8.9449	9.9916	6.4910	6.4052	**6.2039**	6.2793	6.2793
CAL500	165.901	170.916	170.018	170.955	152.008	**128.343**	131.372	130.202	130.202
Average	36.1036	37.1127	36.8739	37.0256	32.3747	**27.8166**	28.1617	27.9197	27.9197
Rank	6	9	7	8	5	1	4	2	2

Table 6.175: Performance of ML methods on selected features for Avg. Precision (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.6850	0.6816	0.6880	0.7303	0.8036	0.7931	**0.8071**	0.8063	0.8063
Image	0.6892	0.6727	0.6955	0.7623	0.7853	0.5827	0.7909	**0.8121**	**0.8121**
Scene	0.7215	0.7234	0.7277	0.8059	0.8428	0.7433	0.8632	**0.8745**	**0.8745**
Yeast	0.6273	0.5697	0.6234	0.6150	0.7658	0.7549	**0.7724**	0.7643	0.7643
CAL500	0.3902	0.1172	0.3177	0.1386	0.4573	**0.5089**	0.4904	0.4916	0.4916
Average	0.6226	0.5529	0.6105	0.6104	0.7310	0.6766	0.7448	**0.7498**	**0.7498**
Rank	6	9	7	8	4	5	3	1	1

Table 6.176: Performance of ML methods on selected features for Accuracy (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.4355	0.4685	0.4639	0.4697	0.5227	0.5490	0.5509	0.5507	**0.5617**
Image	0.4525	0.4798	0.5042	0.5192	0.4574	0.1772	0.4975	0.5668	**0.6260**
Scene	0.5409	0.5796	0.5939	0.5987	0.6117	0.4105	0.6683	0.7113	**0.7548**
Yeast	0.4370	0.4089	0.4192	0.3797	0.5016	**0.5271**	0.5193	0.5036	0.5061
CAL500	0.2125	0.1908	0.2265	0.0228	0.1832	**0.2985**	0.1876	0.2029	0.2029
Average	0.4157	0.4255	0.4415	0.3980	0.4553	0.3925	0.4847	0.5071	**0.5303**
Rank	7	6	5	8	4	9	3	2	1

Table 6.177: Performance of ML methods on selected features for Subset Accuracy (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.1853	0.2242	0.2175	0.1890	0.2766	0.2834	0.3070	0.3102	**0.3136**
Image	0.3000	0.3710	0.3790	0.3815	0.3940	0.0095	0.4185	0.4723	**0.5198**
Scene	0.4375	0.5384	0.5447	0.5235	0.5875	0.1225	0.6253	0.6658	**0.7079**
Yeast	0.0666	0.1328	0.1361	0.0381	0.2019	0.1303	0.1891	0.2100	**0.2100**
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.1979	0.2533	0.2555	0.2264	0.2920	0.1091	0.3080	0.3317	**0.3503**
Rank	8	6	5	7	4	9	3	2	1

Table 6.178: Performance of ML methods on selected features for Ex-F1 (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.5215	0.5528	0.5477	0.5624	0.6028	0.6360	0.6304	0.6292	**0.6431**
Image	0.5068	0.5175	0.5478	0.5672	0.4788	0.2610	0.5242	0.5992	**0.6623**
Scene	0.5770	0.5935	0.6107	0.6242	0.6198	0.5207	0.6827	0.7265	**0.7705**
Yeast	0.5604	0.5099	0.5227	0.5042	0.6005	**0.6421**	0.6237	0.6006	0.6046
CAL500	0.3449	0.3099	0.3587	0.0434	0.3032	**0.4516**	0.3109	0.3323	0.3323
Average	0.5021	0.4967	0.5175	0.4603	0.5210	0.5023	0.5544	0.5776	**0.6026**
Rank	7	8	5	9	4	6	3	2	1

Table 6.179: Performance of ML methods on selected features for Macro-F1 (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.5577	0.5718	0.5682	0.6029	0.6270	0.6656	0.6316	0.6581	**0.6593**
Image	0.5487	0.5161	0.5456	0.5878	0.5451	0.3096	0.5785	0.6320	**0.6566**
Scene	0.6365	0.5981	0.6191	0.6684	0.6900	0.5762	0.7354	0.7681	**0.7746**
Yeast	0.3863	0.3748	0.3862	0.2721	0.3944	**0.4415**	0.3883	NaN	NaN
CAL500	0.2120	0.1816	0.2373	0.1240	0.1871	**0.2419**	0.1691	NaN	NaN
Average	0.4682	0.4485	0.4713	0.4510	0.4887	0.4470	0.5006	0.6861	**0.6968**
Rank	6	8	5	7	4	9	3	2	1

Table 6.180: Performance of ML methods on selected features for Micro-F1 (↑)

Dataset	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
Emotions	0.5802	0.5840	0.5796	0.6106	0.6558	**0.6777**	0.6716	0.6698	0.6716
Image	0.5480	0.5146	0.5461	0.5867	0.5494	0.3232	0.5818	0.6312	**0.6537**
Scene	0.6262	0.5875	0.6071	0.6611	0.6894	0.5542	0.7297	0.7611	**0.7653**
Yeast	0.5834	0.5345	0.5463	0.5293	0.6354	**0.6547**	0.6499	0.6325	0.6338
CAL500	0.3464	0.3150	0.3620	0.0452	0.3044	**0.4530**	0.3083	0.3306	0.3306
Average	0.5368	0.5071	0.5282	0.4866	0.5669	0.5326	0.5883	0.6050	**0.6110**
Rank	5	8	7	9	4	6	3	2	1

TABLE 6.181: Summary of performance comparison of ML methods on selected features

Metric	BR	LP	CC	RAkEL	BRkNN	BPMLL	MLkNN	MLFLD	MAXP
HamLoss	0.2013	0.2272	0.2143	0.1944	0.1598	0.2744	**0.1562**	**0.1562**	0.1572
RankLoss	0.2805	0.3825	0.3069	0.3132	0.1705	0.2086	0.1520	**0.1497**	**0.1497**
OneError	0.4622	0.5644	0.4578	0.4213	0.2544	0.3473	0.2281	**0.2227**	**0.2227**
Coverage	36.1036	37.1127	36.8739	37.0256	32.3747	**27.8166**	28.1617	27.9197	27.9197
AvgPrec	0.6226	0.5529	0.6105	0.6104	0.7310	0.6766	0.7448	**0.7498**	**0.7498**
Accuracy	0.4157	0.4255	0.4415	0.3980	0.4553	0.3925	0.4847	0.5071	**0.5303**
SubAcc	0.1979	0.2533	0.2555	0.2264	0.2920	0.1091	0.3080	0.3317	**0.3503**
Ex-F1	0.5021	0.4967	0.5175	0.4603	0.5210	0.5023	0.5544	0.5776	**0.6026**
Macro F1	0.4682	0.4485	0.4713	0.4510	0.4887	0.4470	0.5006	0.6861	**0.6968**
Micro F1	0.5368	0.5071	0.5282	0.4866	0.5669	0.5326	0.5883	0.6050	**0.6110**
Avg Rank	6.5	8	6.2	7.5	4.1	6.4	2.9	1.6	**1.3**
#Wins	0	0	0	0	0	1	1	4	**8**

Observations: Overall performance of proposed algorithms is increased mainly because of raised performance in Image and Scene for all metrics and Emotions for few parameters. From Table 6.181, experimentation has shown that

- MLFLD-MAXP ranked first, showing 8 wins over 10 metrics, whereas MLFLD ranked second with 4 wins.
- MLFLD-MAXP performed slightly better than MLFLD for two accuracy and three F-measures. Both outperformed MLkNN and other contestants.
- Both proposed algorithms are similar for one error, coverage, rank loss, and avg precision.
- MLFLD achieved the same avg hamming loss as that of MLkNN that is not seen in other experiments.
- The performance of MLFLD-MAXP is slightly lesser than both these algorithms. Both algorithms outperformed compared to competing algorithms for all metrics, except ham loss and coverage.

6.8 Effect of instance selection on proposed algorithms

Algorithm MLIS is run with sampling with a replacement for size 60, 70, 80, 90, 100. An experiment is conducted on five datasets, and then resulting datasets are fed to proposed algorithms.

6.8.1 Effect of instance selection on MLFLD

In this section, how MLFLD has performed, is studied when it is fed with sampled datasets of different sizes processed by MLIS algorithm shown in Table 6.182 to 6.191.

TABLE 6.182: Effect of instance selection on Hamming loss (↓) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.2000	0.1768	0.1805	**0.1682**	0.1938
Image	0.1545	0.1535	0.1455	**0.1440**	0.1631
Scene	0.0819	0.0793	0.0802	**0.0773**	0.0797
Yeast	0.2010	0.1982	0.1962	**0.1922**	0.1981
CAL500	0.1397	0.1404	0.1383	**0.1367**	0.1394
Average	0.1554	0.1496	0.1481	**0.1437**	0.1548
Rank	5	3	2	1	4

TABLE 6.183: Effect of instance selection on Ranking loss (↓) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.1459	0.1361	0.1386	**0.1274**	0.1483
Image	0.1579	0.1477	0.1363	**0.1314**	0.1570
Scene	0.0731	0.0653	0.0637	**0.0620**	0.0682
Yeast	0.1632	0.1560	0.1515	**0.1478**	0.1689
CAL500	0.1709	0.1662	0.1616	**0.1579**	0.1835
Average	0.1422	0.1343	0.1303	**0.1253**	0.1452
Rank	4	3	2	1	5

Table 6.184: Effect of instance selection on One error (↓) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.2457	0.2244	0.2383	**0.2208**	0.2492
Image	0.2878	0.2852	0.2626	**0.2562**	0.2916
Scene	0.2125	0.2054	0.2010	**0.1981**	0.2050
Yeast	0.2374	0.2225	0.2233	**0.2212**	0.2378
CAL500	0.0933	0.1086	0.1025	**0.0889**	0.1160
Average	0.2153	0.2092	0.2055	**0.1970**	0.2199
Rank	4	3	2	1	5

Table 6.185: Effect of instance selection on Coverage (↓) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	1.7057	1.6585	1.6404	**1.5642**	1.7102
Image	0.8816	0.8392	0.7911	**0.7860**	0.8964
Scene	0.4493	0.4101	0.4005	**0.3898**	0.4258
Yeast	6.2042	6.0734	6.0202	**5.9470**	6.2905
CAL500	117.2167	113.1657	113.6925	**109.6889**	130.5240
Average	25.2915	24.4294	24.5089	**23.6752**	27.9694
Rank	4	2	3	1	5

Table 6.186: Effect of instance selection on Avg. Precision (↑) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.8194	0.8343	0.8306	**0.8421**	0.8183
Image	0.8108	0.8166	0.8312	**0.8343**	0.8105
Scene	0.8735	0.8799	0.8837	**0.8861**	0.8785
Yeast	0.7706	0.7787	0.7838	**0.7857**	0.7648
CAL500	0.5048	0.5054	0.5208	**0.5301**	0.4918
Average	0.7558	0.7630	0.7700	**0.7757**	0.7528
Rank	4	3	2	1	5

Table 6.187: Effect of instance selection on Accuracy (↑) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.5198	0.5606	0.5699	**0.6025**	0.5483
Image	0.5504	0.5668	0.5878	**0.5998**	0.5588
Scene	0.6912	0.6865	0.6925	**0.7093**	0.7083
Yeast	0.5227	0.5329	0.5403	**0.5525**	0.5116
CAL500	0.2069	0.2146	0.2255	**0.2382**	0.2023
Average	0.4982	0.5123	0.5232	**0.5405**	0.5059
Rank	5	3	2	1	4

Table 6.188: Effect of instance selection on Subset Accuracy (↑) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.2571	0.3268	0.3532	**0.3717**	0.3051
Image	0.4587	0.4710	0.5035	**0.5086**	0.4632
Scene	0.6486	0.6464	0.6484	0.6602	**0.6629**
Yeast	0.1898	0.2053	0.2176	**0.2217**	0.2046
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3108	0.3299	0.3445	**0.3524**	0.3272
Rank	5	3	2	1	4

Table 6.189: Effect of instance selection on Ex-F1 (↑) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.6082	0.6383	0.6438	**0.6783**	0.6274
Image	0.5815	0.5993	0.6167	**0.6306**	0.5916
Scene	0.7054	0.6999	0.7073	**0.7258**	0.7235
Yeast	0.6231	0.6338	0.6385	**0.6507**	0.6109
CAL500	0.3369	0.3467	0.3612	**0.3762**	0.3311
Average	0.5710	0.5836	0.5935	**0.6123**	0.5769
Rank	5	3	2	1	4

Table 6.190: Effect of instance selection on Macro-F1 (↑) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.6043	0.6532	0.6726	**0.6933**	0.6584
Image	0.6240	0.6419	0.6594	**0.6691**	0.6287
Scene	0.7510	0.7518	0.7516	0.7657	**0.7683**
Yeast	0.4416	0.4609	0.4667	**0.4813**	NaN
CAL500	NaN	NaN	NaN	NaN	NaN
Average	0.6052	0.6270	0.6376	0.6524	**0.6851**
Rank	5	4	3	2	1

Table 6.191: Effect of instance selection on Micro-F1 (↑) for MLFLD

Dataset	MLIS + MLFLD				
	60	70	80	90	100
Emotions	0.6523	0.6914	0.6927	**0.7159**	0.6727
Image	0.6323	0.6431	0.6612	**0.6689**	0.6259
Scene	0.7501	0.7553	0.7552	**0.7674**	0.7617
Yeast	0.6510	0.6599	0.6656	**0.6762**	0.6426
CAL500	0.3381	0.3499	0.3640	**0.3807**	0.3294
Average	0.6714	0.6874	0.6937	**0.7071**	0.6757
Rank	5	3	2	1	4

Table 6.192: Summary of effect of instance selection on MLFLD performance

Dataset	MLIS + MLFLD				
	60	70	80	90	100
HamLoss	0.1554	0.1496	0.1481	**0.1437**	0.1548
RankLoss	0.1422	0.1343	0.1303	**0.1253**	0.1452
OneError	0.2153	0.2092	0.2055	**0.1970**	0.2199
Coverage	25.2915	24.4294	24.5089	**23.6752**	27.9694
AvgPrec	0.7558	0.7630	0.7700	**0.7757**	0.7528
Accuracy	0.4982	0.5123	0.5232	**0.5405**	0.5059
SubAcc	0.3108	0.3299	0.3445	**0.3524**	0.3272
Ex-F1	0.5710	0.5836	0.5935	**0.6123**	0.5769
Macro F1	0.6052	0.6270	0.6376	0.6524	**0.6851**
Micro F1	0.6714	0.6874	0.6937	**0.7071**	0.6757
Avg Rank	4.6	3.0	2.2	**1.1**	4.1
#Wins	0	0	0	**9**	1

Observations: Table 6.192 has shown that instance selection with 70, 80, and 90% replacement has been proved effective to boost MLFLD functionality. 90% is noticed to make more progress. 60% of replacement has not worked well. For the last five parameters, growth is seen in all datasets except Scene for 2 parameters. Emotions seemed to work well for 70 than 80.

6.8.2 Effect of instance selection on MLFLD-MAXP

The performance of MLFLD-MAXP on sampled data is examined in this section for 5 sizes separately.

Table 6.193: Effect of instance selection on Hamming Loss (↓) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.2010	0.1797	0.1812	**0.1692**	0.1938
Image	0.1598	0.1591	0.1466	**0.1446**	0.1656
Scene	0.0843	0.0812	0.0815	**0.0804**	0.0812
Yeast	0.2011	0.1980	0.1962	**0.1922**	0.1977
CAL500	0.1397	0.1404	0.1383	**0.1367**	0.1394
Average	0.1572	0.1517	0.1488	**0.1446**	0.1555
Rank	5	3	2	1	4

Table 6.194: Effect of instance selection on Ranking Loss (↓) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.1459	0.1361	0.1386	**0.1274**	0.1483
Image	0.1579	0.1477	0.1363	**0.1314**	0.1570
Scene	0.0731	0.0653	0.0637	**0.0620**	0.0682
Yeast	0.1632	0.1560	0.1515	**0.1478**	0.1689
CAL500	0.1709	0.1662	0.1616	**0.1579**	0.1835
Average	0.1422	0.1343	0.1303	**0.1253**	0.1452
Rank	4	3	2	1	5

Table 6.195: Effect of instance selection on One Error (↓) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.2457	0.2244	0.2383	**0.2208**	0.2492
Image	0.2878	0.2852	0.2626	**0.2562**	0.2916
Scene	0.2125	0.2054	0.2010	**0.1981**	0.2050
Yeast	0.2374	0.2225	0.2233	**0.2212**	0.2378
CAL500	0.0933	0.1086	0.1025	**0.0889**	0.1160
Average	0.2153	0.2092	0.2055	**0.1970**	0.2199
Rank	4	3	2	1	5

Table 6.196: Effect of instance selection on Coverage (↓) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	1.7057	1.6585	1.6404	**1.5642**	1.7102
Image	0.8816	0.8392	0.7911	**0.7860**	0.8964
Scene	0.4493	0.4101	0.4005	**0.3898**	0.4258
Yeast	6.2042	6.0734	6.0202	**5.9470**	6.2905
CAL500	117.2167	113.1657	113.6925	**109.6889**	130.5240
Average	25.2915	24.4294	24.5089	**23.6752**	27.9694
Rank	4	2	3	1	5

Table 6.197: Effect of instance selection on Average Precision (↑) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.8194	0.8343	0.8306	**0.8421**	0.8183
Image	0.8108	0.8166	0.8312	**0.8343**	0.8105
Scene	0.8735	0.8799	0.8837	**0.8861**	0.8785
Yeast	0.7706	0.7787	0.7838	**0.7857**	0.7648
CAL500	0.5048	0.5054	0.5208	**0.5301**	0.4918
Average	0.7558	0.7630	0.7700	**0.7757**	0.7528
Rank	4	3	2	1	5

Table 6.198: Effect of instance selection on Accuracy (↑) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.5426	0.5846	0.5908	**0.6186**	0.5627
Image	0.6259	0.6244	0.6576	**0.6588**	0.6169
Scene	0.7527	0.7597	0.7600	**0.7635**	0.7599
Yeast	0.5232	0.5346	0.5413	**0.5533**	0.5140
CAL500	0.2069	0.2146	0.2255	**0.2382**	0.2023
Average	0.5303	0.5436	0.5550	**0.5665**	0.5312
Rank	5	3	2	1	4

Table 6.199: Effect of instance selection on Subset Accuracy (↑) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.2743	0.3463	0.3681	**0.3849**	0.3136
Image	0.5245	0.5196	**0.5647**	0.5564	0.5108
Scene	0.7056	**0.7149**	0.7115	0.7116	0.7117
Yeast	0.1898	0.2053	0.2176	**0.2217**	0.2046
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3388	0.3572	0.3724	**0.3749**	0.3481
Rank	5	3	2	1	4

Table 6.200: Effect of instance selection on Ex-F1 (↑) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.6330	0.6639	0.6668	**0.6953**	0.6441
Image	0.6602	0.6598	0.6892	**0.6934**	0.6532
Scene	0.7684	0.7747	0.7762	**0.7809**	0.7761
Yeast	0.6240	0.6363	0.6400	**0.6519**	0.6145
CAL500	0.3369	0.3467	0.3612	**0.3762**	0.3311
Average	0.6045	0.6163	0.6267	**0.6395**	0.6038
Rank	4	3	2	1	5

Table 6.201: Effect of instance selection on Macro-F1 (↑) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.6109	0.6546	0.6756	**0.6952**	0.6609
Image	0.6482	0.6566	0.6827	**0.6901**	0.6482
Scene	0.7657	0.7746	0.7732	0.7789	**0.7795**
Yeast	0.4416	0.4613	0.4677	**0.4815**	NaN
CAL500	NaN	NaN	NaN	NaN	NaN
Average	0.6166	0.6368	0.6498	0.6614	**0.6962**
Rank	5	4	3	2	1

TABLE 6.202: Effect of instance selection on Micro-F1 (↑) for MLFLD-MAXP

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.6575	0.6939	0.6970	**0.7177**	0.6766
Image	0.6526	0.6565	0.6829	**0.6883**	0.6449
Scene	0.7611	0.7692	0.7691	**0.7732**	0.7706
Yeast	0.6512	0.6606	0.6659	**0.6764**	0.6439
CAL500	0.3381	0.3499	0.3640	**0.3807**	0.3294
Average	0.6121	0.6260	0.6358	**0.6473**	0.6131
Rank	5	3	2	1	4

TABLE 6.203: Summary of effect of instance selection on MAXP performance

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
HamLoss	0.1572	0.1517	0.1488	**0.1446**	0.1555
RankLoss	0.1422	0.1343	0.1303	**0.1253**	0.1452
OneError	0.2153	0.2092	0.2055	**0.1970**	0.2199
Coverage	25.2915	24.4294	24.5089	**23.6752**	27.9694
AvgPrec	0.7558	0.7630	0.7700	**0.7757**	0.7528
Accuracy	0.5303	0.5436	0.5550	**0.5665**	0.5312
SubAcc	0.3388	0.3572	0.3724	**0.3749**	0.3481
Ex-F1	0.6045	0.6163	0.6267	**0.6395**	0.6038
Macro F1	0.6166	0.6368	0.6498	0.6614	**0.6962**
Micro F1	0.6121	0.6260	0.6358	**0.6473**	0.6131
Avg Rank	4.5	3.0	2.2	**1.1**	4.2
#Wins	0	0	0	**9**	1

Observations: Table 6.203 has shown that instance selection with 70, 80, and 90% replacement is more effective on MLFLD-MAXP than 60%. They worked well on all datasets except Scene for the last 5 parameters. Size 90 is seen as useful for the growing performance of MLFLD-MAXP.

6.8.3 Comparison of MLFLD and MLFLD-MAXP performance to check the effect of instance selection

How the use of sampled data affects, the performance of the proposed algorithms is studied in this section.

TABLE 6.204: Summary of effect of instance selection on MLFLD and MLFLD-MAXP performance

Metric	MLIS + MLFLD					MLIS + MAXP				
	60	70	80	90	100	60	70	80	90	100
HamLoss	0.1554	0.1496	0.1481	**0.1437**	0.1548	0.1572	0.1517	0.1488	0.1446	0.1555
RankLoss	0.1422	0.1343	0.1303	**0.1253**	0.1452	0.1422	0.1343	0.1303	**0.1253**	0.1452
OneError	0.2153	0.2092	0.2055	**0.1970**	0.2199	0.2153	0.2092	0.2055	**0.1970**	0.2199
Coverage	25.2915	24.4294	24.5089	**23.6752**	27.9694	25.2915	24.4294	24.5089	**23.6752**	27.9694
AvgPrec	0.7558	0.7630	0.7700	**0.7757**	0.7528	0.7558	0.7630	0.7700	**0.7757**	0.7528
Accuracy	0.4982	0.5123	0.5232	0.5405	0.5059	0.5303	0.5436	0.5550	**0.5665**	0.5312
SubAcc	0.3108	0.3299	0.3445	0.3524	0.3272	0.3388	0.3572	0.3724	**0.3749**	0.3481
Ex-F1	0.5710	0.5836	0.5935	0.6123	0.5769	0.6045	0.6163	0.6267	**0.6395**	0.6038
Macro F1	0.6052	0.6270	0.6376	0.6524	0.6851	0.6166	0.6368	0.6498	0.6614	**0.6962**
Micro F1	0.6714	0.6874	0.6937	0.7071	0.6757	0.6121	0.6260	0.6358	0.6473	0.6131
Avg Rank	8.1	5.8	4.5	2.2	7.6	7.5	4.8	3.6	**1.8**	7.1
#Wins	0	0	0	6	0	0	0	0	**7**	1

Observations: Table 6.204 has shown that MLFLD-MAXP has beaten MLFLD for the same sample size. When proposed algorithms are used on datasets preprocessed with instance selection with 70, 80, and 90 percent instances, increasing progress is viewed over proposed algorithms. 60% is not seemed to help for enhancement, but still better than that of contesting algorithm. MLFLD-MAXP with 90% size has outshined with the smallest avg rank 108, and 7 wins out of 10. It is followed by MLFLD with 90 showing avg rank 2.2 and 6 wins.

For accuracy, subset accuracy, and ex-F1, MLFLD-MAXP showed more progress than MLFLD after instance selection comparatively. For one error, coverage, avg precision, and rank loss, MLFLD-MAXP, and MLFLD are observed to work similarly to the same size of datasets. For macro-F1, no result computed for few datasets, hence challenging to compare. But for micro-F1 and hamming loss, MLFLD-MAXP worked well than MLFLD for the same size.

6.8.4 Comparison of effect of instance selection on proposed algorithms

Datasets are fed to the MLIS algorithm with two parameters, namely sampling with replacement and sample size 80. Obtained datasets are used for experimentation. As observed from attribute selection experimentation, MLkNN is a strong contestant among the remaining algorithms. Hence in the next two sections, only MLkNN is used for performance comparison. MLDB is used next to denote a multi-label dataset (Table 6.205 and 6.206).

Table 6.205: Performance of proposed algorithms on sampled MLDB

(a) Hamming Loss (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1937	**0.1805**	0.1812
Image	0.1557	**0.1455**	0.1466
SceneI	0.0839	**0.0802**	0.0815
Yeast	**0.1912**	0.1962	0.1962
CAL500	0.1386	0.1383	**0.1372**
Average	0.1526	**0.1481**	0.1485
Rank	3	1	2

(b) Ranking Loss (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1643	**0.1386**	**0.1386**
Image	0.1553	**0.1363**	**0.1363**
SceneI	0.0682	**0.0637**	**0.0637**
Yeast	0.1536	**0.1515**	**0.1515**
CAL500	0.1623	0.1616	**0.1603**
Average	0.1407	0.1303	**0.1301**
Rank	3	2	1

(c) One Error (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2891	**0.2383**	**0.2383**
Image	0.2969	**0.2626**	**0.2626**
SceneI	0.2094	**0.2010**	**0.2010**
Yeast	**0.1992**	0.2233	0.2233
CAL500	**0.0999**	0.1025	0.1025
Average	0.2189	**0.2055**	**0.2055**
Rank	3	1	1

(d) Coverage (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	1.7556	**1.6404**	**1.6404**
Image	0.8650	**0.7911**	**0.7911**
SceneI	0.4249	**0.4005**	**0.4005**
Yeast	6.1216	**6.0202**	**6.0202**
CAL500	113.2663	113.6925	**113.0725**
Average	24.4867	24.5089	**24.3849**
Rank	2	3	1

TABLE 6.206: Performance of proposed algorithms on sampled MLDB

(e) Average Precision (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.8003	**0.8306**	**0.8306**
Image	0.8089	**0.8312**	**0.8312**
SceneI	0.8773	**0.8837**	**0.8837**
Yeast	**0.7849**	0.7838	0.7838
CAL500	0.5176	0.5208	**0.5249**
Average	0.7578	0.7700	**0.7708**
Rank	3	2	1

(f) Accuracy (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.5367	0.5699	**0.5908**
Image	0.5289	0.5878	**0.6576**
SceneI	0.6816	0.6925	**0.7600**
Yeast	0.5269	0.5403	**0.5413**
CAL500	0.2213	0.2255	**0.2344**
Average	0.4991	0.5232	**0.5568**
Rank	3	2	1

(g) Subset Accuracy (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.3121	0.3532	**0.3681**
Image	0.4494	0.5035	**0.5647**
SceneI	0.6286	0.6484	**0.7115**
Yeast	0.1847	0.2176	**0.2176**
CAL500	0	0	0
Average	0.3150	0.3445	**0.3724**
Rank	3	2	1

(h) Ex-F1 (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6113	0.6438	**0.6668**
Image	0.5559	0.6167	**0.6892**
SceneI	0.6994	0.7073	**0.7762**
Yeast	0.6315	0.6385	**0.6400**
CAL500	0.3560	0.3612	**0.3720**
Average	0.5708	0.5935	**0.6288**
Rank	3	2	1

(i) Macro-F1 (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6253	0.6726	**0.6756**
Image	0.6210	0.6594	**0.6827**
SceneI	0.7492	0.7516	**0.7732**
Yeast	0.4151	0.4667	**0.4677**
CAL500	**0.2498**	NaN	NaN
Average	0.5321	0.6376	**0.6498**
Rank	3	2	1

(j) Micro-F1 (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6637	0.6927	**0.6970**
Image	0.6231	0.6612	**0.6829**
SceneI	0.7476	0.7552	**0.7691**
Yeast	0.6576	0.6656	**0.6659**
CAL500	0.3587	0.3640	**0.3769**
Average	0.6101	0.6277	**0.6384**
Rank	3	2	1

TABLE 6.207: Summary of MLFLD and MLFLD-MAXP performance comparison on sampled MLDB

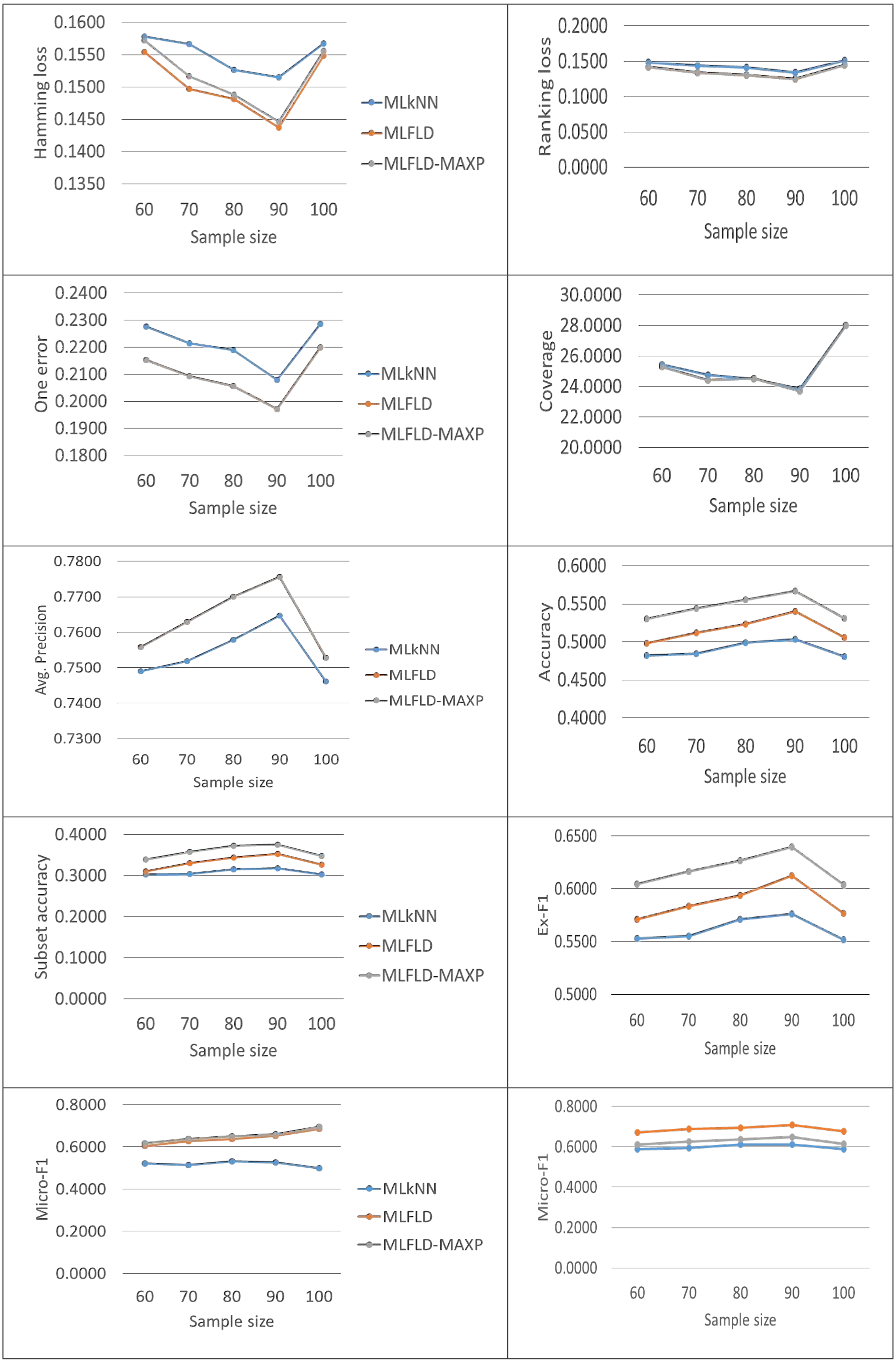
Metric	MLkNN	MLFLD	MLFLD-MAXP
HamLoss	0.1526	**0.1481**	0.1485
RankLoss	0.1407	0.1303	**0.1301**
OneError	0.2189	**0.2055**	**0.2055**
Coverage	24.4867	24.5089	**24.3849**
AvgPrec	0.7578	0.7700	**0.7708**
Accuracy	0.4991	0.5232	**0.5568**
SubAcc	0.3150	0.3445	**0.3724**
Ex-F1	0.5708	0.5935	**0.6288**
Macro F1	0.5321	0.6376	**0.6498**
Micro F1	0.6101	0.6277	**0.6384**
Avg Rank	2.9	1.9	**1.1**
#Wins	0	2	**9**

Observations: Pattern observed in Table 6.207 is slightly different than all the remaining experiments. MLFLD-MAXP exceeded MLFLD for 8 parameters. Both algorithms are similar for one error, while MLFLD is better for ham loss. Both algorithms defeated MLkNN, except for coverage by MLFLD.

6.8.5 Performance comparison of instance selection experiments with different sample sizes

The performance of MLFLD, MLFLD-MAXP, and MLkNN is compared for samples obtained after replacement with size 60, 70, 80, 90, and 100 percent. It is represented in Table 6.208. Figures (b), (c), (d), and (e) show only two algorithms as both MLFLD and MLFLD-MAXP have the same performance for corresponding parameters. All these figures show that algorithms worked well for sample sizes 80 and 90. Performance for size 90 seems superior to the performance obtained for the whole dataset (size 100).

Table 6.208: Performance comparison of instance selection experiments with different sample size



6.9 Effect of Feature and Instance selection on proposed algorithms

As seen in section 6.7, only feature selection has not proven useful for performance improvement in this work. Also, only instance selection with 70, 80, and 90 sample sizes have proven to perform better than sample size 60 when used before MLFLD and MLFLD-MAXP. This section combines both using the MLFSIS algorithm described in chapter 4.

6.9.1 Effect of Feature and Instance selection on MLFLD

In this section, instance selection is made on data for which already multi-label feature selection is carried out.

TABLE 6.209: Effect of feature and instance selection on Hamming Loss (↓) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.1876	0.1870	0.1887	**0.1855**	0.1938
Image	0.1546	0.1523	0.1486	**0.1404**	0.1631
Scene	0.0819	0.0788	**0.0787**	0.0813	0.0797
Yeast	0.2066	0.1967	0.1926	**0.1864**	0.1981
CAL500	0.1376	0.1378	0.1372	**0.1356**	0.1394
Average	0.1537	0.1505	0.1492	**0.1458**	0.1548
Rank	4	3	2	1	5

Table 6.210: Effect of feature and instance selection on Ranking Loss (↓) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.1425	0.1541	**0.1410**	**0.1410**	0.1483
Image	0.1444	0.1441	0.1389	**0.1344**	0.1570
Scene	0.0675	0.0666	0.0658	**0.0654**	0.0682
Yeast	0.1725	0.1593	0.1526	**0.1457**	0.1689
CAL500	0.1689	0.1634	0.1603	**0.1555**	0.1835
Average	0.1392	0.1375	0.1317	**0.1284**	0.1452
Rank	4	3	2	1	5

Table 6.211: Effect of feature and instance selection on One Error (↓) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	**0.2229**	0.2390	0.2234	0.2340	0.2492
Image	0.2702	0.2766	0.2652	**0.2568**	0.2916
Scene	0.2049	0.2083	0.2016	**0.1986**	0.2050
Yeast	0.2402	0.2201	**0.2104**	0.2106	0.2378
CAL500	0.1000	**0.0914**	0.1025	0.1089	0.1160
Average	0.2076	0.2071	**0.2006**	0.2018	0.2199
Rank	4	3	1	2	5

Table 6.212: Effect of feature and instance selection on Coverage (↓) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	1.7229	1.7585	1.6745	**1.6358**	1.7102
Image	0.8265	0.8214	0.8023	**0.7910**	0.8964
Scene	0.4215	0.4149	0.4125	**0.4088**	0.4258
Yeast	6.3672	6.1633	6.0793	**5.8995**	6.2905
CAL500	117.1300	112.4486	113.0725	**109.2311**	130.5240
Average	25.2936	24.3213	24.4082	**23.5932**	27.9694
Rank	4	2	3	1	5

Table 6.213: Effect of feature and instance selection on Average Precision (↑) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.8293	0.8157	**0.8322**	0.8291	0.8183
Image	0.8253	0.8234	0.8298	**0.8351**	0.8105
Scene	0.8793	0.8785	0.8816	**0.8827**	0.8785
Yeast	0.7618	0.7778	0.7860	**0.7910**	0.7648
CAL500	0.5103	0.5182	0.5249	**0.5319**	0.4918
Average	0.7612	0.7627	0.7709	**0.7740**	0.7528
Rank	4	3	2	1	5

Table 6.214: Effect of feature and instance selection on Accuracy (↑) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.5526	0.5492	0.5649	**0.5759**	0.5483
Image	0.5656	0.5720	0.6016	**0.6118**	0.5588
Scene	0.7023	0.6990	**0.7118**	0.7001	0.7083
Yeast	0.5112	0.5306	0.5352	**0.5583**	0.5116
CAL500	0.2141	0.2260	0.2344	**0.2439**	0.2023
Average	0.5092	0.5154	0.5296	**0.5380**	0.5059
Rank	4	3	2	1	5

Table 6.215: Effect of feature and instance selection on Subset Accuracy (↑) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.3029	0.3049	**0.3447**	0.3396	0.3051
Image	0.4787	0.4860	0.5097	**0.5186**	0.4632
Scene	0.6535	0.6577	**0.6661**	0.6514	0.6629
Yeast	0.1939	0.2112	0.2155	**0.2290**	0.2046
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3258	0.3320	0.3472	**0.3477**	0.3272
Rank	5	3	2	1	4

Table 6.216: Effect of feature and instance selection on Ex-F1 (↑) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.6363	0.6338	0.6417	**0.6543**	0.6274
Image	0.5953	0.6011	0.6329	**0.6432**	0.5916
Scene	0.7187	0.7129	**0.7272**	0.7164	0.7235
Yeast	0.6102	0.6288	0.6326	**0.6552**	0.6109
CAL500	0.3457	0.3617	0.3720	**0.3838**	0.3311
Average	0.5812	0.5877	0.6013	**0.6106**	0.5769
Rank	4	3	2	1	5

Table 6.217: Effect of feature and instance selection on Macro-F1 (↑) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.6319	0.6453	0.6584	**0.6639**	0.6584
Image	0.6316	0.6375	0.6634	**0.6805**	0.6287
Scene	0.7577	0.7599	0.7659	0.7568	**0.7683**
Yeast	0.4378	0.4617	0.4880	**0.4976**	NaN
CAL500	NaN	NaN	NaN	NaN	NaN
Average	0.6148	0.6261	0.6439	0.6497	**0.6851**
Rank	5	4	3	2	1

Table 6.218: Effect of feature and instance selection on Micro-F1 (↑) for MLFLD

Dataset	MLFSIS + MLFLD				
	60	70	80	90	100
Emotions	0.6761	0.6785	0.6822	**0.6887**	0.6727
Image	0.6386	0.6443	0.6655	**0.6808**	0.6259
Scene	0.7551	0.7612	**0.765**0	0.7559	0.7617
Yeast	0.6402	0.6578	0.6635	**0.6822**	0.6426
CAL500	0.3476	0.3647	0.3769	**0.3890**	0.3294
Average	0.6775	0.6855	0.6941	**0.7019**	0.6757
Rank	4	3	2	1	5

Table 6.219: Summary of effect of feature and instance selection on MLFLD performance

Metric	MLFSIS + MLFLD				
	60	70	80	90	100
HamLoss	0.1537	0.1505	0.1492	0.1458	**0.1548**
RankLoss	0.1392	0.1375	0.1317	**0.1284**	0.1452
OneError	0.2076	0.2071	**0.2006**	0.2018	0.2199
Coverage	25.2936	24.3213	24.4082	**23.5932**	27.9694
AvgPrec	0.7612	0.7627	0.7709	**0.7740**	0.7528
Accuracy	0.5092	0.5154	0.5296	**0.5380**	0.5059
SubAcc	0.3258	0.332	0.3472	**0.3477**	0.3272
Ex-F1	0.5812	0.5877	0.6013	**0.6106**	0.5769
Macro F1	0.6148	0.6261	0.6439	0.6497	**0.6851**
Micro F1	0.6775	0.6855	0.6941	**0.7019**	0.6757
Avg Rank	4.2	3.0	2.1	**1.2**	4.5
#Wins	0	0	1	**8**	1

Observations: From Table 6.219, datasets preprocessed with multi-label feature and instance selection (MLFSIS) algorithm are worthwhile for upgrading MLFLD functionality for all sample sizes used. 80 and 90% seemed more appropriate for most of the cases. Size 90 is more effective, showing the smallest avg rank 1.2 and maximum wins 8.

6.9.2 Effect of Feature and Instance selection on MLFLD-MAXP

In this section, the functionality of MLFLD-MAXP is examined on datasets preprocessed by the MLFSIS algorithm on 5 datasets.

Table 6.220: Effect of feature and instance selection on Hamming Loss (↓) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	**0.1833**	0.1894	0.1879	0.1862	0.1938
Image	0.1538	0.1528	0.1508	**0.1465**	0.1656
Scene	0.0815	0.0817	**0.0806**	0.0812	0.0812
Yeast	0.2066	0.1965	0.1928	**0.1865**	0.1977
CAL500	0.1376	0.1378	0.1372	**0.1356**	0.1394
Average	0.1526	0.1516	0.1499	**0.1472**	0.1555
Rank	4	3	2	1	5

Table 6.221: Effect of feature and instance selection on Ranking Loss (↓) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.1425	0.1541	**0.1410**	**0.1410**	0.1483
Image	0.1444	0.1441	0.1389	**0.1344**	0.1570
Scene	0.0675	0.0666	0.0658	**0.0654**	0.0682
Yeast	0.1725	0.1593	0.1526	**0.1457**	0.1689
CAL500	0.1689	0.1634	0.1603	**0.1555**	0.1835
Average	0.1392	0.1375	0.1317	**0.1284**	0.1452
Rank	4	3	2	1	5

TABLE 6.222: Effect of feature and instance selection on One Error (↓) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	**0.2229**	0.2390	0.2234	0.2340	0.2492
Image	0.2702	0.2766	0.2652	**0.2568**	0.2916
Scene	0.2049	0.2083	0.2016	**0.1986**	0.2050
Yeast	0.2402	0.2201	**0.2104**	0.2106	0.2378
CAL500	0.1000	**0.0914**	0.1025	0.1089	0.1160
Average	0.2076	0.2071	**0.2006**	0.2018	0.2199
Rank	4	3	1	2	5

TABLE 6.223: Effect of feature and instance selection on Coverage (↓) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	1.7229	1.7585	1.6745	**1.6358**	1.7102
Image	0.8265	0.8214	0.8023	**0.7910**	0.8964
Scene	0.4215	0.4149	0.4125	**0.4088**	0.4258
Yeast	6.3672	6.1633	6.0793	**5.8995**	6.2905
CAL500	117.1300	112.4486	113.0725	**109.2311**	130.5240
Average	25.2936	24.3213	24.4082	**23.5932**	27.9694
Rank	4	2	3	1	5

TABLE 6.224: Effect of feature and instance selection on Average Precision (↑) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.8293	0.8157	**0.8322**	0.8291	0.8183
Image	0.8253	0.8234	0.8298	**0.8351**	0.8105
Scene	0.8793	0.8785	0.8816	**0.8827**	0.8785
Yeast	0.7618	0.7778	0.7860	**0.7910**	0.7648
CAL500	0.5103	0.5182	0.5249	**0.5319**	0.4918
Average	0.7612	0.7627	0.7709	**0.7740**	0.7528
Rank	4	3	2	1	5

TABLE 6.225: Effect of feature and instance selection on Accuracy (↑) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.5802	0.5630	**0.5848**	0.5844	0.5627
Image	0.6415	0.6443	0.6529	**0.6588**	0.6169
Scene	0.7596	0.7585	**0.7628**	0.7600	0.7599
Yeast	0.5127	0.5334	0.5369	**0.5593**	0.5140
CAL500	0.2141	0.2260	0.2344	**0.2439**	0.2023
Average	0.5416	0.5450	0.5544	**0.5613**	0.5312
Rank	4	3	2	1	5

TABLE 6.226: Effect of feature and instance selection on Subset Accuracy (↑) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.3143	0.3098	**0.3511**	0.3434	0.3136
Image	0.5446	0.5475	0.5560	**0.5603**	0.5108
Scene	0.7063	**0.7143**	0.7141	0.7079	0.7117
Yeast	0.1939	0.2124	0.2155	**0.2290**	0.2046
CAL500	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.3518	0.3568	0.3673	**0.3681**	0.3481
Rank	4	3	2	1	5

TABLE 6.227: Effect of feature and instance selection on Ex-F1 (↑) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	**0.6696**	0.6509	0.6662	0.6644	0.6441
Image	0.6746	0.6768	0.6859	**0.6920**	0.6532
Scene	0.7775	0.7734	**0.7793**	0.7775	0.7761
Yeast	0.6124	0.6326	0.6351	**0.6566**	0.6145
CAL500	0.3457	0.3617	0.3720	**0.3838**	0.3311
Average	0.6160	0.6191	0.6277	**0.6349**	0.6038
Rank	4	3	2	1	5

TABLE 6.228: Effect of feature and instance selection on Macro-F1 (↑) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.6468	0.6489	**0.6665**	0.6658	0.6609
Image	0.6662	0.6704	0.6781	**0.7289**	0.6482
Scene	0.7747	0.7723	0.7774	0.7756	**0.7795**
Yeast	0.4382	0.4625	0.4887	**0.4985**	NaN
CAL500	NaN	NaN	NaN	NaN	NaN
Average	0.6315	0.6385	0.6527	0.6672	**0.6962**
Rank	5	4	3	2	1

TABLE 6.229: Effect of feature and instance selection on Micro-F1 (↑) for MLFLD-MAXP

Dataset	MLFSIS + MLFLD-MAXP				
	60	70	80	90	100
Emotions	0.6901	0.6807	0.6894	**0.6903**	0.6766
Image	0.6669	0.6700	0.6789	**0.6865**	0.6449
Scene	0.7707	0.7689	**0.7731**	0.7707	0.7706
Yeast	0.6406	0.6587	0.6637	**0.6823**	0.6439
CAL500	0.3476	0.3647	0.3769	**0.3890**	0.3294
Average	0.6232	0.6286	0.6364	**0.6438**	0.6131
Rank	4	3	2	1	5

TABLE 6.230: Summary of effect of feature and instance selection on MLFLD-MAXP performance

Dataset	MLIS + MLFLD-MAXP				
	60	70	80	90	100
HamLoss	0.1526	0.1516	0.1499	**0.1472**	0.1555
RankLoss	0.1392	0.1375	0.1317	**0.1284**	0.1452
OneError	0.2076	0.2071	**0.2006**	0.2018	0.2199
Coverage	25.2936	24.3213	24.4082	**23.5932**	27.9694
AvgPrec	0.7612	0.7627	0.7709	**0.7740**	0.7528
Accuracy	0.5416	0.5450	0.5544	**0.5613**	0.5312
SubAcc	0.3518	0.3568	0.3673	**0.3681**	0.3481
Ex-F1	0.6160	0.6191	0.6277	**0.6349**	0.6038
Macro F1	0.6315	0.6385	0.6527	0.6672	**0.6962**
Micro F1	0.6232	0.6286	0.6364	**0.6438**	0.6131
Avg Rank	4.1	3.0	2.1	**1.2**	4.6
#Wins	0	0	1	**8**	1

Observations: From Table 6.230, it is noticed that feature and instance selection with replacement is useful for elevating MLFLD-MAXP functionality for all sample sizes compared to the whole dataset. Best performance is obtained for 90 with minimum avg rank 1.2 and maximum wins 8. Performance decreases with sizes 80, 70, 60, and 100.

6.9.3 Comparison of MLFLD and MLFLD-MAXP performance to check the effect of Feature and Instance selection

In this section, the performance of MLFLD and MLFLD-MAXP on selected features and sampled data is studied. It is compared with the execution of algorithms on non-processed datasets.

TABLE 6.231: Summary of effect of feature and instance selection on MLFLD and MLFLD-MAXP performance

Metric	MLIS + MLFLD					MLIS + MAXP				
	60	70	80	90	100	60	70	80	90	100
HamLoss	0.1537	0.1505	0.1492	**0.1458**	0.1548	0.1526	0.1516	0.1499	0.1472	0.1555
RankLoss	0.1392	0.1375	0.1317	**0.1284**	0.1452	0.1392	0.1375	0.1317	**0.1284**	0.1452
OneError	0.2076	0.2071	**0.2006**	0.2018	0.2199	0.2076	0.2071	**0.2006**	0.2018	0.2199
Coverage	25.2936	24.3213	24.4082	**23.5932**	27.9694	25.2936	24.3213	24.4082	**23.5932**	27.9694
AvgPrec	0.7612	0.7627	0.7709	**0.7740**	0.7528	0.7612	0.7627	0.7709	**0.7740**	0.7528
Accuracy	0.5092	0.5154	0.5296	0.5380	0.5059	0.5416	0.5450	0.5544	**0.5613**	0.5312
SubAcc	0.3258	0.3320	0.3472	0.3477	0.3272	0.3518	0.3568	0.3673	**0.3681**	0.3481
Ex-F1	0.5812	0.5877	0.6013	0.6106	0.5769	0.6160	0.6191	0.6277	**0.6349**	0.6038
Macro F1	0.6148	0.6261	0.6439	0.6497	0.6851	0.6315	0.6385	0.6527	0.6672	**0.6962**
Micro F1	0.6775	0.6855	0.6941	**0.7019**	0.6757	0.6232	0.6286	0.6364	0.6438	0.6131
Avg Rank	7.8	5.9	4.4	**2.9**	8.1	6.4	4.8	3.3	2.0	7.4
#Wins	0	0	1	**5**	0	0	0	1	6	1

Observations: From Table 6.231, feature and instance selection are noticed to be very useful for upgrading the performance of proposed algorithms over only feature or instance selection. Also, MLFLD-MAXP has beaten MLFLD when compared with each other for the same sample sizes. Experiment with 90% sample size revealed to be the most appropriate among all, followed by 80%. MLFLD-MAXP with 90% size got minimum avg rank 2 and max. wins 6.

6.9.4 Effect of Feature and Instance selection on proposed algorithms compared with MLkNN

The previous two sections are based on experiments involving either attribute or instance selection. Both are useful to reduce the dimension of a dataset in a different direction. In this section, the effect of both these operations is combined to see the impact on multi-label datasets. Experiments are done using sampling with and without replacement. Later gave better performance.

Table 6.232: Effect of feature and instance selection for proposed algorithms compared with MLkNN

(a) *Hamming loss* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1947	0.1887	**0.1879**
Image	0.1649	**0.1486**	0.1508
Scene	0.0869	**0.0787**	0.0806
Yeast	**0.1905**	0.1926	0.1928
CAL500	**0.1365**	0.1372	0.1372
Average	0.1547	**0.1492**	**0.1499**
Rank	3	1	2

(b) *Ranking loss* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.1554	**0.1410**	**0.1410**
Image	0.1540	**0.1389**	**0.1389**
Scene	0.0717	**0.0658**	**0.0658**
Yeast	**0.1526**	**0.1526**	**0.1526**
CAL500	0.1611	**0.1603**	**0.1603**
Average	0.1390	**0.1317**	**0.1317**
Rank	3	1	1

(c) *One Error* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2615	**0.2234**	**0.2234**
Image	0.3063	**0.2652**	**0.2652**
Scene	0.2244	**0.2016**	**0.2016**
Yeast	**0.2059**	0.2104	0.2104
CAL500	**0.0923**	0.1025	0.1025
Average	0.2181	**0.2006**	**0.2006**
Rank	3	1	1

(d) *Coverage* (↓)

Dataset	MLkNN	MLFLD	MAXP
Emotions	1.7479	**1.6745**	**1.6745**
Image	0.8631	**0.8023**	**0.8023**
Scene	0.4415	**0.4125**	**0.4125**
Yeast	6.1086	**6.0793**	**6.0793**
CAL500	113.2037	**113.0725**	**113.0725**
Average	24.4730	**24.4082**	**24.4082**
Rank	3	1	1

Also, different sample sizes are used during execution like 60, 70, 80, and 90. Sizes 60 and 70 are suitable for some datasets only. 80 and 90 are always viewed better on almost all datasets. Results of size 80 are used for comparison further. Results are shown in Table 6.232 and 6.233.

TABLE 6.233: Effect of feature and instance selection for proposed algorithms compared with MLkNN

(e) *Average Precision* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.8071	**0.8322**	**0.8322**
Image	0.8052	**0.8298**	**0.8298**
Scene	0.8699	**0.8816**	**0.8816**
Yeast	0.7852	**0.7860**	**0.7860**
CAL500	0.5240	**0.5249**	**0.5249**
Average	0.7583	**0.7709**	**0.7709**
Rank	3	1	1

(f) *Accuracy* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.5233	0.5649	**0.5848**
Image	0.5279	0.6016	**0.6529**
Scene	0.6681	0.7118	**0.7628**
Yeast	0.5275	0.5352	**0.5369**
CAL500	0.2309	**0.2344**	**0.2344**
Average	0.4955	0.5296	**0.5544**
Rank	3	2	1

(g) *Subset Accuracy* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.2721	0.3447	**0.3511**
Image	0.4475	0.5097	**0.5560**
Scene	0.6239	0.6661	**0.7141**
Yeast	0.1961	**0.2155**	**0.2155**
CAL500	0.0000	0.0000	0.0000
Average	0.3079	0.3472	**0.3673**
Rank	3	2	1

(h) *Ex-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6049	0.6417	**0.6662**
Image	0.5552	0.6329	**0.6859**
Scene	0.6831	0.7272	**0.7793**
Yeast	0.6297	0.6326	**0.6351**
CAL500	0.3680	**0.3720**	**0.3720**
Average	0.5682	0.6013	**0.6277**
Rank	3	2	1

(i) *Macro-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6183	0.6584	**0.6665**
Image	0.6042	0.6634	**0.6781**
Scene	0.7345	0.7659	**0.7774**
Yeast	0.4212	0.4880	**0.4887**
CAL500	0.2583	NaN	NaN
Average	0.5273	0.6439	0.6527
Rank	3	2	1

(j) *Micro-F1* (↑)

Dataset	MLkNN	MLFLD	MAXP
Emotions	0.6571	0.6822	**0.6894**
Image	0.6066	0.6655	**0.6789**
Scene	0.7346	0.7650	**0.7731**
Yeast	0.6580	0.6635	**0.6637**
CAL500	0.3709	**0.3769**	**0.3769**
Average	0.6054	0.6306	0.6364
Rank	3	2	1

Table 6.234: Summary of comparison of feature and instance selection on proposed algorithms

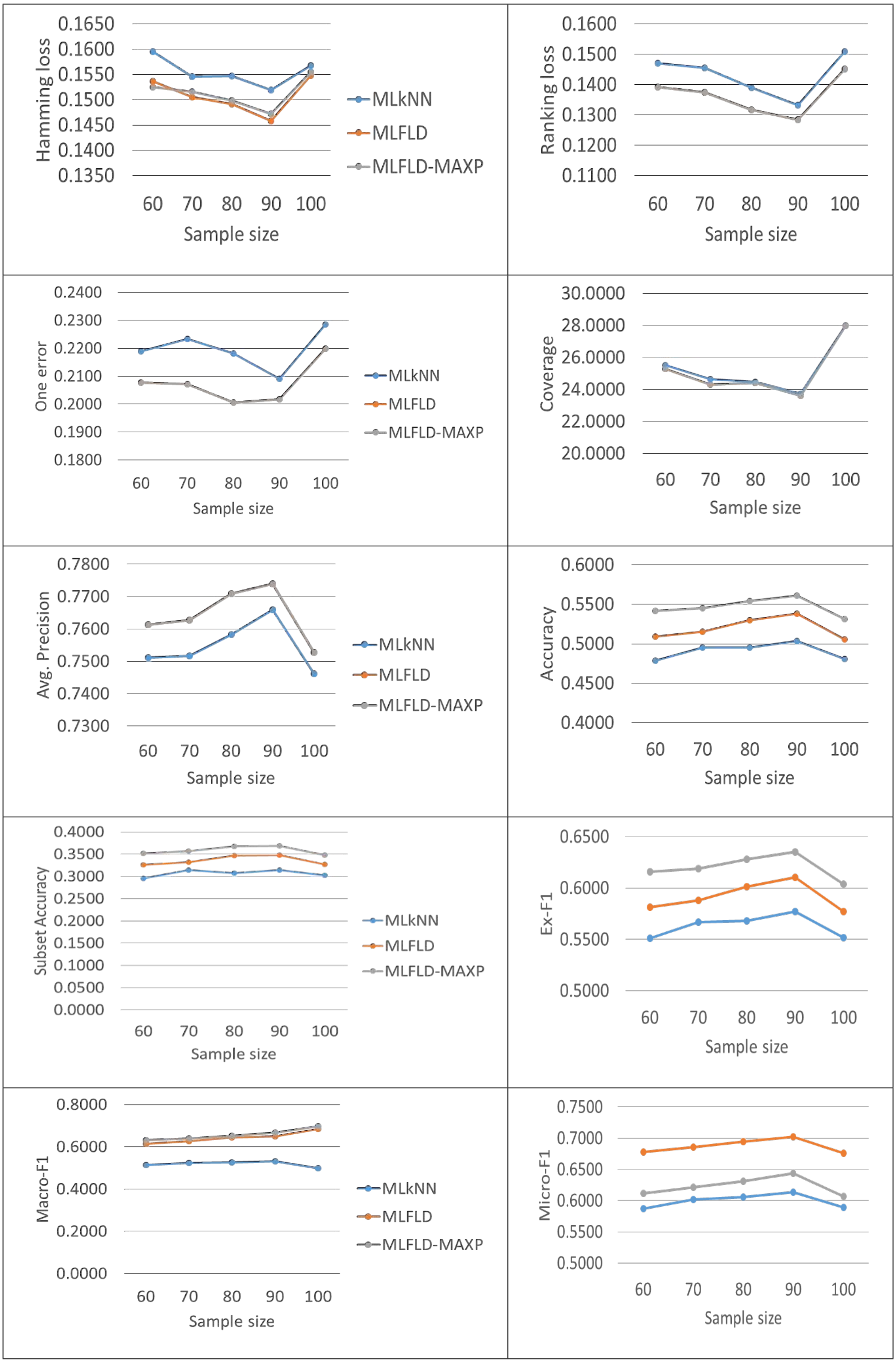
Metric	MLFSIS+ MLkNN	MLFSIS + MLFLD	MLFSIS + MAXP
HamLoss	0.1547	**0.1492**	0.1499
RankLoss	0.1390	**0.1317**	**0.1317**
OneError	0.2181	**0.2006**	**0.2006**
Coverage	24.473	**24.4082**	**24.4082**
AvgPrec	0.7583	**0.7709**	**0.7709**
Accuracy	0.4955	0.5296	**0.5544**
SubAcc	0.3079	0.3472	**0.3673**
Ex-F1	0.5682	0.6013	**0.6277**
Macro F1	0.5273	0.6439	**0.6527**
Micro F1	0.6054	0.6306	**0.6364**
Avg Rank	3.0	1.5	**1.1**
#Wins	0	5	**9**

Observations: Again, MLFLD-MAXP achieved better avg rank 1.1 with 9 on 10 wins, whereas MLFLD stood second with avg rank 1.5 and 5 wins over MLkNN as shown Table 6.234.

6.9.5 Comparison of feature and instance selection experiments for different sample sizes

Similar to instance selection, the performance of three algorithms are compared for a feature and instance selection experiments for 60-100% instances. Again Figures (b)-(e) show only two algorithms as both MLFLF-MAXP and MLFLD have the same behavior for the corresponding metric. 90% of instances are examined to work better among 5 sizes, followed by 80%. Both 90 and 80 performed better than size 100.

TABLE 6.235: Comparison of feature and instance selection experiments for different sample sizes



When all the experiments in the last three sections are examined, it is noticed that the micro-F1 metric is improved the most with feature selection. Subset accuracy and accuracy are more improved by feature and instance selection experiments. Remaining all measures are raised slightly.

6.10 Effect of k variation on proposed algorithm MLFLD

The number of neighbors, k, has always remained an essential point in k nearest neighbors (kNN) classifier. But in the case of multi-label classifiers based on kNN, the scenario is different. While doing the experimentation of MLFLD, k is varied from 5 to 15. The performance of four datasets is examined as shown in Table 6.236 to 6.239, that is also shown graphically in Figure 6.24 to 6.27, respectively. Increased performance for a metric is marked by bold value in each column. It can be seen that "k" has less effect on the performance of MLFLD. Performance metrics show very slightly or no variation with that of k. From these observations and sources from the literature [20] [12] [37] [42] [89], the value of k used for the remaining experimentation is 10.

6.10.1 Effect of k variation on MLFLD using Emotions dataset

Table 6.236 shows that out of 10, 6 parameters show improvement for a higher value of k, and the remaining 4 parameters show an increase for a lower value of k. Hence to keep a balance between all metrics, value 10 is marked better for parameter k, which shows performance near to average. The same is depicted in Figure 6.24. Note that coverage values are scaled between 0 to 1 range in the graph.

TABLE 6.236: Effect of k variation on MLFLD using Emotions dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
5	0.1929	0.1590	0.2559	1.7678	0.8074	**0.5666**	0.3051	**0.6524**	0.6659	**0.6843**
6	0.1969	0.1596	0.2593	1.7763	0.8060	0.5564	0.3102	0.6392	0.6615	0.6767
7	0.1910	0.1598	0.2847	1.7610	0.7990	0.5638	**0.3288**	0.6407	0.6636	0.6823
8	0.1944	0.1602	0.2746	1.7780	0.7998	0.5582	0.3136	0.6371	0.6666	0.6773
9	0.1918	0.1557	0.2627	1.7492	0.8096	0.5613	0.3186	0.6418	0.6595	0.6796
10	0.1938	0.1483	0.2492	1.7102	0.8183	0.5483	0.3051	0.6274	0.6584	0.6727
11	0.1958	0.1474	0.2508	1.7153	0.8167	0.5444	0.3051	0.6235	0.6531	0.6685
12	0.1907	0.1512	0.2559	1.7339	0.8136	0.5561	0.3153	0.6357	0.6659	0.6815
13	0.1935	**0.1455**	**0.2407**	1.7068	**0.8198**	0.5558	0.3186	0.6338	0.6640	0.6771
14	**0.1876**	0.1474	0.2559	1.7000	0.8165	0.5602	0.3237	0.6379	**0.6706**	0.6838
15	0.1932	0.1457	0.2525	**1.6898**	0.8170	0.5499	0.3136	0.6269	0.6622	0.6740
Avg	0.1929	0.1527	0.2584	1.7353	0.8112	0.5565	0.3143	0.6360	0.6628	0.6780

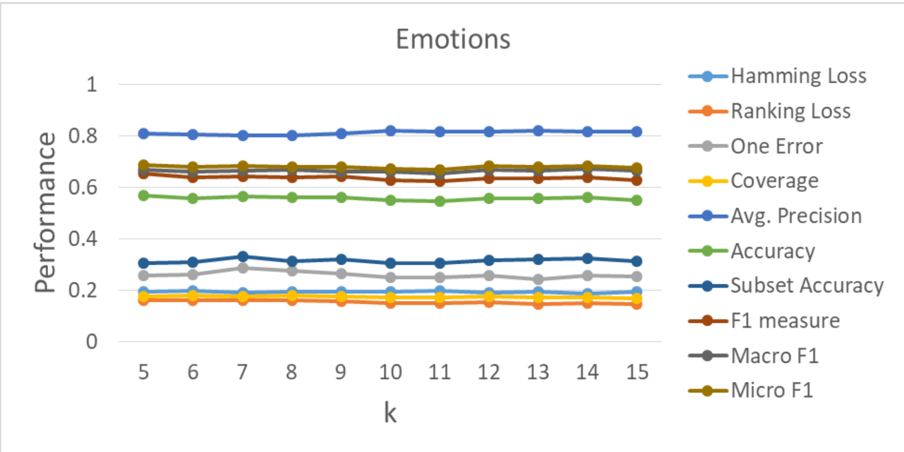


FIGURE 6.24: Effect of k variation on MLFLD using Emotions dataset

6.10.2 Effect of k variation on MLFLD using Scene dataset

Table 6.237 shows that nine metrics show enhancement for a k value 13, and Macro-F1 shows growth for k 11 that is very close to performance for k 13. But working for a k value 10 and 13 is seen similar, which is very close to the average performance. The same is depicted in Figure 6.25.

Table 6.237: Effect of k variation on MLFLD using Scene dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
5	0.0837	0.0734	0.2138	0.4525	0.8733	0.6935	0.6538	0.7069	0.7534	0.7473
6	0.0809	0.0740	0.2096	0.4563	0.8742	0.6967	0.6583	0.7096	0.7591	0.7531
7	0.0804	0.0715	0.2071	0.4450	0.8762	0.7008	0.6600	0.7144	0.7611	0.7561
8	0.0793	0.0697	0.2017	0.4354	0.8792	0.7065	0.6637	0.7208	0.7674	0.7609
9	0.0785	0.0693	0.2050	0.4333	0.8777	0.7021	0.6608	0.7159	0.7654	0.7610
10	0.0797	0.0682	0.2050	0.4258	0.8785	0.7083	0.6629	0.7235	0.7683	0.7617
11	0.0785	0.0676	0.2029	0.4217	0.8805	0.7149	0.6713	0.7296	**0.7736**	0.7656
12	0.0795	0.0674	0.2046	0.4208	0.8800	0.7076	0.6625	0.7228	0.7679	0.7613
13	**0.0783**	**0.0646**	**0.1987**	**0.4058**	**0.8836**	**0.7176**	**0.6725**	**0.7327**	0.7732	**0.7663**
14	0.0789	0.0650	0.1987	0.4108	0.8827	0.7151	0.6679	0.7309	0.7704	0.7652
15	0.0797	0.0661	0.2050	0.4162	0.8799	0.7119	0.6629	0.7283	0.7680	0.7621
Avg	0.0798	0.0688	0.2047	0.4294	0.8787	0.7068	0.6633	0.7214	0.7662	0.7601

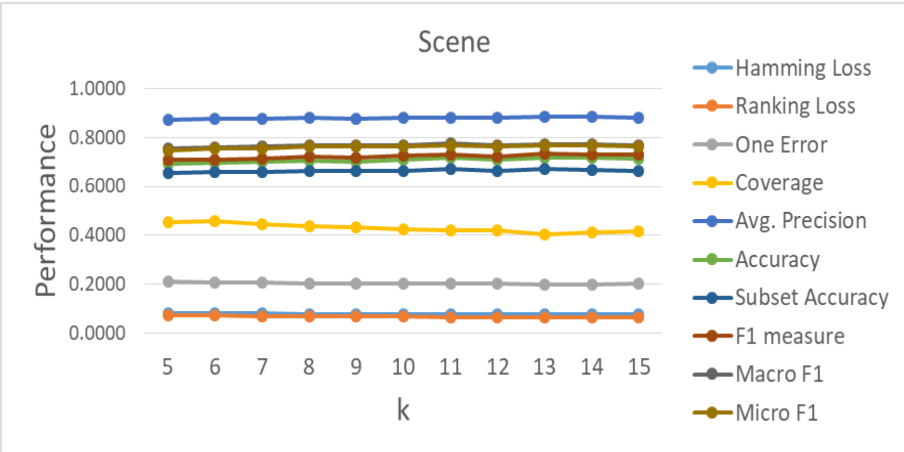


Figure 6.25: Effect of k variation on MLFLD using Scene dataset

6.10.3 Effect of k variation on MLFLD using an Image dataset

Table 6.238 shows that seven metrics have shown improvement for a k value above ten, and the remaining three metrics for a k value below 10. Hence k value ten is viewed better to keep the balance between performances of parameters. The same is depicted in Figure 6.26.

TABLE 6.238: Effect of k variation on MLFLD using Image dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
5	0.1667	0.1724	0.2996	0.9580	0.8018	0.5338	0.4467	0.5636	0.6113	0.6093
6	0.1605	0.1655	0.2911	0.9345	0.8069	0.5524	0.4577	0.5848	0.6265	0.6249
7	0.1622	0.1629	0.2861	0.9255	0.8091	0.5599	0.4587	0.5947	0.6312	0.6294
8	0.1619	0.1598	0.2876	0.9095	0.8106	**0.5692**	**0.4702**	**0.6031**	0.6347	0.6329
9	0.1624	0.1588	0.2871	0.9089	0.8110	0.5410	0.4497	0.5722	0.6180	0.6165
10	0.1631	0.1570	0.2916	0.8964	0.8105	0.5588	0.4632	0.5916	0.6287	0.6259
11	**0.1594**	**0.1538**	**0.2831**	0.8884	0.8146	0.5547	0.4572	0.5878	0.6285	0.6281
12	0.1608	0.1542	0.2846	**0.8854**	**0.8148**	0.5626	0.4612	0.5971	0.6333	0.6314
13	0.1614	0.1555	0.2896	0.8944	0.8112	0.5582	0.4572	0.5928	0.6291	0.6293
14	0.1602	0.1569	0.2846	0.9009	0.8124	0.5647	0.4577	0.6010	**0.6376**	**0.6349**
15	0.1622	0.1549	0.2846	0.8914	0.8134	0.5575	0.4497	0.5943	0.6309	0.6294
Avg	0.1619	0.1592	0.2881	0.9085	0.8106	0.5557	0.4572	0.5894	0.6282	0.6265

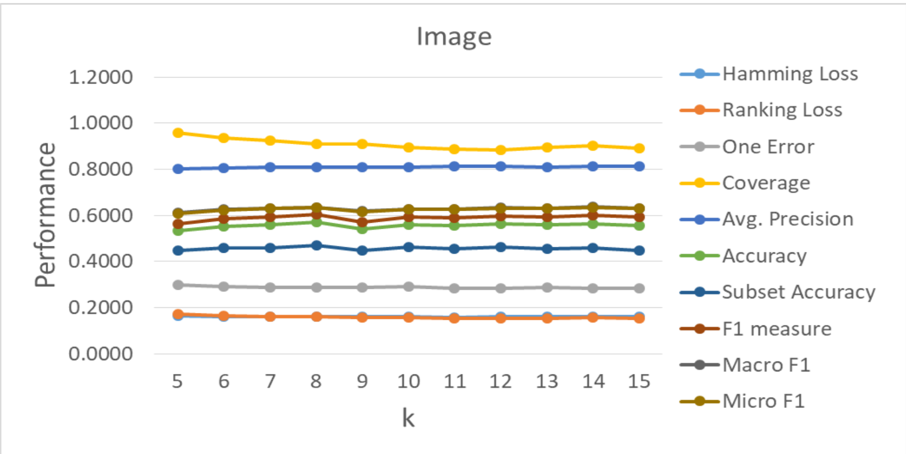


FIGURE 6.26: Effect of k variation on MLFLD using Image dataset

6.10.4 Effect of k variation on MLFLD using Yeast dataset

Table 6.239 shows that 5 metrics show growth for a k value above 10, and the remaining 5 parameters show improvement for a k value below 10. Hence k value 10 is marked better to keep a balance between performances of metrics that also appeared very close to average overall the k values. The same is depicted in Figure 6.27.

TABLE 6.239: Effect of k variation on MLFLD using Yeast dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
5	0.2015	0.1714	0.2494	6.2651	0.7612	0.5180	**0.2124**	0.6152	NaN	0.6442
6	**0.1971**	0.1723	0.2461	6.3133	0.7608	0.5073	0.1963	0.6064	NaN	0.6395
7	0.1991	0.1719	0.2390	6.3071	0.7619	0.5197	0.2104	0.6190	NaN	0.6467
8	0.2011	0.1708	0.2444	6.3037	0.7623	0.5070	0.2004	0.6055	NaN	0.6366
9	0.2004	0.1688	**0.2303**	6.2884	**0.7660**	0.5118	0.2012	0.6121	NaN	0.6410
10	0.1981	0.1689	0.2378	6.2905	0.7648	0.5116	0.2046	0.6109	NaN	0.6426
11	0.1983	0.1684	0.2361	6.2689	0.7643	0.5192	0.2100	0.6178	NaN	0.6477
12	0.1990	**0.1679**	0.2357	6.2544	0.7659	0.5081	0.2066	0.6059	NaN	0.6393
13	0.2004	0.1679	0.2361	6.2627	0.7657	0.5089	0.1979	0.6090	NaN	0.6399
14	0.1985	0.1683	0.2394	**6.2419**	0.7656	0.5190	0.2029	0.6197	NaN	0.6470
15	0.1977	0.1682	0.2398	6.2631	0.7647	**0.5235**	0.2075	**0.6241**	NaN	**0.6507**
Avg	0.1992	0.1695	0.2395	6.2781	0.7639	0.5140	0.2046	0.6132	NaN	0.6432

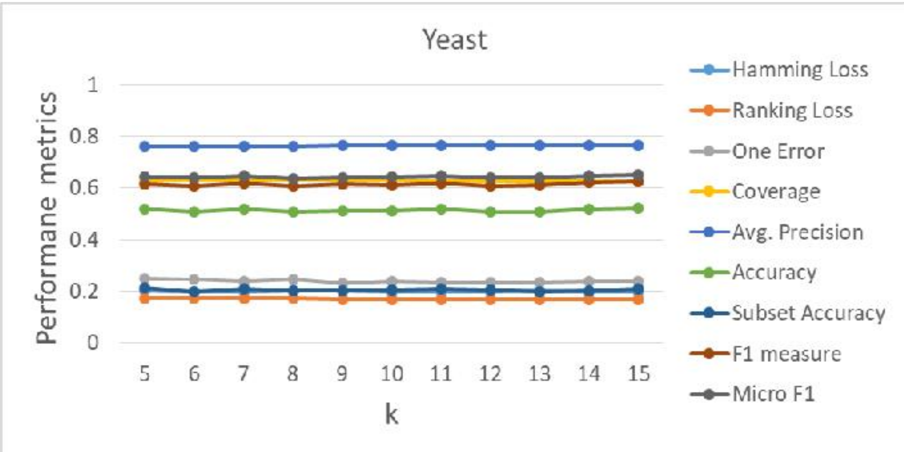


FIGURE 6.27: Effect of k variation on MLFLD using Yeast dataset

6.11 Effect of threshold variation on proposed algorithm MLFLD

A threshold is a significant parameter in the MLFLD algorithm. Whenever labels are to be predicted for an unseen instance, then a probability is calculated for each label. If it is above threshold t for a particular label c, then that label c is said to be associated with instance under consideration. The performance of four datasets is examined, as shown in Table 6.240 to 6.243. The threshold is varied from 0.3 to 0.7. As threshold value is increased, evaluation metrics, namely ranking loss, one error, coverage, and average precision, show stable performance. These metrics are not included in Figures 6.28 to 6.31. The remaining metrics show varying performance.

6.11.1 Effect of threshold variation on Emotions dataset

For the Emotions dataset, threshold 0.5 has shown minimum hamming loss and performance better than the average value for the last 5 metrics. Threshold 0.3 has shown better accuracy and harmonic means but results in a more hamming loss. Threshold values near 0.7 have not performed well.

TABLE 6.240: Effect of threshold variation on MLFLD using Emotions dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.3	0.2093	0.1483	0.2492	1.7102	0.8183	**0.5877**	0.2932	**0.6818**	**0.6906**	**0.6992**
0.35	0.2054	0.1483	0.2492	1.7102	0.8183	0.5813	0.3017	0.6719	0.6803	0.6924
0.4	0.2025	0.1483	0.2492	1.7102	0.8183	0.5770	**0.3220**	0.6621	0.6754	0.6880
0.45	0.1941	0.1483	0.2492	1.7102	0.8183	0.5645	**0.3220**	0.6446	0.6648	0.6807
0.5	**0.1938**	0.1483	0.2492	1.7102	0.8183	0.5483	0.3051	0.6274	0.6584	0.6727
0.55	0.1977	0.1483	0.2492	1.7102	0.8183	0.5304	0.2932	0.6081	0.6439	0.6591
0.6	0.1983	0.1483	0.2492	1.7102	0.8183	0.5090	0.2746	0.5866	0.6304	0.6457
0.65	0.2051	0.1483	0.2492	1.7102	0.8183	0.4679	0.2542	0.5379	0.5948	0.6101
0.7	0.2065	0.1483	0.2492	1.7102	0.8183	0.4340	0.2271	0.5018	0.5579	0.5840
Avg	0.2014	0.1483	0.2492	1.7102	0.8183	0.5333	0.2881	0.6136	0.6441	0.6591

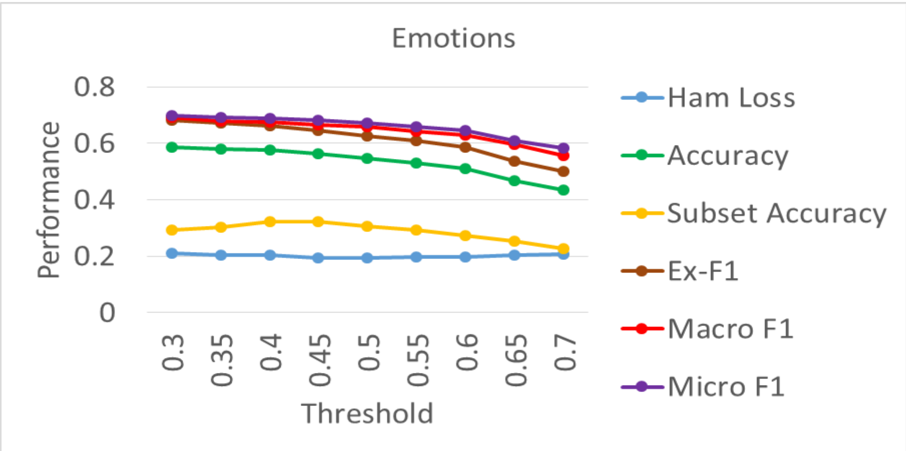


FIGURE 6.28: Effect of threshold variation on MLFLD using Emotions dataset

6.11.2 Effect of threshold variation on Scene dataset

For the Scene dataset, both 0.45 and 0.5 thresholds (Th) have shown minimum hamming loss. But Th 0.3 and 0.35 have shown better accuracy and Ex-F1 at the cost of increased hamming loss. 0.45 is the best choice for threshold in the case of the Scene dataset. A 0.5 value is used throughout experimentation. Its performance is very similar to that at 0.45 as well as better than average across all thresholds.

TABLE 6.241: Effect of threshold variation on MLFLD using Scene dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.3	0.0893	0.0682	0.2050	0.4258	0.8785	0.7402	0.6338	**0.7762**	0.7755	0.7642
0.35	0.0849	0.0682	0.2050	0.4258	0.8785	**0.7407**	0.6529	0.7702	**0.7794**	**0.7686**
0.4	0.0816	0.0682	0.2050	0.4258	0.8785	0.7326	0.6704	0.7534	0.7758	0.7673
0.45	**0.0797**	0.0682	0.2050	0.4258	0.8785	0.7243	**0.6750**	0.7408	0.7732	0.7662
0.5	**0.0797**	0.0682	0.2050	0.4258	0.8785	0.7083	0.6629	0.7235	0.7683	0.7617
0.55	0.0804	0.0682	0.2050	0.4258	0.8785	0.6872	0.6479	0.7004	0.7576	0.7526
0.6	0.0808	0.0682	0.2050	0.4258	0.8785	0.6633	0.6304	0.6744	0.7456	0.7428
0.65	0.0804	0.0682	0.2050	0.4258	0.8785	0.6522	0.6217	0.6624	0.7422	0.7397
0.7	0.0826	0.0682	0.2050	0.4258	0.8785	0.6297	0.6025	0.6388	0.7283	0.7269
Avg	0.0822	0.0682	0.2050	0.4258	0.8785	0.6976	0.6442	0.7156	0.7607	0.7544

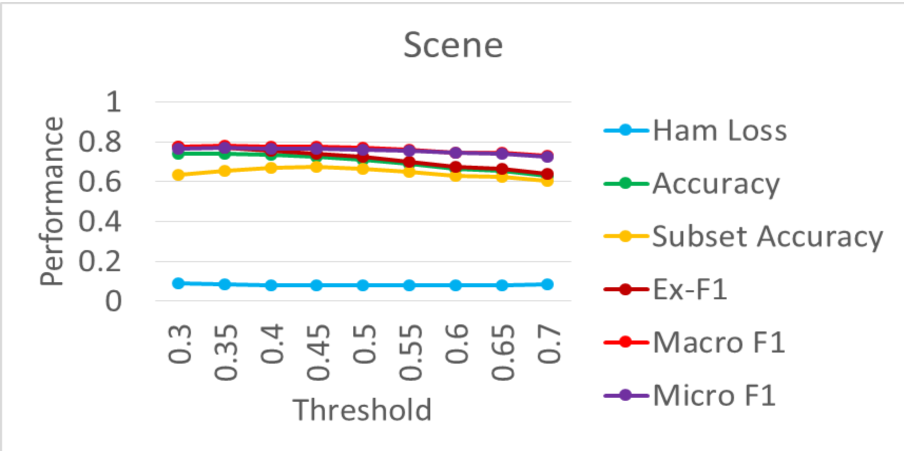


FIGURE 6.29: Effect of threshold variation on MLFLD using Scene dataset

6.11.3 Effect of threshold variation on an Image dataset

For Image dataset, threshold (Th) 0.45 and 0.55 have shown better subset accuracy and least hamming loss, respectively. Th 0.3 has shown better accuracy and all F1 measures. Performance at Th 0.5 is seen well than average overall thresholds.

Table 6.242: Effect of threshold variation on MLFLD using Image dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.3	0.1859	0.1570	0.2916	0.8964	0.8105	**0.6008**	0.4142	**0.6664**	**0.6565**	**0.6531**
0.35	0.1804	0.1570	0.2916	0.8964	0.8105	0.6007	0.4337	0.6595	0.6554	0.6518
0.4	0.1670	0.1570	0.2916	0.8964	0.8105	0.5959	0.4697	0.6398	0.6527	0.6498
0.45	0.1625	0.1570	0.2916	0.8964	0.8105	0.5869	**0.4747**	0.6255	0.6468	0.6442
0.5	0.1631	0.1570	0.2916	0.8964	0.8105	0.5588	0.4632	0.5916	0.6287	0.6259
0.55	**0.1624**	0.1570	0.2916	0.8964	0.8105	0.5294	0.4447	0.5582	0.6089	0.6071
0.6	0.1639	0.1570	0.2916	0.8964	0.8105	0.5043	0.4267	0.5307	0.5909	0.5909
0.65	0.1652	0.1570	0.2916	0.8964	0.8105	0.4672	0.3997	0.4900	0.5656	0.5671
0.7	0.1698	0.1570	0.2916	0.8964	0.8105	0.4172	0.3577	0.4373	0.5296	0.5313
Avg	0.1689	0.1570	0.2916	0.8964	0.8105	0.5401	0.4316	0.5777	0.6150	0.6135

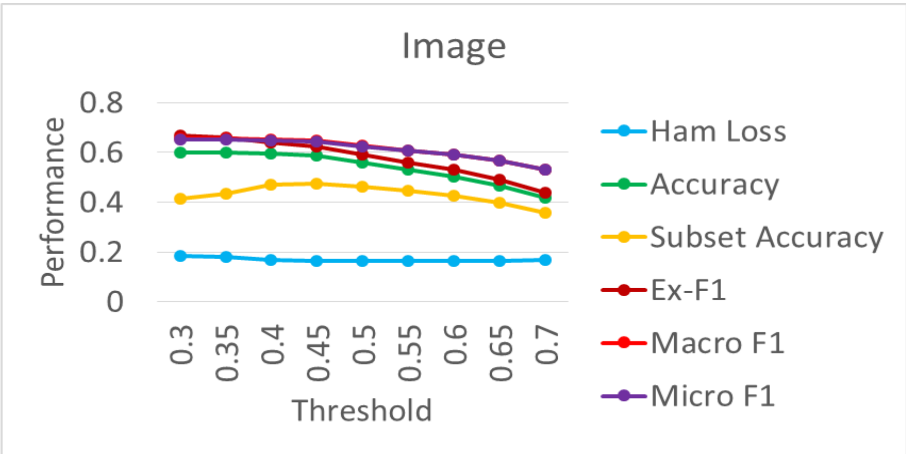


Figure 6.30: Effect of threshold variation on MLFLD using Image dataset

6.11.4 Effect of threshold variation on Yeast dataset

For the Yeast dataset, a minimum hamming loss is obtained for threshold 0.5; however, better accuracy and ex-F1 is noticed for thresholds 0.3 to 0.4. MLFLD is not able to compute Macro-F1 for Yeast shown by NaN.

TABLE 6.243: Effect of threshold variation on MLFLD using Yeast dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.3	0.2225	0.1689	0.2378	6.2905	0.7648	0.5490	0.1813	**0.6578**	NaN	0.6679
0.35	0.2115	0.1689	0.2378	6.2905	0.7648	**0.5513**	0.1979	0.6566	NaN	**0.6698**
0.4	0.2038	0.1689	0.2378	6.2905	0.7648	0.5484	**0.2116**	0.6505	NaN	0.6680
0.45	0.1996	0.1689	0.2378	6.2905	0.7648	0.5341	0.2112	0.6352	NaN	0.6587
0.5	**0.1981**	0.1689	0.2378	6.2905	0.7648	0.5116	0.2046	0.6109	NaN	0.6426
0.55	0.2017	0.1689	0.2378	6.2905	0.7648	0.4797	0.1942	0.5753	NaN	0.6162
0.6	0.2018	0.1689	0.2378	6.2905	0.7648	0.4614	0.1855	0.5543	NaN	0.6033
0.65	0.2064	0.1689	0.2378	6.2905	0.7648	0.4258	0.1627	0.5162	NaN	0.5728
0.7	0.2133	0.1689	0.2378	6.2905	0.7648	0.3830	0.1407	0.4681	NaN	0.5357
Avg	0.2065	0.1689	0.2378	6.2905	0.7648	0.4938	0.1877	0.5917	NaN	0.6261

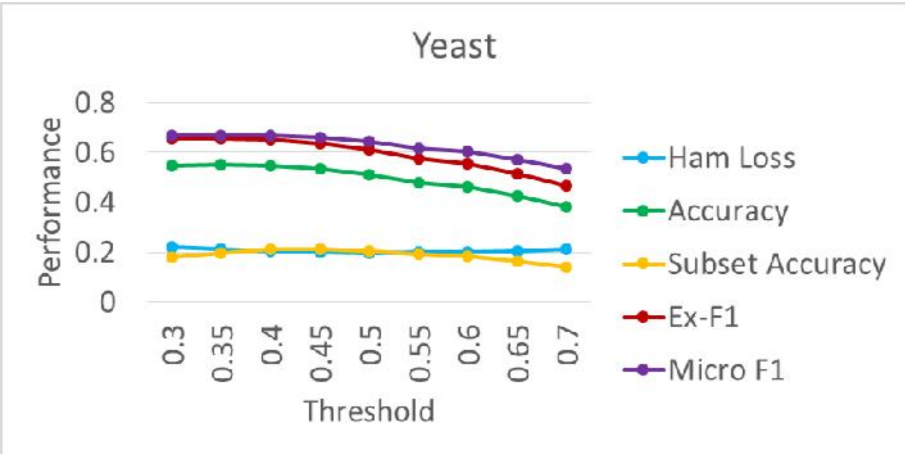


FIGURE 6.31: Effect of threshold variation on MLFLD using Yeast dataset

The threshold is varied from 0.3 to 0.7, and the performance of each dataset is seen. It is noticed that if you try to minimize hamming loss, then you have to compromise on some other performance measures like accuracy, ex-F1, macro, and micro F1. That is, it is not possible to optimize hamming loss, accuracy, and F measure simultaneously. This work has focused on minimizing the hamming loss metric. Also, it can be seen that hamming loss at threshold 0.5 is near to optimum value among all threshold values except for the Image dataset. From these observations and sources from the literature [20] [12] [37] [42] [89], the threshold value used throughout the remaining experimentation is 0.5.

6.12 Effect of Smoothing parameter variation on proposed algorithms

Like k variation, the performance of MLFLD is monitored for smoothing parameter variation. It takes values 0.25, 0.5, 0.75 and 1. Four datasets used for k variation are used for experimentation. Tables 6.244 to 6.247 show the performance of MLFLD for the same. It is noted that measures like hamming loss, accuracy, F measure do not affect. But the ranking loss, one error, coverage, and average precision show the minimal effect of smoothing factor variation.

6.12.1 Effect of smoothing parameter variation using Emotions dataset

For Emotions, rank loss, coverage, and avg precision, have a favorable impact of the smoothing factor increase. But an adverse effect is seen on one error, as shown in Table 6.244.

TABLE 6.244: Effect of Smoothing parameter variation on MLFLD using Emotions dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.25	0.1938	0.1499	0.2475	1.7220	0.8170	0.5483	0.3051	0.6274	0.6584	0.6727
0.5	0.1938	0.1496	0.2492	1.7169	0.8173	0.5483	0.3051	0.6274	0.6584	0.6727
0.75	0.1938	0.1490	0.2492	1.7136	0.8178	0.5483	0.3051	0.6274	0.6584	0.6727
1.0	0.1938	0.1483	0.2492	1.7102	0.8183	0.5483	0.3051	0.6274	0.6584	0.6727

6.12.2 Effect of smoothing parameter variation using Scene dataset

Only for Scene dataset, all metrics are showing slight adverse effect. Ranking loss, one error and coverage are showing slight increase whereas avg precision is showing a slight decrease with increasing smoothing factor.

Table 6.245: Effect of Smoothing parameter variation on MLFLD using Scene dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.25	0.0797	0.0681	0.2042	0.4250	0.8791	0.7083	0.6629	0.7235	0.7683	0.7617
0.5	0.0797	0.0680	0.2037	0.4242	0.8793	0.7083	0.6629	0.7235	0.7683	0.7617
0.75	0.0797	0.0680	0.2042	0.4242	0.8792	0.7083	0.6629	0.7235	0.7683	0.7617
1.0	0.0797	0.0682	0.2050	0.4258	0.8785	0.7083	0.6629	0.7235	0.7683	0.7617

6.12.3 Effect of smoothing parameter variation using Image dataset

For all values, no change in metric values is observed for Image as shown in Table 6.246.

Table 6.246: Effect of Smoothing parameter variation on MLFLD using Image dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.25	0.1631	0.1570	0.2916	0.8964	0.8105	0.5588	0.4632	0.5916	0.6287	0.6259
0.5	0.1631	0.1570	0.2916	0.8964	0.8105	0.5588	0.4632	0.5916	0.6287	0.6259
0.75	0.1631	0.1570	0.2916	0.8964	0.8105	0.5588	0.4632	0.5916	0.6287	0.6259
1.0	0.1631	0.1570	0.2916	0.8964	0.8105	0.5588	0.4632	0.5916	0.6287	0.6259

6.12.4 Effect of smoothing parameter variation using Yeast dataset

For Yeast, rank loss, one error, coverage and avg precision, have the favorable impact of smoothing factor increase, as shown in Table 6.247.

Table 6.247: Effect of Smoothing parameter variation on MLFLD using Yeast dataset

k	Ham Loss (↓)	Rank Loss (↓)	One Error (↓)	Coverage (↓)	Avg. Prec. (↑)	Accuracy (↑)	Subset Accuracy (↑)	Ex-F1 (↑)	Macro F1 (↑)	Micro F1 (↑)
0.25	0.1981	0.1697	0.2394	6.3025	0.7639	0.5116	0.2046	0.6109	NaN	0.6426
0.5	0.1981	0.1693	0.2390	6.2992	0.7643	0.5116	0.2046	0.6109	NaN	0.6426
0.75	0.1981	0.1690	0.2386	6.2925	0.7647	0.5116	0.2046	0.6109	NaN	0.6426
1.0	0.1981	0.1689	0.2378	6.2905	0.7648	0.5116	0.2046	0.6109	NaN	0.6426

The performance of 4 datasets is shown in Table 6.244 to 6.247. It is noticed that variation in the smoothing factor has less effect on the performance of MLFLD. From these observations and sources from the literature [20] [12] [37] [42] [89], the smoothing factor is set to 1 for the remaining experimentation.

To summarize, the selection of the most appropriate neighbors is crucial for any kNN based algorithm. Computation of feature similarity has been commonly used in existing kNN based approaches, including MLkNN. In the case of multi-label data, an instance is associated with multiple labels. Two multi-label algorithms are proposed in this work. The first algorithm called Multi-Label Classification, using Feature similarity and Label Dissimilarity (MLFLD). The second algorithm proposed in this work is called MLFLD with MAXimum Probability (MLFLD-MAXP). Both algorithms utilize important implicit information embedded in features as well as labels in order to identify the most appropriate neighbors for a given test instance. Evaluation of both the algorithms was carried out using cross-validation as well as train-test sets. For cross-validation, ten folds were used on five benchmark datasets. For train-test splits, thirteen benchmark datasets were used for which splits are available from their sources. The performance is measured for eight example-based metrics, namely hamming loss, ranking loss, one error, coverage, average precision, accuracy, subset accuracy, example-based F1, and two label-based metrics, namely, the macro-F1 and micro-F1. It is observed that not all the parameters can be improved simultaneously. MLFLD and MLFLD-MAXP dominate the state-of-the-art algorithms used for comparison for the subset accuracy and demonstrate their effectiveness. For searching nearest neighbors, both algorithms use features along with labels of instances, as observed in the dataset. It helps to increase the correct prediction of the label set, causing growth in subset accuracy.

Chapter 7

Conclusion and Future Scope

Lots of data in the real world inherently is multi-label data. Thus, multi-label classification has gained significant importance and application in the recent past and thereby attracted researchers too. Existing methods for multi-label classification belong to two approaches: one that reorganizes data called problem transformation approach. It needs higher computation time and also relatively less accurate as it loses important implicit information due to data reorganization. Another time-efficient method uses data directly without any data reorganization. It is referred to as an algorithm adaptation approach. It is found to be superior w. r. t. to classifier performance when compared to the transformation approach.

This work proposes a novel multi-label classification algorithm MLFLD that follows the algorithm-adaptation approach. It considers label dissimilarity along with feature similarity to enhance classifier performance. The work also proposes MLFLD-MAXP, an extension of MLFLD.

Concluding remarks based on the work carried out are provided in this section. The notable research finding is summarized and provides directions for further research.

7.1 Multi-label classification using MLFLD and MLFLD-MAXP algorithms

Many researchers have designed the lazy (kNN) multi-label classification methods that follow the algorithm adaptation approach. Such classifiers identify appropriate neighbors of the given test instance and classify the test instance. MLkNN is one such existing algorithm and appears currently to be the best algorithm that follows the algorithm adaptation approach.

The selection of the most appropriate neighbors is crucial for any kNN based algorithm. Computation of feature similarity has been commonly used in existing kNN based approaches, including MLkNN. In the case of multi-label data, as the instance is associated with multiple labels, a new method may be devised for further performance enhancements by considering label dissimilarity in addition to feature similarity.

Two multi-label algorithms are devised in this work. The first algorithm called Multi-Label Classification, using Feature similarities and Label Dissimilarities (MLFLD). The second algorithm proposed in this work is called MLFLD with MAXimum Probability (MLFLD-MAXP). Both algorithms utilize important implicit information embedded in features as well as labels in order to identify the most appropriate neighbors for a given test instance.

Evaluation of both the algorithms is carried out using cross-validation as well as train-test sets. For cross-validation, ten folds are used on five benchmark datasets. For train-test splits, thirteen benchmark datasets are used for which splits are available from their sources. The performance is measured for eight example-based metrics, namely hamming loss, ranking loss, one error, coverage, average precision, accuracy, subset accuracy, example-based F1, and two label-based metrics, namely, the macro-F1 and micro-F1. It is observed that not all the parameters are used generally in the reported literature. Many of the researchers use only the first five parameters, while others either use only hamming loss, accuracy, ex-F1, or only macro and micro F1. This work has used all the ten parameters for the evaluation of the performance of proposed algorithms. Like other domains, the performance parameters conflict with each other, and thus it is not possible for any algorithm to optimize each of these parameters.

While a lower value is desired for one error, coverage, hamming and ranking loss, a higher value is desired in case of the remaining six metrics. Hence the metrics average rank and number of wins are used for analyzing the performances of algorithms. The algorithm that provides the lowest average rank and the maximum number of wins indicates the best algorithm.

Several experiments are carried out using in all seven existing transformation-based and algorithm adaptation-based algorithms, including the best known ML-kNN algorithm. A summary of important observations is provided in the following sections.

Comparative analysis shows that the performance of both MLFLD and MLFLD-MAXP is identical w. r. t. one error, coverage, average precision, and rank loss.

7.1.1 Evaluation using Cross-Validation

Algorithm MLFLD has outperformed all the seven competing algorithms. It provides the smallest average rank of 1 and 10 on 10 wins. In fact, it has outshined w. r. t. subset accuracy for all datasets with overall 8% improvements as well as 5% improvements for accuracy when compared with MLkNN. Also, an average increase of 5%, 4%, and 3% is observed in the case of ex-F1, micro, and macro F1, respectively. The average improvement in one error and rank loss is 4%, each with a 1% improvement in hamming loss and average precision each. Value for coverage by algorithm MLFLD is observed to be the same as that of MLkNN while it is better than the remaining methods.

MLFLD-MAXP, as like MLFLD, outperforms all other competing algorithms with average rank of 1 and 10 wins out of 10. MLFLD-MAXP shows further improvements in subset accuracy by 15% and 10% for accuracy as compared to MLkNN. Ex-F1, macro, and micro F1 are improved by 9%, 7%, and 4% respectively while one error and rank loss are enhanced by 3%. A gain of 1% and 0.1% is observed in average precision and coverage, respectively. Thus performance improvement shown by MLFLD-MAXP is almost twice that of MLFLD for accuracy, subset accuracy, ex-F1, and 1% higher for label-based measures.

Misclassification, in the case of MLFLD, is 0.5% lower than MLFLD-MAXP, and thus it provides better hamming loss. It is obvious because of intentionally assigning at least one label to each instance whenever no label is predicted in the case of MLFLD-MAXP.

Thus at the cost of a slight degradation in hamming loss, MLFLD-MAXP provides the percentage improvement for subset accuracy, accuracy, and ex-F1.

When both the algorithms are compared with MLkNN, MLFLD-MAXP outperforms ML-kNN as well as MLFLD with the smallest avg. rank of 1.1 and 9 wins. MLFLD provides an average rank of 1.5 and 5 wins while MLkNN provides a rank of 3.1 and no wins.

7.1.2 Evaluation using Train-Test

Experimentation using thirteen benchmark datasets show that the algorithm MLFLD-MAXP provides superior performance with the smallest average rank of 1.8 over 10 measures. MLkNN provides an average rank of 3.1 that is much higher than that of MLFLD-MAXP. The average rank of MLFLD is twice that of MLFLD-MAXP.

Algorithm MLFLD-MAXP provides the rank of 1 for the subset accuracy with 11% and 35% improvements over algorithms CC and MLkNN, respectively. It outperforms algorithm ML-kNN w.r.t. accuracy as well as algorithm CC w.r.t. ex-F1 by 30% and 1-3%. Both MLFLD and MLFLD-MAXP algorithms have lesser misclassifications than others except MLkNN and defeat all other algorithms except MLkNN w.r.t. one error, ranking loss, average precision and coverage. It should be noted that all nearest neighbor-based algorithms, namely, BRkNN, MLkNN, and MLFLD, do not perform well on accuracy and F measure based metrics in these experiments.

7.2 Effect of Distance Metrics

7.2.1 Effect of distance metrics on the computation of feature similarity

Though the proposed algorithms perform well using Euclidean distance, it is interesting to see the effect of distance metrics on multi-label classification. Three distance metrics, namely Euclidean, Manhattan, and Minkowski, are used for computing feature similarity and to evaluate the algorithms. Some of the observations are:

- MLFLD-MAXP with particular distance measure is better than MLFLD for all performance parameters except hamming loss.

- In the case of experiments using cross-validation with five benchmark datasets, MLFLD-MAXP with Manhattan outperforms all other algorithms at the cost of computation time that is almost three times higher than that of MLkNN.

- For train-test experiments with thirteen benchmark datasets, it was noted that MLFLD-MAXP using Euclidean defeats MLkNN for average rank. Performance improvement for subset accuracy is 30%, while for accuracy and ex-F1, it is 30% and 7-10% for label-based measures.

7.2.2 Effect of distance metrics for large datasets

When two large datasets are used for experiments with distance metrics, the time required with Manhattan is double while it is more than double for Euclidean and Minkowski experiments compared to that of MLkNN, respectively. The use of the Manhattan distance measure has enhanced the performance of MLFLD more than MLFLD-MAXP. Both have exceeded MLkNN.

7.2.3 Effect of distance metrics on the computation of label dissimilarity

Throughout the experimentations, the main focus is to examine how the use of label dissimilarity measure affects the performance of MLFLD and MLFLD-MAXP. Initially, only Hamming distance is used for label dissimilarity, and three feature similarity measures are tested. Later Jaccard and SimIC distance measures are also used for label dissimilarity. Overall 18 variants obtained from of 2 proposed algorithms, 3 measures for feature similarity and label dissimilarity each, are compared with MLkNN. It is observed that

- Pattern noted for one error, coverage, average precision and rank loss is the same for Hamming, Jaccard and SimIC. The performance of MLFLD variants seems the same as that of corresponding MLFLD-MAXP variants.

- *With cross-validation* on five datasets, MLFLD-MAXP, Jaccard, Manhattan triplet tops among 19 experiments. All variants of proposed algorithms defeat MLkNN in average rank. For MLFLD-MAXP, experiments with Hamming and Jaccard distance measures seem to behave similarly, and both are viewed to be better than SimIC variants.

- *In train-test experiments* with thirteen datasets, MLFLD-MAXP, Hamming, Euclidean triplet tops among 19 experiments. MLFLD-MAXP and Hamming distance with Euclidean, Manhattan, Minkowski distances exceed MLkNN in average rank. The remaining variants could not defeat MLkNN.

7.3 Effect of Outliers

As outliers affect the predictive performance of a classifier, experimentation is performed on datasets with and without outlier removal, and performance is analyzed for cross-validation as well as train-test splits. Some observations are:

- When the performance of proposed algorithms without outlier removal is compared with only MLkNN, the proposed algorithms behave identically w.r.t. hamming and ranking losses, coverage, one error, and average precision, whereas MLFLD-MAXP provides better improvements compared to MLFLD for the remaining five metrics.

- The performance of the proposed algorithms after removing outliers from datasets is found to be better than all competing algorithms. MLFLD-MAXP provides better improvements compared to MLFLD. Although MLFLD is always better for a hamming loss when compared with MLFLD-MAXP, the performance of MLFLD-MAXP is found to be better after outlier removal.

- *For cross-validation experiments* using five datasets, both proposed algorithms have shown the same performance for one error, ranking loss, coverage, and average precision with 37, 33, 10, and 2 percent improvement over MLkNN, respectively. Maximum growth is seen for subset accuracy, which is 46% and 35%, whereas 32% and 24% for accuracy with MLFLD-MAXP and MLFLD, respectively. The execution time of all experiments is comparable.

- *For train-test experiments*, MLFLD has improved hamming loss by 18% than MLFLD-MAXP by 14% compared to MLkNN. Proposed algorithms perform equally well for ranking loss, one error, coverage, and average precision with 20, 16, 8, and 7 % improvement than MLkNN, respectively. More improvement is seen in subset accuracy and example-based accuracy by MLFLD-MAXP as 81% and 73% than 34% and 29% improvement of MLFLD, respectively. MLFLD-MAXP has outperformed with all datasets for ex-F1 and 11 datasets for micro-F1 by 70% and 47% respectively for the same. MLFLD results in 27% and 23% growth respectively. The time required by the proposed algorithms is almost twice than of MLkNN due to label dissimilarity computation.

7.4 Effect of Data Preprocessing

The use of feature and instance selection is a common practice in the case of single-label classifiers and often provides higher classification performance. The effect of using such pre-processing techniques on the proposed multi-label classifier algorithm is summarized below.

7.4.1 Effect of feature selection

When proposed algorithms are evaluated with and without feature selection, MLFLD-MAXP shows a slight improvement. Feature selection has little effect on the overall performance of both the algorithms. MLFLD-MAXP provides improvements in seven metrics. Enhancement in subset accuracy, coverage and macro-F1 is only 0.61%, 0.18% and 0.09% respectively.

Proposed algorithms when compared with other existing algorithms, it is observed that MLFLD-MAXP stands first with an average rank of 1.3 and 8 wins, whereas MLFLD stands second with an average rank of 1.6 and 4 wins. MLFLD-MAXP performs slightly better than MLFLD for the two accuracy measures and the three F-measures. Both the algorithms outperform MLkNN and other contestant algorithms. The performance of the proposed algorithms is identical w. r. t. one error, coverage, ranking loss, and average precision. The performance of MLFLD and MLkNN is identical w.r.t. average hamming loss.

7.4.2 Effect of instance selection

The performance of MLFLD, MLFLD-MAXP, and MLkNN is compared using sampling with replacement with sample sizes of 60 to 100 percent. Both MLFLD and MLFLD-MAXP exhibit identical performance. Moreover, these algorithms provide better results for sample sizes of 80%, and 90% and a size of 90% offer superior performance.

Steady improvement is observed in the performances of the proposed algorithms when the sample size is varied between 60% and 90%. The sample size of 60% is not helpful for performance enhancements; however, it is still better than that of contesting algorithms. MLFLD-MAXP with a sample size of 90% outshines with the smallest average rank of 1.8, and 7 wins. It is followed by MLFLD with an average rank of 2.2 and 6 wins. Both algorithms defeat MLkNN w.r.t. all performance parameters, except for the coverage in case of MLFLD.

Better progress in the performance of MLFLD-MAXP w.r.t. accuracy, subset accuracy, and ex-F1 is observed compared to MLFLD after instance selection. Performance growth for both algorithms w.r.t. one error, coverage, average precision, and ranking loss is identical.

7.4.3 Effect of feature and instance selection

The use of the feature and instance selection is found to be very useful in upgrading the performance of proposed algorithms as compared to using the only feature or instance selection. When the performance of MLFLD and MLFLD-MAXP is examined using both feature and instance selection, MLFLD-MAXP outperforms MLFLD.

Performance comparison of proposed algorithms with MLkNN using both feature and instance selection shows that MLFLD-MAXP provides the best performance with an average rank of 1.1 and 9 wins, whereas MLFLD stands at second position with the average rank of 1.5 and 5 wins and MLkNN stands at third position. Significant gain in performance is noticed w.r.t. subset accuracy and accuracy with feature and instance selection.

7.5 Effect of Model Input Parameters

Classifier models (either eager or lazy) are built using specific input parameters. The values used for building the model decide classifier performance. In the case of lazy learners like the kNN classifiers, parameters such as k many times, determine the performance in the case of single-label classifiers. The following sections throw light on the effect of such input parameters on multi-label classification, and are briefly outlined below.

7.5.1 Effect of k variation

The number of neighbors, k, is a crucial parameter for the k nearest neighbors (kNN) classifiers. But in the case of multi-label classifiers based on kNN, the scenario is different. While doing the experimentation of MLFLD, k is varied from 5 to 15. It is noted that "k" has little effect on the performance of MLFLD. From these observations and sources from the literature, the value of 10 for k is used in the experimentation.

7.5.2 Effect of threshold variation

The threshold is varied from 0.3 to 0.7, and the performance of the proposed algorithm MLFLD is monitored. It is noticed that if one attempts to minimize hamming loss, then one has to compromise some other performance measures such as accuracy, ex-F1, macro, and micro F1. Thus, it is not possible to optimize hamming loss, accuracy, and F measure simultaneously. Also, it is observed that hamming loss at the threshold value of 0.5 is near to its optimum value in the case of most of the datasets. From these observations, as well as from sources in the literature, the threshold value of 0.5 is used throughout the remaining experimentation.

7.5.3 Effect of the smoothing parameter

Like variation in parameter k, the performance of MLFLD is monitored by varying the value of the smoothing parameter between 0.25 and 1 with a step of 0.25. It is noticed that variation in the smoothing factor has little effect on the performance of MLFLD.

7.6 Concluding Remarks

The previous section provides observations about the performance of the proposed algorithms. To summarize,

- For this work, the hypothesis is that the use of label dissimilarity along with feature similarity can enhance the performance of a lazy learner such as the nearest neighbor (kNN) based multi-label classifier. The work proposes two novel multi-label classification algorithms called MLFLD and MLFLD-MAXP that incorporate the idea of using label dissimilarity as well as feature similarity for deciding nearest neighbors.

- Empirical evaluation using benchmark datasets from various domains confirms the hypothesis. It shows that both MLFLD and MLFLD-MAXP outperform all existing approaches, including the best known MLkNN in terms of rank and number of wins.

- It also presents a trade-off between the performance and computation time to make appropriate choice of suitable distance metrics.

- Though the time taken is more for MLFLD and MLFLD-MAXP, enhancement in the accuracy is notable, which is essential in some applications like medical.

- MLFLD and MLFLD-MAXP dominate the state-of-the-art algorithms used for comparison for the subset accuracy and demonstrate their effectiveness. For searching nearest neighbors, both algorithms use features along with labels of instances, as observed in the dataset. It helps to increase the correct prediction of the label set, causing growth in subset accuracy.

- The relative performance of MLFLD-MAXP is better than MLFLD for all measures except hamming loss.

- For cross-validation, the higher performance of proposed algorithms is strongly notable, especially for subset accuracy because of its potential to find out complete label sets. Both MLFLD and MLFLD-MAXP provide maximum wins.

- The grouping of instance selection and feature selection helps in further boosting the performance of proposed algorithms.

- Both MLFLD and MLFLD-MAXP thus are superior to MLkNN, and can be better choices for multi-label classification.

7.7 Future Scope

The work may be extended further to perform the following tasks.

1. Use of the partial label set: Proposed algorithms make use of the whole label set to compute the dissimilarity of labels. Instead of using the entire label set, it can be tested whether the use of partial label set to measure label dissimilarities affects their performance. It may be achieved using label correlations.

2. Handling of datasets containing nominal attributes: All the datasets used in this work consists of numeric attributes only. There exist a few multi-label datasets that include nominal attributes or a mix of nominal and numeric attributes. Further investigations are needed to decide suitable modifications in proposed algorithms if any.

3. Dealing with Multi-class (class labels having more than two values) Multi-label classification: In the majority of datasets, class labels in datasets are binary. It may be possible for class labels to have more than two values.

4. Processing datasets with class labels having a hierarchical relationship: All the datasets used in this work possess class labels which are at the same level. There also exist datasets that consist of labels arranged in the hierarchy. A label is described further by sub-labels forming label hierarchy. A count of siblings and the depth of a label in the hierarchy may be considered for classification.

5. Use of divide and conquer strategy or parallel processing to speed up the algorithm: Proposed algorithms are designed as sequential algorithms. Both work well for smaller datasets while take considerable time for large datasets. To reduce time, divide and conquer strategy can be used, or parallel processing can be applied to handle large/big datasets in a reasonable amount of time.

References

[1] Dai Y, Li Y, Li S-T (2020) "Multi-label learning for concept-oriented labels of product image data", Image and Vision Computing 93, 103821

[2] Rafal Rak et al, "Multi-label Associative Classification of Medical Documents from MEDLINE", Proc. of the Fourth International Conf. on Machine Learning and Applications (ICMLA'05), 2005 IEEE

[3] Yannis Papanikolaou et al, "Large-Scale Online Semantic Indexing of Biomedical Articles via an Ensemble of Multi-Label Classification Models", Journal of Biomedical Semantics 2017 8:43

[4] Qinghua Yu et al, "Combining local and global hypotheses in deep neural network for multilabel image classification", Neurocomputing 235 (2017) 38–45

[5] Kong, X. et al, "Multi-label feature selection for graph classification". In Data Mining (ICDM), 2010 IEEE 10th Int. Conf. (pp. 274-283)

[6] Ricardo Cerri et al, "Comparing methods for multi-label classification of proteins using machine learning techniques", Springer 2009

[7] M. L. Zhang, Z. H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization", IEEE Transactions on Knowledge and Data Engineering 18(10)(2006)1338–1351

[8] Trohidis, K. et al, 2008, September. "Multi-Label Classification of Music into Emotions". In ISMIR (Vol. 8, pp. 325-330)

[9] Z. H. Zhou, M. L. Zhang, “Multi-Instance Multi-Label Learning with Application to Scene Classification”, Advances in Neural Information Processing Systems, 2006, pp. 1609-1616

[10] Xin Chen et al, “Mining Social Media Data for Understanding Students Learning Experiences”, IEEE Transactions On Learning Technologies, Vol. 7, No. 3, July-September 2014

[11] Eneldo Loza Mencia, “Multilabel Classification in Parallel Tasks”, 2nd Int. Workshop on Learning from Multi-Label Data, Israel, 2010

[12] M. L. Zhang, Z.H. Zhou, “ML-kNN: A lazy learning approach to multilabel learning”, Pattern Recognit., vol.40, no.7, pp.2038–2048, 2007

[13] Alberto F. De Souza et al, “Automated multi-label text categorization with VG-RAM weightless neural networks, Neurocomputing 72 (2009) 2209–2217

[14] Zhiqiang Zeng et al, “Multimedia annotation via semi-supervised shared-subspace feature Selection”, Journal of Visual Communication and Image Representation, Volume 48, October 2017, Pages 386-395

[15] Tsoumakas G, Katakis I (2007) “Multi-label classification: An overview, International Journal of Data Warehousing and Mining”, vol. 3, no. 3, pp. 1-13

[16] Tsoumakas G, Zhang ML, Zhou ZH (2009) “Tutorial on learning from multi-label data”, in ECML PKDD, Bled, Slovenia [Online]. Available: http://www.ecmlpkdd2009.net/wpcontent/uploads/2009/08/learning-from-multi-label-data.pdf

[17] Carvalho A de, Freitas AA (2009) “A tutorial on multi-label classification techniques”, in Studies in Computational Intelligence 205, A. Abraham, A. E. Hassanien, and V. Snásel, Eds. Berlin, Germany: Springer, pp. 177–195

[18] Tsoumakas G et al (2010) "Mining multilabel data, Data Mining and Knowledge Discovery Handbook", O. Maimon and L. Rokach, Eds. Berlin, Germany: Springer, pp. 667-686

[19] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning", Pattern Recognit., vol. 45, no. 9, pp. 3084–3104, 2012

[20] Zhang, Min-Ling and Zhou, Zhi-Hua. (2014). "A Review on Multi-Label Learning Algorithms". Knowledge and Data Engineering, IEEE Transactions on. 26. 1819-1837. 10.1109/TKDE.2013.39

[21] Zhou, Zhi-Hua, and Zhang, Min-Ling. "Multi-instance multi-label learning with application to scene classification" Advances in neural information processing systems. 2006.

[22] Tsoumakas G., Katakis I. and Vlahavas I. (2011). "Random k-Labelsets for Multi-Label Classification". IEEE Transactions on Knowledge Data Engineering, 23. 1079-1089. 10.1109/TKDE.2010.164

[23] Read, J., Pfahringer, B., Holmes, G., Frank, E.: "Classifier chains for multi-label classification". In: Proc. of European Conf. on Machine Learning and Knowledge Discovery in Databases: Part II. ECML PKDD '09, Berlin, Heidelberg, Springer-Verlag (2009) 254-269

[24] G. Tsoumakas and I. Katakis, "Effective and efficient multilabel classification in domains with large number of labels", in Proc. Work. Notes ECML PKDD Workshop MMD, Antwerp, Belgium, 2008

[25] Schapire R.E., Singer Y., "Boostexter: a boosting-based system for text categorization", Machine Learning 39 (2000) 135-168

[26] A. Clare, R.D. King, "Knowledge discovery in multi-label phenotype data", in: Proc. of 5th European Conf. on PKDD, 2001, pp. 42–53

[27] Ross Quinlan (1993). “C4.5: Programs for Machine Learning”. Morgan Kaufmann Publishers, San Mateo, CA

[28] M. L. Zhang et al, “Feature selection for multi-label naive Bayes classification”, Information Sciences 179 (2009) 3218–3229

[29] J. Read, “A pruned problem transformation method for multi-label classification”, in: Proceedings of the New Zealand Computer Science Research Student Conference, 2008, pp. 143–150

[30] J. Read et al, “Multi-label classification using ensembles of pruned sets”, Proc. of 8th IEEE Int. Conf. on Data Mining, 2008, pp.995–1000

[31] S. Godbole, S. Sarawagi, “Discriminative methods for multi-labeled classification”, in: Advances in Knowledge Discovery and Data Mining, Springer, Berlin/ Heidelberg, 2004, pp. 22–30.

[32] J. Arunadevi et al, “An evolutionary multi-label classification using associative rule mining for spatial preferences”, IJCA Special Issue on “Artificial Intelligence Techniques - Novel Approaches and Practical Applications” AIT, 2011

[33] Ravi Patel, Jay Vala, Kanu Patel, “Classification on multi-label dataset using rule mining technique”, IJRET: International Journal of Research in Engineering and Technology Vol. 03 Issue: 06, Jun-2014

[34] Raed Alazaidah et al, “A multi-label classification approach based on correlations among labels”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, 2015

[35] H Haripriya et al, “Multi-label prediction using association rule generation and simple k-means”, 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), IEEE

[36] F. Sebastiani, “Machine learning in automated text categorization”, ACM Compu. Surv. 34 (1) (2002) 1-47

[37] E. Spyromitros-Xioufis, G. Tsoumakas, and I. Vlahavas, “An empirical study of lazy multilabel classification algorithms”, in Proc. 5th Hellenic Conf. Artif. Intell., Syros, Greece, 2008, pp. 401–406

[38] Celine Vens, Jan Struyf, Leander Schietgat, Saˇso Dˇzeroski, and Hendrik Blockeel (2008), “Decision trees for hierarchical multi-label classification”. Machine Learning, 2(73):185–214

[39] K. Dembczy´nski et al (2010), “Bayes optimal multilabel classification via probabilistic classifier chains”, in Proc. 27th Int. Conf. Mach. Learn., Haifa, Israel, pp. 279–286

[40] Liang Sun et al (2008), “Hypergraph spectral learning for multi-label classification”, KDD’08, August 24–27, ACM

[41] Hung-Yi Lo et al (2014), “Generalized k-Labelsets Ensemble for Multi-Label and Cost-Sensitive Classification”, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 7, pp. 1679-1691

[42] Jung-Yi Jiang, Shian-Chi Tsai, Shie-Jue Lee (2012), “FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors”, Expert Systems Applications, vol.39(3), pp. 2813-2821

[43] S. Lee, J. Jiang (2014), “Multilabel text categorization based on fuzzy relevance clustering”, IEEE T. Fuzzy Systems, vol.22, no.6, pp.1457-1471

[44] Rubiya P U et al, “A fuzzy based approach for multilabel text categorization and similar document retrieval”, IJARCSSE, Volume 5, Issue 9, September 2015 ISSN: 2277 128X

[45] Gao, Sheng et al, "A MFoM learning approach to robust multiclass multi-label text categorization". In Proc. of the 21st international conf. on Machine learning, p. 42. ACM, 2004

[46] Zhang, Yin, and Zhi-Hua Zhou. "Multilabel dimensionality reduction via dependence maximization". ACM Transactions on Knowledge Discovery from Data (TKDD) 4, no. 3 (2010): 14

[47] Ji, S., Tang et al, 2008, August. "Extracting shared subspace for multi-label classification". In Proc. of 14th ACM SIGKDD international conf. on Knowledge discovery and data mining, pp. 381-389

[48] J. Han, M. Kamber (2012), "Data Mining: Concepts and Techniques", The Morgan Kaufmann Series in Data Management Systems

[49] Doquire G., Verleysen M. (2011) "Feature Selection for Multi-label Classification Problems". In: Advances in Computational Intelligence. IWANN 2011. Lecture Notes in Computer Science, vol 6691. Springer, Berlin, Heidelberg

[50] Li, S., Zhang, Z., Duan, J. (2014). "An ensemble multi-label feature selection algorithm based on information entropy". Int. Arab J. Inf. Technol., 11(4), 379-386pp

[51] Li, L. et al, December. "Multi-label feature selection via information gain". In International Conference on Advanced Data Mining and Applications (pp. 345-355), Springer International Publishing, 2014

[52] Jungjit et al, "A new genetic algorithm for multi-label correlation-based feature selection", ESANN 2015 proc., Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015

[53] Zhang, M.L. and Wu, L., 2015. "LIFT: Multi-label learning with label specific features". IEEE transactions on pattern analysis and machine intelligence, 37(1), pp.107-120

[54] Kenji Kira and Larry A. Rendell (1992), "A practical approach to feature selection", Machine Learning Proceedings, Pages 249–256

[55] Newton Spolaˆor et al (2013) "Relief for multi-label feature selection", Brazilian Conference on Intelligent Systems IEEE

[56] Newton Spolaˆor et al, "A comparison of multi-label feature selection methods using the problem transformation approach", Electronic Notes in Theoretical Computer Science 292 (2013) 135–151

[57] Zhang, M.L. and Zhang, K., 2010, July. "Multi-label learning by exploiting label dependency". In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 999-1008

[58] S.-J. Huang, Y. Yu, and Z.-H. Zhou, "Multi-label hypothesis reuse", in Proc. 18th ACM SIGKDD Conf. KDD, Beijing, China, 2012, pp. 525–533

[59] Huang, S.J. and Zhou, Z.H., 2012, July. "Multi-label learning by exploiting label correlations locally". In Twenty-Sixth AAAI Conference on Artificial Intelligence

[60] Ying Yu, Witold Pedrycz, Duoqian Miao (2014) "Multi-label classification by exploiting label correlations", Expert Systems with Applications 41 (2014) 2989–3004

[61] Gauthier Doquire, Michel Verleysen, "Mutual information based feature selection for multilabel classification", Neurocomputing 122(2013)148–155

[62] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review", ACM Comput. Surv., vol. 31, no. 3, pp. 264–323, 1999

[63] Nasierding, Gulisong, Tsoumakas, Grigorios and Kouzani, Abbas Z. 2009, "Clustering based multi-label classification for image annotation and retrieval", in SMC 2009: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, IEEE, Piscataway, N. J., pp. 4514-4519

[64] Pranav Gupta, Ashish Anand, "Multilabel classification using label clustering", Appearing in Proceedings of the 1st Indian Workshop on Machine Learning, IIT Kanpur, India, 2013

[65] Zhilou Yu et al, "An improved classifier chain algorithm for multilabel classification of big data analysis", HPCC, 2015 IEEE

[66] Lu Sun and Mineichi Kudo and Keigo Kimura, "A scalable clustering-based local multilabel classification method", ECAI 2016

[67] Rosane M. M. et al, "Multilabel OCS with genetic algorithm for rule discovery", GECCO '09 Proc. of 11th Annual conference on Genetic and evolutionary computation, pp. 1323-1330, 2009

[68] E. C. Goncalves, A. Plastino and A. A. Freitas, "A Genetic Algorithm for Optimizing the Label Ordering in Multi-label Classifier Chains," 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, 2013, pp. 469-476, doi: 10.1109/ICTAI.2013.76.

[69] Jungjit S. et al, "Two extensions to multilabel correlation-based feature selection: a case study in bioinformatics", Systems, Man, and Cybernetics (SMC), 2013 IEEE

[70] Quinlan, J. R. (1996). "Bagging, boosting, and C4.5". In Proceedings of Thirteenth National Conf. on Artificial Intelligence, pp.725–730

[71] Tsoumakas, Grigorios, and Ioannis Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification", Machine learning: ECML 2007. Springer Berlin Heidelberg, 2007, 406-417

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

[73] Read, Jesse, and Peter Reutemann (2012) "MEKA: a multi-label extension to WEKA". URL http://meka.sourceforge.net

[74] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, (2011) "MULAN: A Java library for multi-label learning", J. Machine Learning Res., vol. 12, pp. 2411-2414

[75] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, Ian H. Witten (2009) "The WEKA data mining software: An update", SIGKDD Explore, vol. 11, no. 1, pp. 10(18)

[76] C.-C. Chang and C.-J. Lin (2011) "LIBSVM: A library for support vector machines", ACM Trans. Intelligent Systems Technol., vol. 2, no. 3, Article 27. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[77] Gjorgji Madjarov, Ivica Dimitrovski, Dejan Gjorgjevikj, and Saˇso Dˇzeroski, "Evaluation of Different Data-Derived Label Hierarchies in Multi-label Classification", Springer International Publishing Switzerland 2015, A. Appice et al. (Eds.): NFMCP 2014, LNAI 8983, pp. 19–37, 2015, DOI: 10.1007/978-3-319-17876-9 2

[78] Johannes F¨urnkranz, Eyke H¨ullermeier, Eneldo Loza Menc´ıa, Klaus Brinker (2008) "Multilabel classification via calibrated label ranking". Machine learning, 73(2), pp.133-153

[79] R. Kohavi (1995), "A study of cross-validation and bootstrap for accuracy estimation and model selection", IJCAI 1995

[80] Piotr Szyma´nski, Tomasz Kajdanowicz (2017), "scikit-multilearn: A scikit-based Python environment for performing multi-label classification", Feb 2017

[81] L. A. Zadeh (1965), "Fuzzy Sets", Inf. Control, vol. 8, pp. 338-353

[82] Z. Younes, F. Abdallah, and T. Denoeux, "Fuzzy multi-label learning under veristic variables", in IEEE International Conference on Fuzzy Systems (IEEE-FUZZ 2010). IEEE, 2010, pp. 1–8

[83] J. Chen, Y. Y. Tang, C. Chen, B. Fang, Y. Lin, and Z. Shang, “Multi-label learning with fuzzy hypergraph regularization for protein subcellular location prediction”, IEEE Trans. on NanoBioscience, vol. 13, no. 4, pp. 438–447, 2014

[84] R. C. Prati, “Fuzzy rule classifiers for multi-label classification”, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2015, pp1–8

[85] J. C. H ¨uhn and E. H¨ullermeier, “FURIA: an algorithm for unordered fuzzy rule induction”, Data Min. Knowl. Discov., vol. 19, no. 3, pp. 293–319, 2009

[86] Reem Al-Otaibi, Peter Flach, and Meelis Kull, “Multi-label Classification: A Comparative Study on Threshold Selection Methods”, ECML PKDD 2014

[87] Yang, Y., “A study of thresholding strategies for text categorization”. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR ’01, New York, NY, USA, ACM (2001) 137–145

[88] Rong-En Fan and Chih-Jen Lin. 2007. “A study on threshold selection for multilabel classification”. Department of Computer Science, National Taiwan University (2007), 1-23

[89] Min-Ling Zhang and Zhi-Hua Zhou, “A k-nearest neighbor based algorithm for multi-label classification,” 2005 IEEE International Conference on Granular Computing, Beijing, 2005, pp. 718-721 Vol. 2, doi: 10.1109/GRC.2005.1547385

[90] Zoulficar Younes, Fahed Abdallah and Thierry Denoeux, “Fuzzy Multi-Label Learning Under Veristic Variables”, WCCI 2010 IEEE World Congress on Computational Intelligence, July, 18-23, 2010 - CCIB, Barcelona, Spain, pp. 1696-1703

[91] Chunming Liu and Longbing Cao, “A Coupled k-Nearest Neighbor Algorithm for Multi-label Classification”, Springer International Publishing Switzerland 2015, PAKDD 2015, Part I, LNAI 9077, pp. 176–187, 2015. DOI: 10.1007/978-3-319-18038-0 14

[92] Darko Aleksovski, Dragi Kocev, and Saˇso Dˇzeroski, "Evaluation of distance measures for hierarchical multi-label classification in functional genomics", ECML PKDD 2009

[93] Catia Pesquita*, Daniel Faria, Hugo Bastos, André O. Falcão and Francisco M. Couto, "Evaluating GO-based Semantic Similarity Measures", The 10th Annual Bio-Ontologies Meeting, ISMB/ECCB 2007

[94] Veloso, A., Meira, Jr., W., Gon¸calves, M., and Zaki, M. (2007). "Multi-label lazy associative classification". In PKDD '07: 11th European conference on Principles and Practice of Knowledge Discovery in Databases, pages 605–612, Berlin, Heidelberg. Springer.

[95] Cerri R et al (2019) "Inducing Hierarchical Multi-label Classification rules with Genetic Algorithms", Applied Soft Computing Journal 77, pp. 584–604

[96] Len Feremans, Boris Cule, Celine Vens, Bart Goethals (2020) "Combining instance and feature neighbours for extreme multi-label classification"

[97] Jesse Read, "Scalable Multi-label Classification", The University of Waikato, September 2010 (Thesis)

[98] A. Elisseeff, J. Weston, "A kernel method for multi-labelled classification", in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, 2002, pp. 681–687

[99] Zhu, S., Ji, X., Xu, W., and Gong, Y. (2005). "Multi-labelled classification using maximum entropy method". In SIGIR: '05: 27th Annual ACM Conference on Research and Development in Information Retrieval, pages 274–281

[100] Godbole, S. and Sarawagi, S. (2004). "Discriminative methods for multilabeled classification". In PAKDD '04: Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 22–30. Springer

[101] Kiritchenko, S. (2005). "Hierarchical Text Categorization and its Application to Bioinformatics". PhD thesis, Queen's University, Kingston, Canada

[102] Ghamrawi, N. and McCallum, A. (2005). "Collective multi-label classification". In CIKM '05: 14th ACM international Conference on Information and Knowledge Management, pages 195–200, New York, NY, USA. ACM Press

[103] Z. Younes, F. Abdallah, T. Denoeux, H. Snoussi, "A dependent multi-label classification method derived from the k-nearest neighbor rule", Hindawi Publishing Corporation, EURASIP Journal on Advances in Signal Processing, Volume 2011, Article ID 645964, 14 pages, doi:10.1155/2011/645964

Publications

Copyright

Vaishali S. Tidake, Dr. Shirish S. Sane, "A Novel Algorithm for Multi-label Classification"

- Dated: 09/01/2018
- Registration Number: L-72017/2018
- Diary Number: 16885/2017-CO/L
- Copyright Reg. of: Literary/Dramatic work

Journal Papers

1. V. S. Tidake, S. S. Sane, "Multi-label learning with MEKA", CSI Communication 2016 August issue, pp 33-37
2. V. S. Tidake, S. S. Sane, "Multi-label Classification: A Survey", International Journal of Engineering and Technology, 7 (4.19) (2018) pp 1045-1054
3. Vaishali S. Tidake, Shirish S. Sane, "A Novel Algorithm for Multi-label Classification by exploring feature and label dissimilarities", International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 11(2019) pp 161-169

Conference Papers with Publications

- V. S. Tidake, S. S. Sane, "Evaluation of Multi-label Classifiers in Various Domains using Decision Tree", Intelligent Computing and Information Communication, Springer 2018, pp 117-127 **(Won best paper award)**

- Vaishali S. Tidake, Shirish S. Sane, “Effect of Distance Metrics on Multi-label Classification”, First Doctoral Symposium on Natural Computing Research 2020 **(Won best paper award in the track, Received Letter of award for Best Paper by Springer Nature)**

Paper Presentations in Conference

Based on the work done so far, presentations are done in the following conferences:

- Paper entitled “Survey of multi-label learning” is presented in cPGCON 2015 held by Savitribai Phule Pune University conducted at MET IOT, Nashik on 13-14 Mar 2015.
- Paper entitled “Survey of fuzzy multi-label learning” is presented in cPGCON 2016 held by Savitribai Phule Pune University conducted at PCCOE, Pune on 25-26 Mar 2016.
- Paper entitled “Multi-label learning: A Comparative Study” presented in cPGCON 2017 held by Savitribai Phule Pune University conducted at SITRC, Nashik on 24-25 Mar 2017.