Development of Multi-Label Classification Algorithm Based on Correlations among Labels

By

Raed Hasan Saleh Diab

Supervisor

Dr. Fadi Fayez Thabtah

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Deanship of Academic Research and Graduate Studies

Philadelphia University

January 2013

جامعة فيلادلفيا
نموذج تفويض

أنا رائد حسن صالح ذياب ، أفوض جامعة فيلادلفيا بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأشخاص عند طلبها.

التوقيع :

التاريخ :

Philadelphia University
Authorization Form

I am, Raed Hasan Saleh Diab, authorize Philadelphia University to supply copies of my thesis to libraries or establishments or individuals upon request.

Signature:

Date:

Development of Multi-Label Classification Algorithm Based on Correlations among Labels

By

Raed Hasan Saleh Diab

Supervisor

Dr. Fadi Fayez Thabtah

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Deanship of Academic Research and Graduate Studies

Philadelphia University

January 2013

Successfully defended and approved on _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Examination Committee Signature		**Signature**
Dr. Academic Rank:	, Chairman.	_ _ _ _ _ _ _ _ _ _ _ _
Dr. Academic Rank:	, Member.	_ _ _ _ _ _ _ _ _ _ _ _
Dr. Academic Rank:	, Member.	_ _ _ _ _ _ _ _ _ _ _ _
Dr. Academic Rank:	, External Member.	_ _ _ _ _ _ _ _ _ _ _ _

( )

Dedication

I fully dedicate this thesis:

To Nae'la Al-Salman

For being the most important part of the dream

For the support, courage, and unconditional friendship

Raed Diab

Acknowledgment

Thanks TO ALLAH first, before and after everything for giving me the knowledge and ability to complete this work in this final form.

I would like to thank our university for offering scientific nutrition necessary to complete our study and our research. I would like to express my sincere thanks to the staff of the college who provide a warm and lively environment to encourage and help graduate students in their graduate study. Especially, Prof. Saeed AL-Goul, Dr. Nameer Al-Emam, Dr. Moayad Al-Athami, Dr. Samer Hanna and Dr. Omar Heiasat for their support of the educational process.

I would like to extend my regards and sincere gratitude to Dr. Fadi Fayez Thabta, who has guided me through my work from the beginning and supported me and to all of those who have helped and encouraged me.

I would like to thank my family, for their unconditional love and support, and never forget to thank my friends in the pioneer center for gifted students.

Raed Diab

Table of Contents

List of Figures

List of Tables

Abstract

Multi label classification is concerned with learning from set of instances that are associated with a set of labels, that is, an instance could be associated with multiple labels at the same time. This task occurs frequently in application areas like text categorization, multimedia classification, bioinformatics, protein function classification and semantic scene classification.

Current multi-label classification methods could be divided into two parts. The first part is called problem transformation methods, which transform multi-label classification problem into single label classification problem, and then apply any single label classifier to solve the problem. The second part is called algorithm adaptation methods, which adapt an existing single label classification algorithm to handle multi-label data.

The following are some of the research challenges in the field of multi-label classification problem:

1. Design a hierarchical structure for multi- label to manage label correlationships.
2. To extract relevant label sets from multi-label data set.
3. A novel approach that uses both problem transformation methods, and algorithm adaptation methods, to improve performance and accuracy for multi-label classifier.

In this thesis, we propose a multi-label classification algorithm based on correlations among labels, that uses both problem transformation methods and algorithm adaptation method. The algorithm begins with transforming multi-label dataset into single label dataset using least frequent label criteria, and then applies PART algorithm on the transformed dataset. Also the algorithm tries to get benefit from positive correlations among labels using predictive Apriori algorithm. The output of the algorithm is multi-labels rules. The algorithm has been evaluated using two multi-label datasets ("Emotions"," Yeast") and three evaluation measures (Accuracy, Hamming Loss, Harmonic Mean). Further, we show by experiments that this algorithm has a fair accuracy comparing with other related algorithms.

Chapter 1

Introduction

1.1 Motivation

Classification is one of the data mining tasks, which aims to predict the class label of unseen instances as accurate as possible (Thabtah et al., 2004). Classification usually involves separating data into training and testing sets. Each instance in the training test contains one class label and several attributes. Common applications for classifications are credit card scoring and insurance fraud detection.

When talking about classification, we need to distinguish between two types of classification, the first type is called traditional label classification or single label classification which is based on assumption that labels are mutually exclusive, that is, there are no relationships between labels, and labels are independent by them selves. The second type of classification is called multi-label classification, which assumes that labels are not mutually exclusive and therefore are not independent, that is, there are some relationships between labels (Sorower ,2010).

Traditional single label classification is concerned with learning from set of instances that are associated with disjoint labels .If the number of disjoint labels equals to "2", a classification task is called binary classification, and if the number of disjoint labels greater than 2, a classification task is called multi-class classification.

On the other hand, multi label classification is concerned with learning from set of instances that are associated with a set of labels, that is, an instance could be associated with multiple labels at the same time. This task occurs frequently in application areas like text categorization, multimedia classification, bioinformatics, protein function classification and semantic scene classification.

Consider a task of classifying E-mails , any incoming mails could be spam or not spam but not both at the same time , so we have to choose between two disjoint labels (spam , not spam) ,this type of classification is a single label classification (Binary classification) .Now, suppose that we have the famous movie "Omar Mokhtar " ,and we need to classify this movie , in this case we could associate this movie to three labels at the same time "Drama" ,"Documentary " and "Action" .This kind of classification is called multi-label classification.

Consider for example a medical diagnosis problem classification, where we have symptoms such as fever, blocked sinus, and coughing. In such case, symptoms could be associated with multiple labels at the same time such as "cold", "flu" and "fever ". A problem like this is a good example for multi-label classification where an instance could be associated with multiple labels at the same time.

Current multi-label classification methods could be divided into two parts. The first part is called problem transformation methods, which transform multi-label classification problem into single label classification problem, and then apply any single label classifier to solve the problem. The second part is called algorithm adaptation methods, which adapt an existing single label classification algorithm to handle multi-label data (Tsoumakas et al., 2007).

1.2 Research Problem

Based on the literature review of multi-label classification, we can assure that there is no guided multi-label classification algorithm that seeks the important correlations between labels. No guided algorithm that tries to capture the important correlations between labels in order to reduce problem search space could be found in multi-label classification literature. Therefore we are trying to design a guided multi-label classification algorithm based on correlations among labels in class label attribute.

1.3 Research Questions

Many questions need to be studied and investigated in depth about correlations among labels such as:

- How cardinality and diversity distinguish multi-label data set from each other? And what is the relationship between cardinality and the accuracy of the classifier?
- How much labels are correlated with each others?
- What is the average number of association rules between labels?

Since data sets differ from each other in many ways such as cardinality, diversity, number of distinct label sets and average number of association rules between labels, we think there is a great need to answer the above questions, which might be very helpful in determining how to classify the instances and what is the best method to use for specific domain.

1.4 Data Mining

Multi-label classification is a type of classification which in turns is a branch of a larger area of scientific study known as Data Mining (DM). (Sorower ,2010) defined data mining as one of the main phases in Knowledge Discovery from Databases (KDD), which extracts useful patterns from data. In this section, we give a brief introduction to the area of data mining, and show its main tasks and domain applications.

Automated data collection tools, large memory capacities, and the availability of high speed computers, are reasons for making the process of collecting and storing huge quantities of information, possible and some what easy(Thabtah et al., 2004). Governments, companies, and even users store all the information they need in databases. Moreover, people believe that: by storing data in databases, they might save some information that might turn up to be potentially useful in the future, in spite that; it is not of direct value at the moment.

It is only in the late 1980s and early 1990s that the database community has shown its interest in KDD and DM. However, since mid-1990s both fields have gone through a rapid

expansion, due to an extraordinary support and attention of software industry. Even that data mining is the most important phase in KDD, other phases that comprises KDD is also very important, such as data selection, data preprocessing, pattern interpretation and visualization as shown in figure 1.1.

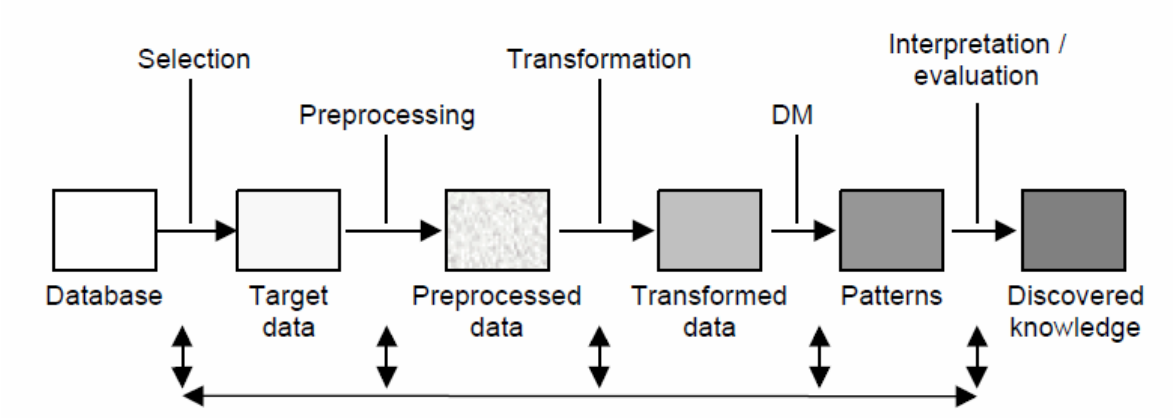

Figure 1.1 KDD process

The first phase in DM is **Selection** that aims to select typical data from the database, in order to make the target data set as representative as possible. The second phase is **preprocessing** that aims to eliminate noise from the target data set and possibly generates specific data sequences in the set of preprocessed data. The next phase is **transformation** of the preprocessed data into a suitable form for performing the desired DM task. The last phase is **interpretation / evaluation** that aims to keep only those patterns that are interesting and useful to the user and discard the rest. Those patterns that remain represent the discovered knowledge. Discovered patterns are usually represented using a certain well-known knowledge representation technique, including inference rules (If-Then rules), decision trees, tables, diagrams, images, analytical expressions, and so on…

Some of the most common data mining tasks include classification, regression, association rule discovery, and clustering. Those tasks could be accomplished using some data mining techniques adopted and borrowed from different scientific field such as artificial intelligence, statistics, and machine learning. An important fact is that: there is no single data mining technique that could be applicable to all tasks (Thabtah et al., 2004). Some of the common data mining tasks include:

1. Classification – Is a task of assigning objects to one of several predefined classes as accurate as possible. Examples include detecting spam E-mail messages based on the header or the contents, classifying books in the library based on title or subject.

2. Clustering – Is an unsupervised learning task that aims to group objects with certain similarities, where the similarity between the resulting clusters are minimized, and similarities between objects inside each cluster are maximized.
3. Regression – Which is a special case of classification where the outcome class is numeric. In regression we consider the class as linear combinations of different attributes, with pre-specified weight obtained from the training data.
4. Association Rule Discovery – Which is a task for discovering important and interesting relationships which are hidden in large data sets, Association Rule Discovery is unsupervised learning task which is typically applied to market basket analysis.

KDD and DM help people improve efficiency of the data analysis they perform. They also make possible for people to become aware of some useful facts and relations that hold among the data they analyze, and that could not be known otherwise, simply because of the overload caused by heaps of data (Sorower, 2010). Once such facts and relations become known, people can greatly improve their business in terms of savings, efficiency, quality, and simplicity.

1.5 Multi-Label Classification

On the contrary of previous traditional classification; multi-label classification does not consider labels (L) to be mutually exclusive and map set of instances with a set of labels Y where $Y \subseteq L$ as in figure 1.2. (Boutell et al., 2003).That is, the goal in multi label classification is to learn from a set of instances where each instance belongs to one or more label in L.

Multi-label classification was mainly motivated by the tasks of text categorization and medical diagnosis. While , nowadays, multi-label classification is increasingly required by modern applications such as music categorization into emotions, semantic video annotation, direct marketing, automated tag suggestion, protein function classification (Diplaris et al., 2005) and semantic scene classification (Boutell et al. , 2004). As an example of multi-label classification, suppose an article concerning Syrian refugees in Jordan. This article could be classified as "political" as well as "Social" .And the famous

movie "Omar Mokhtar" could be classified as "Drama" movie or "Action" as well as a "Documentary" one.

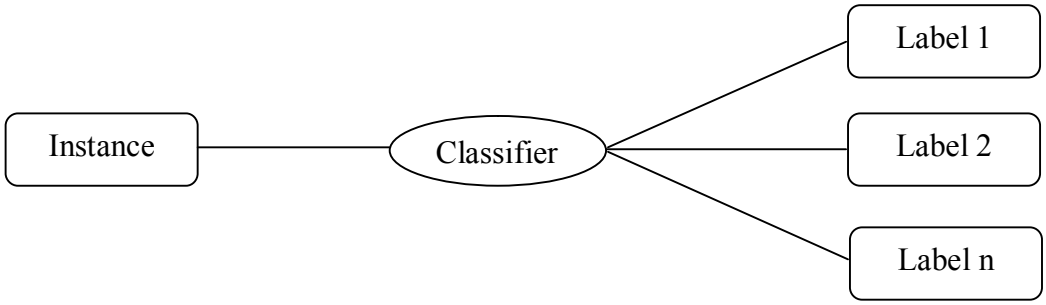

Figure 1.2 Multi-Label Classification

According to (Sorower ,2010) some of the most important challenges and open researches in multi-label classification include exploiting labels correlations and exploring the conditional and unconditional dependencies between labels ,also even it has been approved that label cardinality can strongly affect the performance of multi-label algorithm, there is no systematic study on how and why the performance varies over different data properties, in addition to the need of designing an on-line algorithm that scales with large and sparse domain.

Following are some of the research challenges in the field of multi-label classification problem (Purvi et al., 2012).

1. Design a hierarchical structure for multi- label to manage label correlationships.
2. To extract relevant label sets from multi-label data set.
3. A novel approach that uses both problem transformation methods, and algorithm adaptation methods, to improve performance and accuracy for multi-label classifier.

1.6 Association Rule Discovery

Association rule mining is one of the most important and well researched techniques of data mining, which aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items, in transactional databases or other data repositories. Association rules are widely used in various areas such as telecommunication networks, marketing and risk management, inventory control.

One of the most important applications in association rule discovery is market basket analysis, where huge amount of customers purchase data are collected daily, at the checkout counters of grocery stores. This huge amount of data contains valuable information that could be used in many important decisions such as marketing promotions, inventory management, and customer relationship management.

1.6.1 Basic Definitions

In this section, some basic concepts and definitions will be explained using the following table (Pang et. al ,2005) which represents an example of market basket transactions.

Table 1.1 Transactional Data Set

TID	Itemset
1	{ Bread , Milk }
2	{ Bread , Diapers , Beer , Eggs }
3	{ Milk , Diapers , Beer , Cola }
4	{ Breads , Milk , Diapers , Beer }
5	{ Breads , Milk , Diapers , Cola }

Itemset: let I = { i_1 , i_2 ,… , i_d } be a set of all items in a market basket data and T={t_1 , t_2 , … , t_N} be the set of all transactions. Each transaction t_i contains a subset of items chosen from I . A collection of zero or more items is termed an itemset. If an itemset contains k items, it is called a k-itemset. For example, {Bread, Diapers, Beer} is an example of 3-itemset.

Support count: refers to the number of transactions that contain a particular itemset. For example the support count for {Breads, Milk, Diapers} is equal to two because there are only two transactions that contain all three items.

Association Rule: is an implication expression of the form X→Y, where X and Y are disjoint itemsets.

Support: A measure for evaluating the strength of rule which determines how often a rule is applicable to a given dataset.

Confidence: A measure for evaluating the strength of rule that determines how frequently items in Y appear in transactions that contain X. The formal definitions of these metrics are:

$$\text{Support, } s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N}; \quad (1.1)$$

$$\text{Confidence, } c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}. \quad (1.2)$$

Association Rule Discovery: Given a set of transaction T, find all the rules having support >= minsupp and confidence >= minconf, where minsupp and minconf are the corresponding support and confidence thresholds.

A trivial way to discover all association rule is to compute the support and confidence for every possible rule. A trivial way since it is a very expensive way because there are too many rules that could be discovered. In fact, the total number of rules that could be discovered from any dataset contains d items is:

$$R = 3^d - 2^{d+1} + 1. \quad (1.3)$$

Moreover, most of these rules are discarded and of no significant use, therefore, most of computations become wasted (Agrawal et al., 1993).

A common strategy to discover association rules is to divide the problem into two sub problems as follows:

1- **Frequents Itemset Generation**: which aims to find all the itemsets that satisfy the minsupp threshold. Theses itemsets are called frequent itemsets

2- **Rule Generation**: which aims to find all the high confidence rules from the frequent itemsets generated in the previous step. These rules are called strong rules.

Efficient techniques for generating frequent itemsets and association rules are discussed in the following two sub sections. The computational requirements for frequent itemset generation are generally more expensive than those of rule generation.

1.6.2 Frequent Itemset Generation using Apriori algorithm

Many algorithms and strategies are used to generate frequent itemset but, we will introduce in this section the most important algorithm which is called Apriori algorithm.

Theorem (Apriori principle): if an itemset is frequent, then all of its subsets must also be frequent (Agrawal et al., 1993).

To exemplify the idea behind the Apriori principle suppose, { A,B,C } is frequent itemset, then all of the following subsets are frequent :{A},{B},{C},{AB},{AC},{BC}. Conversely if an itemset like {D, Y} is infrequent itemset, then all of its superset must be infrequent too.

Figure 1.3 provides a high level illustration for frequent itemset generation using Apriori algorithm which is shown in figure 1.4, where minimum support count equals to 3 (minsupp =0.60).

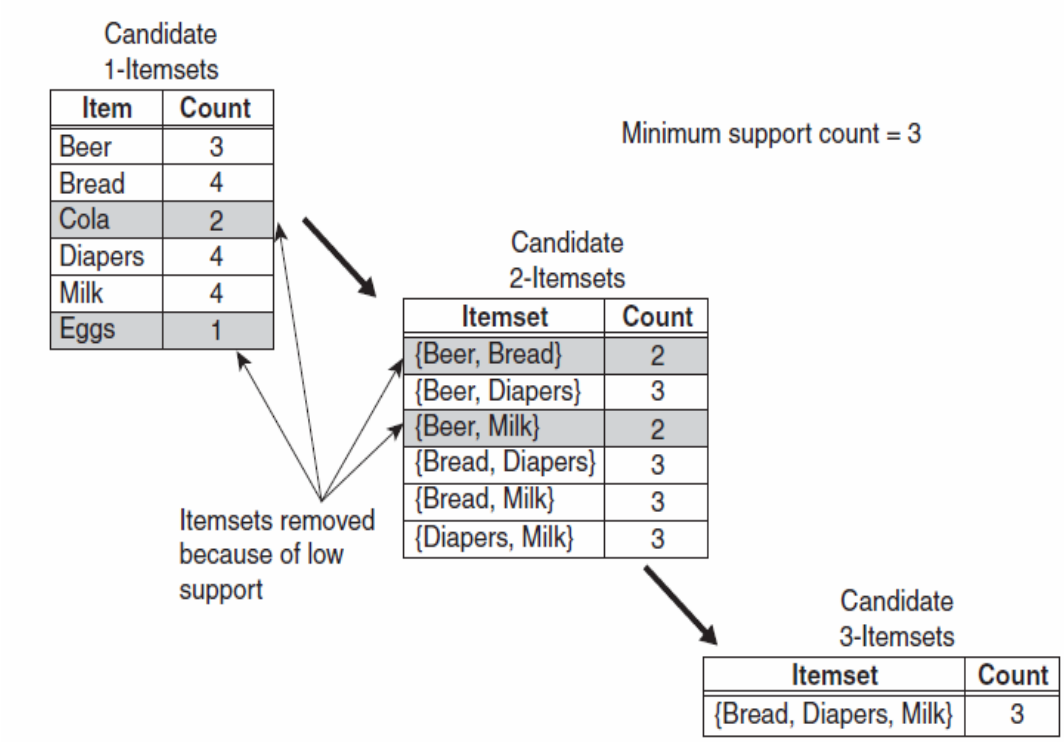

Figure 1.3 Applying Apriori algorithm to the dataset in table 1.1

1: $k = 1$.
2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$. {Find all frequent 1-itemsets}
3: **repeat**
4: $k = k + 1$.
5: $C_k = \text{apriori-gen}(F_{k-1})$. {Generate candidate itemsets}
6: **for each** transaction $t \in T$ **do**
7: $C_t = \text{subset}(C_k, t)$. {Identify all candidates that belong to t}
8: **for each** candidate itemset $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$. {Increment support count}
10: **end for**
11: **end for**
12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup\}$. {Extract the frequent k-itemsets}
13: **until** $F_k = \emptyset$
14: Result $= \bigcup F_k$.

Figure 1.4 Apriori algorithm (Agrawal et al., 1993)

1.6.3 Rule Generation

In this section we will describe how to extract association rules from the frequent itemsets generated in the previous step using Apriori algorithm that is shown in figure 1.5. Each frequent k-itemset, can produce up to $2^k - 2$ association rules , ignoring rules that have empty antecedents or consequents.

Example:

Let Y = { A , B , C } be a frequent itemset of the length 3 , there could be 3! Which is equal to 6 of association rules that might be extracted from Y as following:

{A} ⟶ {B,C} , {B} ⟶ {A,C} , {C} ⟶ {A,B} , {A,B} ⟶ {C} , {A,C} ⟶ {B} , {B, C} ⟶ {A}

As each of their support is identical to the support for Y , the rules must satisfy the support threshold.

1: **for each** frequent k-itemset f_k, $k \geq 2$ **do**
2: $H_1 = \{i \mid i \in f_k\}$ {1-item consequents of the rule.}
3: **call** `ap-genrules`$(f_k, H_1.)$
4: **end for**

Figure 1.5 Rule generation of the Apriori algorithm (Agrawal et al., 1993)

1.7 Thesis Contributions

In this thesis, we aim to meet the following contributions:

- An extensive study about multi-label classification, and the methods that are used to handle the problem of multi-label classification of both groups: problem transformation methods, and algorithm adaptation methods. In addition to study the evaluation measures which are used in the domain of multi-label classification.
- Development of multi-label classification algorithm based on correlations among labels.
- An evaluation process for the proposed algorithm, using some of the evaluation measures that are used in multi-label classification
- Compare the proposed model with other methods, of both groups: problem transformation methods and algorithm adaptation methods.

1.8 Multi-label Data Set Statistics

In some applications, examples are associated with small number of labels for each example. While in other applications, examples are associated with large number of labels for each example.

Definition1: Label cardinality of dataset is the average number of labels for each example in the data set (Boutell et al., 2003)

.

$$\text{Label-Cardinality} = \frac{1}{m}\sum_{i=1}^{m}|Y_i| \tag{1.4}$$

Where, m: is the number of instances in the data set. $|Y_i|$: Number of labels per instance.

Definition2: Label Density of dataset is the average number of labels for examples in the data set divided by the total number of labels (q) (Boutell et al., 2003).

$$\text{Label-Density} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i|}{q} \tag{1.5}$$

Two data sets with the same label cardinality but with a great difference in the number of labels (different label density) might not exhibit the same properties and cause different behavior to the multi-label learning methods (Tsoumakas et al., 2007). The number of distinct label sets is also important for many algorithm transformation methods that operate on subsets of labels.

1.9 Thesis Outline

The thesis consists of 5 chapters. Chapter 2 reviews general multi-label classification methods, for both problem transformation methods and algorithm adaptation methods. Also, chapter 2 focuses on important evaluation measures that are used to evaluate multi-label classifier.

Chapter 3 presents our "Multi – label classification method based on correlations among labels". Chapter 4 gives detailed information about data and experiments. Chapter 5 summarizes the main achievements of this thesis, presents the general conclusions and suggests further research directions.

1.10 Summary

In this chapter, we have introduced a brief introduction on classification, and types of classification. We are interested in multi-label classification, where an instance could be associated with more than one label at the same time. Examples of some modern domains that used multi-label classification include: music categorization into emotions, semantic video annotation, direct marketing, automated tag suggestion, protein function classification and semantic scene classification. Association rule discovery using Apriori algorithm was discussed too, in addition to the most important statistics of multi-label data set.

Chapter 2: Overview of multi-label classification methods

2.1 Introduction

Generally, classification problems can be divided into three main categories; these are Binary classification, Multi-Class classification and Multi-Label classification. In binary classification, a class has only two possible values: as shown in Table 2.1, where only two class labels exist in the training data (X, Y). The letter "A" in columns (1-4) in Table 2.1 corresponds to "attribute" and the last column represents the class attribute. Most real world application domains however, contain several classes and therefore multi-class approach has been proposed. Assume we added two new data objects into Table 2.1 that are associated with new class (Z), i.e. rows (5, 6) in Table 2.2, the data becomes multi-class.

Table 2.1 Binary Data

A1	A2	A3	A4	Class
5	A	2	R	X
3	B	0	A	Y
3	B	2	A	Y
5	B	0	D	X

Table 2.2 Multi-Class Data

A1	A2	A3	A4	Class
5	A	2	R	X
3	B	0	A	Y
3	B	2	A	Y
5	B	0	D	X
3	B	4	T	Z
3	B	6	T	Z

Multi-label classification data, on the other hand, allows training data objects to be associated with multiple labels as shown in Table 2.3. This may result in learning rules that predict more than just single label, whereas most of the current classification approaches do not consider the generation of rules with multiple labels from multi-class or multiple label data (Thabtah et al., 2004).

Table 2.3 Multi-Label Data

A1	A2	A3	A4	Class
5	A	2	R	X,Y
3	B	0	A	X,W,Z
3	B	2	A	Z
3	B	6	T	Y,Z

2.2 Multi-label classification problem Definition

A traditional classification problem can be defined as follows: "let D denotes the domain of possible training instances, and Y be a list of class labels, let H: D → Y denotes the set of classifiers. Each instance d ∈ D is assigned a single class label y that belongs to Y. The goal is to find a classifier h ∈ H that maximize the probability that h(d) = y, for each test case (d , y).In multi- label problem , however , each instance d ∈ D can be assigned multiple labels $y_1,y_2,\ldots,y_k$ for $y_i \subseteq$ Y , and is represented as a pair (d , $(y_1,y_2,\ldots,y_k)$) where $(y_1,y_2,\ldots,y_k)$ is a list of ranked class labels from Y associated with the instance d in the training data". (Thabtah et al., 2004)

2.3 Multi-label classification methods

Existing methods for handling multi-label classification can be grouped into two main groups. The first group, which is an algorithm independent, is called problem transformation methods, while the second group is an algorithm dependent, and is called algorithm adaptation method. The first group transforms multi-label classification problem into one or more single classification problem, while the second group extends a specific learning algorithm, in order to handle multi-label data directly (Boutell et al., 2004).

2.3.1 Problem Transformation Methods

Several problem transformation methods exist in the literature that are used to convert multi-label classification problem into one or more single label classification problem. To exemplify these methods, we will use the dataset of table 2.4 which consists of four examples that belong to the following class set { Reading , Swimming , Painting ,TV Watching }

Table 2.4 Multi-label data set

Instance	Reading	Swimming	Painting	TV Watching
1		X	X	
2	X		X	
3		X		X
4		X		

The first problem transformation method discards every multi-label instance from the data set (Tsoumakas et al., 2007). Therefore, in the previous example, instances 1, 2, 3 will be discarded. Another problem transformation method selects one of the multiple-labels of each multi-label instance either randomly or subjectively. So the previous example instances may be transformed randomly into the following:

Instance	Reading	Swimming	Painting	TV Watching
1		X		
2			X	
3				X
4	X			

2.3.1.1 Copy transformation method

The copy transformation method transforms every multi-label instance to single label instance by replacing multi-label instance (xi, yi) with $|y_i|$ instances. Several transformation methods could be then chosen such as copy-weight which associates a weight of $(1 / |y_i|)$ to each of the transformed examples, select-max (most frequent), select-min (least frequent), and select-random. Finally we could use the ignore transformation methods that discards all multi-label instances (Tsoumakas et al., 2007).

Instance	Label
1a	Swimming
1b	Painting
2a	Reading
2b	Painting
3a	Swimming
3b	TV Watching
4a	Swimming

Copy transformation method

Instance	Label	weight
1a	Swimming	0.5
1b	Painting	0.5
2a	Reading	0.5
2b	Painting	0.5
3a	Swimming	0.5
3b	TV Watching	0.5
4a	Swimming	1

Copy - Weight transformation method

Instance	Label
1	Swimming
2	Painting
3	Swimming
4	Swimming

Select-Max (most frequent)

Instance	Label
1	Painting
2	Reading
3	TV Watching
4	Swimming

Select-Min (Least frequent)

Instance	Label
1	Painting
2	Reading
3	TV Watching
4	Swimming

Select-Random

Instance	Label
4	Reading

Ignore multi-label examples

2.3.1.2 Binary Relevance (BR)

One of the most popular transformation methods, that learn single binary classifier for every label in the label set, is called Binary Relevance (BR). It transforms the original data set into |L| data sets, which contain all the instances from the original data set. It gives a positive sign for a label, if it exists in the data set, and negative sign otherwise. For classification of new instance, BR outputs the union of all the labels that are predicted by the |L| classifiers (Boutell et al., 2004).

Instance	Label
1	Swimming
2	- Swimming
3	Swimming
4	Swimming

Instance	Label
1	Painting
2	Painting
3	- Painting
4	- Painting

Instance	Label
1	- Reading
2	Reading
3	- Reading
4	- Reading

Instance	Label
1	-TV Watching
2	-TV Watching
3	TV Watching
4	-TV Watching

Although Binary Relevance is a simple transformation method, it is based on implicit assumption of label independence which might be completely incorrect in the data.

2.3.1.3 Label Powerset (LP) is a straight forward method that works as follows: it considers each unique set of labels that exists in the data set as a new single label in single – label classification task as shown down:

Instance	Label
1	Swimming, Painting
2	Reading, Painting
3	Swimming , TV Watching
4	Swimming

For predicting of new instance, LP outputs the most probable class which actually could be a set of labels in the original data set. Computational complexity of LP is upper- bounded by (min (|L|, 2^k)) where k: is the total number of classes in the data set before transmission , and usually it is much less than 2^k .LP has an advantage of taking labels correlations into account, on the contrary of BR, but it has a disadvantage when a large number of classes in the original data set associated with small number of instances, which may cause an imbalance problem for learning (Tsoumakas et al., 2007).

The previous mentioned problem of LP was addressed by the pruned problem transformation methods (Read, 2008) which used a user – defined threshold to prune some label sets that occur less than this threshold .The pruned set could be replaced by disjoint subsets of these labels that are more frequent in the data set.

2.3.1.4 RAKEl (Random K label sets)

RAKEl is an effective transformation method proposed by Grigorios Tsoumakas that breaks the initial set of labels into a number of small random subsets called labelsets and then employs LP to train a corresponding classifier, where k is a parameter that determines

the size of the subsets (Tsoumakas et al., 2007).RAKEL offers advantages over LP for the following reasons:

a- The resulting single label classification tasks are computationally simpler

b- Resulting single label classification tasks are characterized by much more balance distribution of class values.

In RAKEL, parameter K which is used to determine the size of the subsets and specified by the user, should be small to avoid the problems of LP.

2.3.1.5 Ranking by Pair wise Comparison (RPC)

RPC transforms multi-label classification problem into single label classification problem through performing pair wise comparisons of labels (Furnkranz et. al.,2003) , It learns (|L| * (|L| - 1)) / 2 binary classifiers, one model for each different pair of labels. For predicting new instance, all models are invoked and ranking is obtained through counting the votes received by each label. An extension of RPC called Calibrated Label Ranking (CLR) introduces a virtual label (often called calibration label, L_0) that aims to separate relevant labels from irrelevant ones (Johannes et. al. 2005).

2.3.1.6 Classifier Chains (CC)

Classifier Chains is a problem transformation method, based on Binary Relevance (BR), which tries to enhance BR through taking label correlations into account. CC builds |L| binary classifier for each label as in BR. Then Classifiers are linked along a chain where each classifier deals with the binary relevance problem associated with label $l_j \in L$. The feature space of each line in the chain is extended with 0/1 label association of all previous links. In short word, by passing label correlation information along a chain of classifiers, CC counteracts the disadvantages of the binary method while maintaining acceptable computational complexity (Read et. al., 2009).

2.3.1.7 Ensemble of Classifier Chains (ECC)

ECC is an enhancement version of CC which in turn is an enhancement of BR. ECC trains m CC Classifiers $C_1,C_2,\ldots,C_m$, Where each C_k is trained with a random chain ordering of L and a random subset of D. Each C_k model is likely to be unique and able to give different multi label predictions. These predictions are then summed by label so that each label receives a number of votes. A threshold is used to select the most popular labels which form the final prediction of multi label set (Read et. al., 2009).

2.3.1.8 Pruned Sets (PS)

This problem transformation method is an enhancement of Label Powerset(LP) which treats every unique subset of labels as a single label, and suffers from label imbalance specially, when number of training examples is small and number of labels is to large. PS try to solve this problem by focusing only on the most important correlations, which reduce complexity and improve accuracy (Read et. al., 2009).

2.3.2 Algorithm Adaptation methods

Algorithm Adaptation methods extend a specific single label learning algorithm in order to handle multi-label data directly. In this section, we introduce a brief plethora of algorithm adaptation methods grouped by the learning concept that they extend.

2.3.2.1 Decision trees methods

The Decision tree (DT) is one of the common learning approaches used in data mining and machine learning. This approach roots back to 1979 when Quinlan proposed his first decision tree version and called it ID3 algorithm, latterly, Quinlan developed an enhanced decision tree learning method known as C 4.5 as an extension of that ID3. Often, the process of constructing tree can be depicted according to (Witten et. al., 2005) as follows: the learning method starts by selecting an attribute as a root node (A) and constructs a single branch for every possible value (A1 and A2). Accordingly, this will divide the data set into two subsets (B and Class1). The same process is repeated recursively for each branch until all data examples in the training data set at the node level have a similar classification.

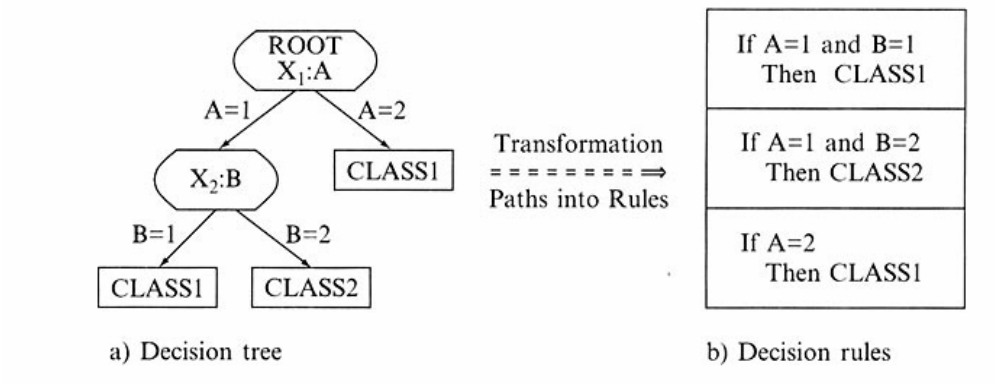

Fig 2.1 Transformation of DT into "IF-Then" Rules

In general, the size of the generated tree is very large even after pruning unnecessary branches, which makes the produced tree complex and hard to understand (Kantardzic, 2003), Normally, each path from the root towards the leaf nodes is transformed into "If-Then" rules, Where the IF part includes all tests documents on a path, and THEN is the final classification for that document as illustrated in Figure 2.1

A decision tree is a hierarchal structure consisting of nodes and directed edges which reflects a series of questions and their possible answers. In decision tree, there are three types of nodes

1- A root node that has no incoming edges and zero or more outgoing edges.
2- Internal nodes, each of which has exactly one incoming edge and two or more outgoing edges.
3- Leaf or terminal nodes, each of which has exactly one incoming edges and no outgoing edges.

In traditional decision tree, each leaf node is assigned just one class label, while the non terminal nodes, which consist of the leaf node, and internal nodes contain attribute test conditions to separate records that have different characteristics.
There are many measures that can be used to determine the best way to split the records. These measures are defined in terms of the class distribution of the records before and after splitting. The measures developed for selecting the best split are often based on the degree of impurity of the child nodes. The smaller the degree of impurity, the more skewed the class distribution.

(Clare and King, 2001) developed a re-sampling technique and modified the C4.5 algorithm to deal with a gene hierarchy multi-label classification problem. Their aim was to generate rules from phenotype experiments data that describe functional classes for a mutated gene, and not prediction. The problem is difficult since a gene exists in a hierarchy and it may belong to more than one functional class. The C4.5 algorithm is only suitable for binary and multi-class classification problems and expects every example to belong to only one class. If the standard C4.5 algorithm is used to produce the rules from the phenotype data, only the largest frequency class for each data object will be considered in the learning phase by C4.5, ignoring very important knowledge. Therefore, a modification to the current implementation of C4.5 was made that allows a leaf to represent a set of class labels. The results indicated that genes normally belong to the top two functional

classes of the hierarchy. C4.5 algorithm was adapted for the handling of multi-label data with the modification of entropy definition as follows:

Entropy = - $\sum$ { P(c_i) log p (c_i) + q (c_i) log q (c_i) }

where $p(c_i)$ = relative frequency of class c_i and $q(c_i) = 1 - p(c_i)$. They also allowed multiple labels in the leaves of the tree.

2.3.2.2 Tree based Boosting

Boosting is a machine learning meta-algorithm for performing supervised learning. When first introduced, the hypothesis boosting problem simply referred to the process of turning a weak learner into a strong learner. "Informally, the hypothesis boosting problem asks whether an efficient learning algorithm that outputs a hypothesis whose performance is only slightly better than random guessing, i.e. a weak learner, implies the existence of an efficient algorithm that outputs hypothesis of arbitrary accuracy, i.e. a strong learner. Algorithms that achieve hypothesis boosting quickly became simply known as "boosting".

AdaBoost is very popular and perhaps the most significant historically as it was the first algorithm that could adapt to the weak learners. AdaBoost.MH and AdaBoost.MR are two extensions of AdaBoost for multi-label data (Tsoumakas et al., 2007), where AdaBoost.MH aims to reduce Hamming loss and AdaBoost.MR aims to increase accuracy.

2.3.2.3 Lazy Learning

Lazy learning is a learning method in which generalization beyond the training data is delayed until a query is made to the system, as opposed to in eager learning, where the system tries to generalize the training data before receiving queries.

The main advantage gained in employing a lazy learning method, is that the target function will be approximated locally, such as in the k-nearest neighbor algorithm. Because the target function is approximated locally for each query to the system, lazy learning systems can simultaneously solve multiple problems and deal successfully with changes in the problem domain.

The disadvantages with lazy learning include the large space requirement to store the entire training dataset. Particularly noisy training data increases the case base unnecessarily, because no abstraction is made during the training phase. Another disadvantage is that lazy learning methods are usually slower to evaluate, though this is coupled with a faster

training phase. Lazy classifiers are most useful for large datasets with few attributes. Several numbers of methods are based on the popular K nearest Neighbors (KNN) lazy learning (Zhang & Zhou, 2007). All of these methods share the same first step with KNN (retrieving the k nearest example) and differ from each others on the aggregation of the label sets of these examples.

2.3.2.4 Associative based Methods

The problem of producing rules with multiple labels was investigated in (Thabtah et al., 2004). Multi-class, Multi-label Associative Classification algorithm (MMAC) was introduced .in addition to four measurements for evaluating the accuracy of classification approaches to a wide range of traditional and multi-label classification problems. MMAC is an associative rule learning based covering algorithm, that recursively learns a new rule and each time removes the examples associated with that rule. Labels for the test instances are ranked according to confidence, support, and rule's cardinality (number of conditions in the left hand side (LHS) of the rule).

2.3.2.5 Neural Network and Support Vector Machines

Support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The neural network (NN) is another classification approach that contains a set of nodes divided into distinctive layers. According to (Feldman and Sanger 2007) NN consists of several layers. The input nodes layer receives the feature values of the documents (X1, X2 etc), followed by zero or more hidden layers, and the output nodes that generate classification status values (Y1, Y2 etc). Where the dependencies among these nodes are called the Link weights as illustrated in figure 2.2

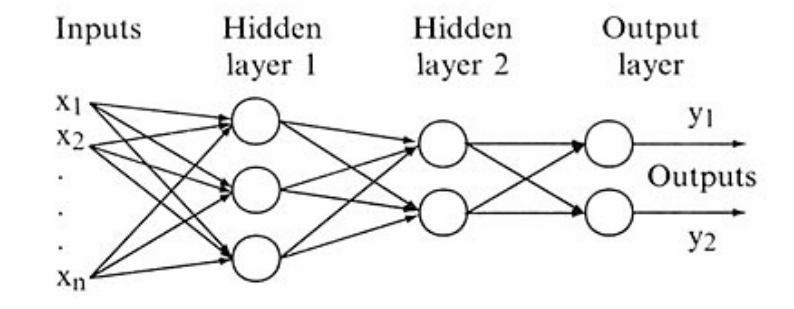

Figure 2.2 Multi-layers Neural Network

One of the known training techniques in NN is the back propagation. According to (Feldman and Sanger, 2007) the working mechanism of NN performs as follows: the training documents will be populated into the input nodes, if the documents are misclassified during this stage. The error is propagated through the network backwards, and this process is repeated along with modifying the link weights in order to reduce the number of errors. The experimental study of (Zaghloul et. al., 2009) revealed that NN is a highly competitive learning approach for text classification in comparison with other learning approaches.

Back-propagation – Multi Label Learning (BP-MLL) is an adaptation of the back propagation algorithm for multi-label learning with the modification of introducing new error function that takes multiple labels into account. Multi-class, Multi-label Perceptron (MMP) is a family of online algorithms for label ranking from multi-label data based on the Perceptron algorithm.

Table 2.5 Comparative study between Problem Transformation methods and algorithm adaptation methods

Problem Transformation Methods	**Algorithm Adaptation methods**
Algorithm independent	Algorithm dependent
Multiple models or single model is used	Single model is used
Data preprocessing is required	Limited preprocessing is required

2.4 Evaluation Measures

Evaluating performance of multi-label classification differs from evaluating the performance of single-label classification. In fact, the evaluating process seems to be more complicated in multi-label classification, since the result of the classifier could be fully correct, fully incorrect or partially correct. For an example, suppose that we have to predict an instance that belongs to both (Swimming, Reading) labels, we may get one of the following results:

1- Swimming , Reading (fully correct)
2- Swimming , Writing (partially correct)
3- Reading , Writing (partially Correct)
4- TV Watching , Traveling (fully incorrect)

The above results differ from each others in the degree of correctness.

In (Schapire et al., 2000) three kinds of measures were used to customize ranking tasks: one-error, coverage, and precision.

One-error evaluates how many times the top-ranked label is not in the set of ground truth labels.

$$One-error, O = \frac{1}{n}\sum_{i=1}^{n} I(\arg\min_{\lambda \in \mathcal{L}} r_i(\lambda) \notin Y_i^l) \quad (2.1)$$

Coverage measures how far one needs, on average, to go down the list of labels in order to cover all ground truth labels.

$$Coverage, C = \frac{1}{n}\sum_{i=1}^{n} \max_{\lambda \in Y_i} r_i(\lambda) - 1 \quad (2.2)$$

Precision is a measure which is borrowed from information retrieval (IR) that measures the percentage of positive predictions that were correct. All of the above measures are used in single-label classification, but they do not fit well with multi – label classification.

Several measures have been proposed in the literature of the evaluation of multi-label classifiers. In the next paragraph, a brief description of the most famous measures is shown.

Hamming Loss: A measure that is interested in errors prediction (incorrect labels) and missing errors (Labels that are not predicted) .The optimal value of Hamming loss is 0 . The smaller the value of hamming loss is, the better the performance is (Schapire et al. , 2000).

$$\text{Hamming Loss} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \Delta Z_i|}{|L|}, \tag{2.3}$$

Where |L|: number of labels, |D|: number of instances in the training dataset, Y_i: set of ground truth labels, Z_i: set of predicted labels, Δ: Symmetric difference

Accuracy: Accuracy measures how close Y_i is to Z_i (Godbole et al. , 2004)

$$\text{Accuracy} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{2.4}$$

Precision (confidence): is the percentage of true positive examples from all the examples classified as positive by the classifier.

$$\text{Precision} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{2.5}$$

Recall (sensitivity): is the percentage of examples classified as positive by classifier that is true positive

$$\text{Recall} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{2.6}$$

Subset Accuracy: A very constructive accuracy metrics which considers a classification as correct if all the labels predicted by a classifier are correct.(Ghamrawi et al. ,2005)

Where N: total number of instances.

$$\text{Subset Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I\left(|Z_i| = |Y_i|\right) \tag{2.7}$$

Harmonic Mean (F1 Measure): harmonic mean of precision and recall.

Harmonic Mean (F1 Measure) = $$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|}$$ (2.8)

As in single label and multi-class classification, the higher the value of accuracy, precision, recall, and F1- measure, the better the performance of the learning algorithm.

Example:

Table 2.6 Multi-label data set

Instance Number	Y_i	Z_i
1	{C1,C3}	{C1,C4}
2	{C2,C4}	{C2,C4}
3	{C1,C4}	{C1,C4}
4	{C2,C3}	{C2}
5	{C1}	{C1,C4}

Accuracy = (1/3 + 2/2 + 2/2 + 1/2 + 1/2) / 5 = 0.667

Precision = (1/2 + 2/2 + 2/2 + 1/1 + 1/2) / 5 = 0.80

Recall = (1/2 + 2/2 + 2/2 + 1/2 + 1/1) / 5 = 0.80

Harmonic Mean (F1 Measure) = ((1/4 + 2/4 + 2/4 + 1/3 + 1/3) *2) / 5 = 0.77

Hamming Loss = (2 + 0 + 0 + 1 + 1) / 5 / 4 = 0.20

In (Thabtah et al, 2005), four measures are presented to evaluate the accuracy of classification approaches to a wide range of traditional and multi-label classification problems

1- Top-label: an evaluation measure that is interested in only the top-ranked class label. It estimates how many times the top-ranked class label is the correct class label.
2- Any-label: an optimistic evaluation method that considers the classification result as correct if any of the predicted class label of a test data object matches the true class.
3- Label-weight: this method gives the ability to every class to play a role in classifying a test object based on its ranking. Each class can be assigned a weight according to how many times that class has been associated with the object.
4- Support-weight: This evaluation measure gives the top-ranked label the maximum weight, and each of the rest labels a weight equals to the number of times that the label is associated with the instance divided by the number of times it is associated with the top-ranked label.

Receiver Operating Characteristics (ROC) Curve

ROC curve is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings. TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate. The ROC is also known as a relative operating characteristic curve, because it is a comparison of two operating characteristics (TPR and FPR) as the criterion changes.

In ROC curve , The TPR is plotted among the y axis, and FPR is plotted on the x axis, each point along the curve corresponds to one of the models inducted by the classifier. If the model is perfect then its area under the ROC curve would equal one. A model that is strictly better than another would have a larger area under the ROC curve. Finally ROC curve is useful tool for comparing the relative performance among different classifiers.

2.5 Rule Based Classification Algorithms

Rule based classification algorithms such as IREP (Furnkranz and Widmer, 1994), RIPPER(Cohen,1995), PART(Frank and Witten,1998) and PRISM(Cendrowski, 1988) present their output as a set of "if-then" rules, which makes it easy for the end-user to understand and interpret the classifier. Moreover, unlike decision tree algorithms, one can update or tune a rule in rule based classification algorithms without affecting the complete rules set, where as the same task requires reshaping the whole tree in decision tree approach.

Other advantages of rule-based classifiers are

- Easy to generate.
- Can classify new instance rapidly
- As highly expressive as decision trees.
- Performance comparable to decision trees.

Two approaches are used in rule based classification algorithms; the first approach directly learns the rules from the training data. In the second approach, the rules are constructed in indirect fashion such as in the case of learning a decision tree , then convert it to rules, or in the case of learning neural networks and then convert it to rules.

2.5.1 Incremental Reduced Error Pruning(IREP)

IREP was proposed in 1994 by Furnkranz and Widmer with the aim of integrating a separate-and-conquer approach with Reduced Error Pruning. REP is a classification method with an efficient ability to produce and prunes a small set of classification rules. REP keeps a part of training data as an independent test data which is used to estimate the error at each node of the decision tree. IREP build a rule set in greedy manner, it randomly partitioned the data into a growing set and pruning set, growing set contains 66.7% of the training data objects. After that the process of constructing the rules began in a greedy fashion ,beginning with an empty rule.

Then First–Order-Inductive-Learner(FOIL)-gain measure is used to determine which condition to add. IREP continuously adds conditions that maximize Foil-gain value, to the current rule until the rule covers no data objects from the growing set. After a rule is built,

IREP immediately considers pruning it backwards by removing the final sequence of conditions from it. Starting from the last condition for each generated rule, IREP considers removing one condition at a time and chooses the deletion that improves the following function:

$$v = (rule, p, n, P, N) \equiv \frac{p(N-n)}{P+N} \tag{2.9}$$

where P, N are the total numbers of data objects in the pruning set and p, n are the numbers of data objects in the pruning set covered by the pruned rule. The process of pruning a rule is stopped once no deletion improves the value v . Once a rule is pruned, it will be inserted into the classifier and all data objects associated with it are removed from the growing and pruning sets.

2.5.2 Repeated Incremental Pruning to Produce Error Reduction (RIPPER)

(Cohen,1995) developed a rule induction algorithm and called it Repeated Incremental Pruning to Produce Error Reduction algorithm (RIPPER). This algorithm constructs the rules as following: first the training data is divided into two parts, a pruning set and a growing set. Then in a repeated process and using the previous two set, RIPPER constructs the classifier starting from an empty rule set and heuristically adding one condition at a time till the error on the growing set is minimized.

We could describe RIPPER as an enhancement version of IRIP with some modifications as follows:

- IREP stops adding rules as soon as a rule learned has an error rate greater than 50% on the pruning data, which could be an early stopping, especially in application domains with large number of low coverage rules. On the other side RIPPER stops adding a rule using the Minimum Description Length principle (MDL) where after a rule is inserted, the total description length of the rules set and the training data is estimated. If this description length is larger than the smallest MDL obtained so far, RIPPER stops adding rules. The MDL assumes that the best model (set of rules) of

data is the one that minimizes the size of the model plus the amount of information required to identify the exceptions relative to the model.

- Another important modification is an optimization procedure that cuts down the number of rules derived by pruning the discovered rules set. This post-pruning process has been applied to the classifier produced by IREP as an optimization phase, aiming to simplify the rule set features. For each rule r_i in the rule set, two alternative rules are built; the replacement of r_i and the revision of r_i. The replacement of r_i is created by growing an empty rule $_i r'$ and then pruning it in order to reduce the error rate of the rules set including $_i r'$ on the pruning data set. The revision of r_i is constructed similarly except that the revision rule is built heuristically by adding one condition at a time to the original r_i rather than to an empty rule. Then the three rules are examined on the pruning data to select the rule with the least error rate. The integration of IREP and the optimization procedure forms the RIPPER algorithm.

2.5.3 PRISM

Prism was developed by Cendrowski (Cendrowski, 1988) and can be categorized as a covering algorithm for constructing classification rules. The covering approach starts by taking one class among the available ones in the training data set, and then it seeks a way of covering all instances to that class, at the same time it excludes instances not belonging to that class. This approach usually tries to create rules with maximum accuracy by adding one condition to the current rule antecedent. At each stage, Prism chooses the condition that maximizes the probability of the desired classification. The process of constructing a rule terminates as soon as a stopping condition is met. Once a rule is derived, Prism continues building rules for the current class until all instances associated with the class are covered. Once this happens, another class is selected, and so forth.

2.6 Summary

In this chapter, we have introduced the definition of multi-label classification problem. Methods that handle multi-label classification problem cab be divided into two groups: problem transformation methods, which transform multi-label problem into one single label problem or more, and algorithm adaptation methods, which extend single label learning algorithm to handle multi-label data. Also, we have discussed some of the most important evaluation measures, that are used for both single and multi-label classification, such as accuracy, precision, recall, hamming loss, harmonic mean, and many other evaluation measures. A brief description of some rule-based classifier has been discussed, and examples of rule-based classifier were introduced in the last section.

Chapter Three

The Proposed Model: Development of Multi-Label Classification Algorithm based on Labels Correlations

3.1 Introduction

Based on the previous literature review of multi-label classification, we can assure that, there is no guided multi-label classification algorithm, which seeks the important correlations among labels before learning. No guided algorithm that tries to capture the important correlations among labels in order to reduce problem search space could be found in multi-label classification literature. Therefore we are proposing a guided multi-label classification algorithm based on correlations among labels in class label attribute and then applying a classic classification DM algorithm to learn rules from the training dataset.

Most of multi-label classifications methods, both problem transformation methods and algorithm adaptation methods depend - for its classification task- on a function that maps between the attributes and the labels in the training data. The proposed model introduces new approach to solve the problem of multi-label classification. This approach is based on correlations among labels learned by predictive classification, which try to answer a major research question, that is: what can we gain when capturing the important correlations among different labels?
Other questions could be inspired from the previous major question such as

- How label's cardinality and diversity distinguish multi-label data set from each other?
- What is the relationship between label's cardinality and the accuracy of the classifier?
- To what degree labels are correlated with each others?
- How can we benefit from positive association among labels to produce multi-label classifier?

In the following sections, we introduce the proposed model, and the evaluation process of the model, using two multi-label datasets and some of the most important evaluation measures.

3.2 General Structure of the Proposed Model

The proposed model consists of three phases: a) transforming multi-label dataset into single label dataset and discovering correlations among labels. b) Applying a rule-based classification algorithm on the transformed dataset. c) Generating the multi-label rules based on the output of the rule-based classifier and the correlations among labels. Figure 3.1 shows the general structure of the proposed model.

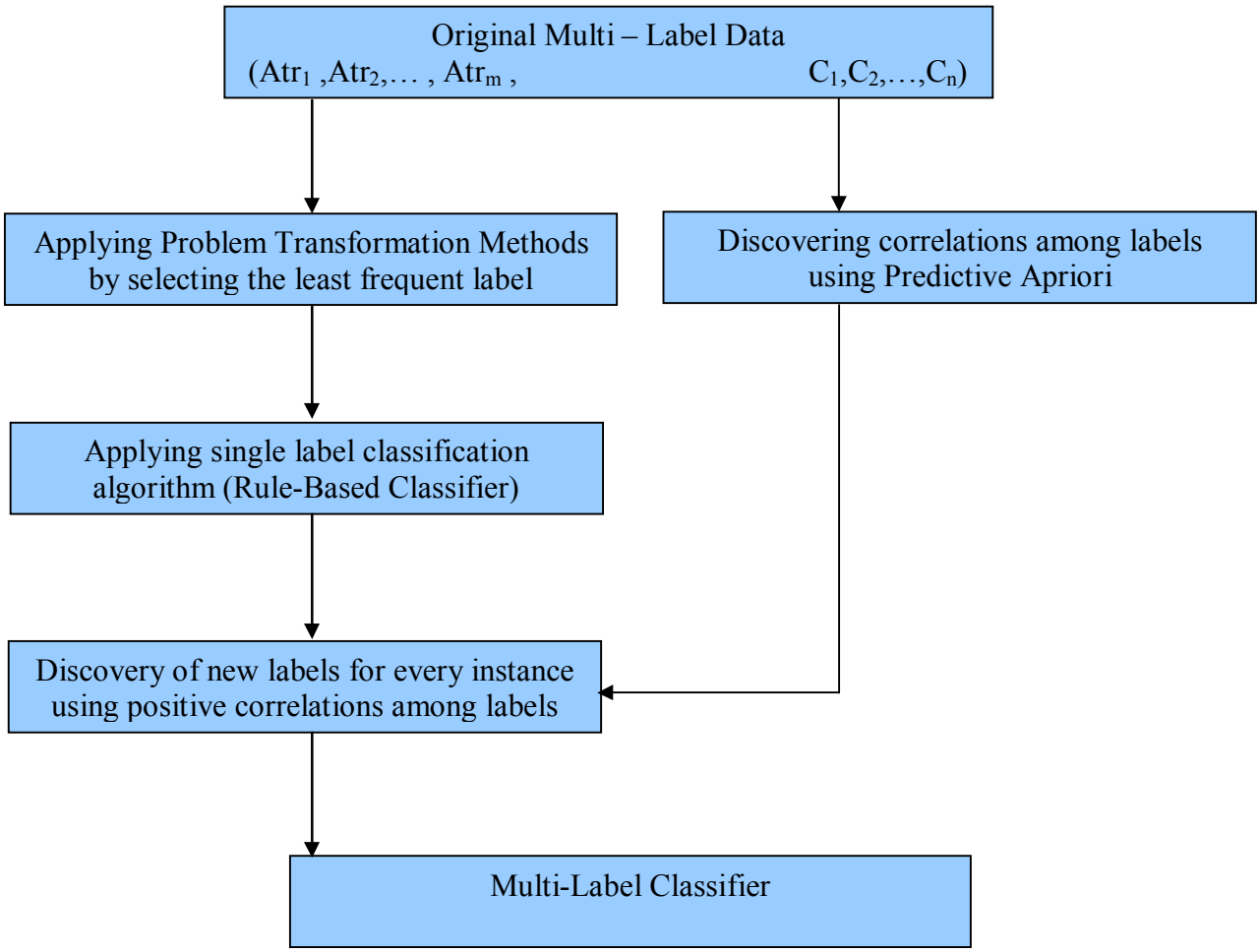

Fig. 3.1 General structure of the proposed model

As we can see in the previous figure, the input of the algorithm is a multi-label dataset. Two operations are performed on the multi-label dataset in parallel.

The first operation is transforming multi-label dataset into single label dataset, where we have many methods to choose such as selecting the most frequent label, selecting the least frequent label or select any label randomly. For our proposed model we choose to select the least frequent label as transformation criteria.

The second operation that is performed in the multi-label dataset is to find all positive association among labels using predictive Apriori (Scheffer, 2001). This operation tries to associate each label with labels from the label set; if that is possible.

So, after performing the previous two operations we will have:

1. Single label dataset which has been extracted or transformed from multi-label dataset using the least frequent label criteria.
2. Rules between labels with different rule's cardinality, starting from cardinality one up to rule's cardinality=dataset cardinality -1.

Now, we are ready to apply single rule-based classifier on the transformed dataset. Many rule-based classifiers could be used in this stage such as RIPPER, IREP, PART or Prism. The outputs of any single rule based classifier will be set of "If-Then" rules with one consequent on the right-hand-side of the rule like the following rule:
IF (con_1 and con_2 and … con_n) Then Label. Using both, output of the single rule based classifier and rules based on the correlations among labels previously discovered, we will be able to build multi-label rules classifier in the form of
IF (con_1 and con_2 and … con_n) Then {Label1, Label2,… $Label_n$ }.

The last step in the proposed model is the evaluation of the outputted model. This evaluation will be carried out using different evaluation measures which are: Accuracy, Hamming Loss, and Harmonic Mean (F1 Measure). These evaluation measures are explained in section 2.3. The main steps of the proposed model are described in algorithm 1.

Algorithm 1: input: Multi-label dataset as training data. Output: A set of Multi-Label rules.
Phase 1:

a. Transforming multi-label dataset into single label dataset by selecting the least frequent label associated with each training instance.
b. For every label in the label set of the dataset, find the highest accuracy positive rule in the form of: IF label X exists THEN label Y exists.

Phase 2:

a. Applying a rule based classifier on the transformed data set and producing the rules set.

Phase 3:

a. Generating the multi-label rules set, using the single rules set produced by the classifier in Phase 2, and the associative rules for each instance that has been discovered in phase 1.

3.3 Data Representation

All of multi-label datasets that have been used in this thesis are structured datasets, which vary from each others in the number of instances, number of attributes, number of labels, and also types of attributes (nominal, numeric).Table 3.1 describes information about the datasets which have been used in the thesis. These datasets are downloaded from the following address (http://mulan.sourceforge.net/datasets.html).

Table 3.1 Multi-label dataset information

Dataset name	Domain	# of Instances	# of Attributes	
			Nominal	Numeric
Emotions	Music	593	0	72
Yeast	Biology	2417	0	103

3.4 Data Transformation

Many data transformation methods could be used to transform multi-label dataset into single label one such as selecting the least frequent label, selecting the most frequent label or simply selecting any label randomly. In our thesis, we have made some experiments on "Emotions" dataset to discover which transformation method to select. We have found that using most frequent label as a transformation criteria yields to low accuracy of the classifier (0.451) while, when using least frequent label as a transformation criteria the accuracy of the classifier is (0.767). We conclude that using least frequent label as transformation criteria is more suitable choice for the proposed model in addition to solving the problem of imbalance class distribution.

Table 3.2 "Emotions" dataset labels statistics

Label	Amazed	Happy	Relaxing	Quite-still	Sad	Angry
Frequent	173	166	264	148	168	189

Table 3.2 shows that the emotions dataset contains six labels, and after counting how many times these labels have been used in the dataset we will have:

Most Frequent Label: "Relaxing"

Least Frequent Label: "Quite-still"

The above table will be used to transform multi-label dataset into single label one by using the least frequent label as shown in table 3.3

Table 3.3 transforming multi-label dataset into single label dataset using least frequent label

Amazed	Happy	Relaxing	Quite-Still	Sad	Angry	Class
0	0	1	1	1	0	quite
1	0	0	0	0	1	amazed
0	0	0	0	1	0	sad
0	1	1	0	0	0	happy
0	0	0	0	1	0	sad
0	0	1	0	1	0	sad

As we can see in the previous table, the first example is associated with three labels at the same time (Relaxing, Quite-Still, Sad), and since "Quite-Still" has frequent 148 which is less than the frequent of "Relaxing" (264) and "Sad" (168), it will be transformed to the

single label "Quite". The second example is associated with two labels: "Amazed" with frequent equals to 173 and "Angry" with frequent 189, so it was transformed to the least frequent label which is "Amazed", and so on for the rest of examples.

3.5 Learning Step

The learning step in our proposed model consists of two different tasks. The first task is an unsupervised learning task, which aims to discover the correlations among labels using Predictive Apriori. While the second task is a supervised learning task that aims to predict the class label of unseen instance as accurate as possible using a rule based classifier.

3.5.1 Discovering of Positive Correlations among Labels.

Suppose we have the itemsets (Labels) C1, C2, and C3. We are interested in having association rules with good confidence between every possible Pairwise of the three previous labels. For the first two labels C1, C2 we may have the following rules for example:

1- If C2=1 Then C1=0
2- If C1=1 Then C2=1

In our proposed model, we are interested in a rules like the second rule, we are looking for a rule in a form of (If label x exists Then label y exists).

For each label(x) in the dataset we want to find another label(y) that has a positive correlation with it, i.e. label(x). In case we have more than one label positively associated with the label in the antecedent, we select the rule with the highest confidence or accuracy. For example suppose that we have the following association between C1, C2 and C3:

1- If C1 =1 Then C2=1 (Accuracy = 0.80)
2- If C1=1 Then C3=1 (Accuracy = 0. 71)

In the previous case, we choose the rule with the highest accuracy, so rule one will be selected, and rule two is ignored. In fact ignoring such a rule with a meaningful confidence such 0.71 may cause too much information loss but let us stuck on the choice of selecting the best rule, and leave ignoring other rules with meaningful confidence to be discussed later in the future work section.

After having all positive associations of length "1" between labels in the dataset , we move forward to find all positive associations of cardinality "2" as the following rule (If C1=1 and C2=1 Then C3=1) and so forth.

For the proposed model , we will choose the rule with the highest accuracy without any pre specified condition about the value of accuracy, such as the accuracy should be grater than or equals to predefined user threshold. For example, suppose we have the following rules:

1- If C1=1 Then C2=1 (Accuracy = 0.27)
2- If C1=1 Then C3=1 (Accuracy = 0.19)

In such a case, we will select the first rule even if it has a low accuracy. In future, we will experiment the choice of neglecting rules that have accuracy less than some pre defined user threshold.

Table 3.4 contains the correlations among labels after applying predictive Apriori on "Emotions" dataset.

Table 3.4 Positive Association Rules among Labels for Emotions dataset

Rule's Number	Rule	Accuracy
1	If amazed then angry	0.53
2	If happy then relaxing	0.44
3	If Quite-still then sad	0.71
4	If Sad then Relaxing	0.57
5	If angry then Relaxing	0.03
6	If Relaxing then Relaxing	1.00

Rule "5" has a low accuracy, but we will stuck in the choice of having the highest positive association among labels, and since no other rule could be found to be associated with the label "angry", and has accuracy greater than this rule, this rule is chosen.

3.5.2 Applying Rule-Based Classifier.

After having the transformed version of "Emotions" data set, and finding the highest positive association rules among labels, we are ready to apply any single rule-based classification algorithm to the transformed data, and we choose PART classifier.

PART is a rule-based classification algorithm that combines between to approach. The first one is creating rules using decision tree, and the second one is separate and conquer learning method. The algorithm produces accurate rules in the same size as those generated by decision tree C4.5 algorithm

The step of applying a rule-based classification algorithm on the transformed dataset is very important in building multi-label rules, with the help of association rules among labels. PART algorithm has been chosen for being accurate, efficient and fast.

Let us give a sample rule from the rules set that we've got after applying PART algorithm on the transformed dataset. The sample rule is:(To see which features those conditions represent see the appendix)
If AQ > 0.217678 AND B <= 0.090652 AND V > 0.580398 AND AZ > 3.787686 AND AX > 0.060033 AND BD <= 0.173826 then Sad**.**

Using Association rules among labels that have been discovered earlier, and since there is a rule indicates that (If sad then Relaxing), we could rebuild the rule that had been discovered from the rule based classifier as following:

If AQ > 0.217678 AND B <= 0.090652 AND V > 0.580398 AND AZ > 3.787686 AND AX > 0.060033 AND BD <= 0.173826 then {Sad, Relaxing}

We repeat the previous process for all rules extracted from the rule based classifier and using the association rules discovered in the first step until we have the complete set of multi-label rules, which will be used to classify the test instances.

3.6 Prediction Step

Set of multi-label rules have been learned form both correlations among labels and rule-based classifier outputs. These multi-label rules will be used in the prediction step, and an evaluation process will be done using some important evaluation measurements.

3.7 Complete Example for the Proposed Model

In this section, we show a complete step by step example for the proposed model, and using "Emotions" dataset. The first step in the proposed model is to transform "Emotions" dataset into single label dataset, and using least frequent label, as in table 3.5

Table 3.5 transforming "Emotions" dataset into single label dataset

Amazed	Happy	Relaxing	Quite-Still	Sad	Angry	Class
0	0	1	1	1	0	quite
1	0	0	0	0	1	amazed
0	0	0	0	1	0	sad
0	1	1	0	0	0	happy
0	0	0	0	1	0	sad
0	0	1	0	1	0	sad

The second step is to find positive correlations among labels using predictive Apriori. Best correlations are chosen without determining any threshold value in this stage, and since "Emotions" dataset is of cardinality "2"; association rules will be with "1" condition only in the antecedent as we have mention earlier in section 3.2. Table 3.5 shows the complete positive correlations among labels in "Emotions" dataset.

Table 3.6 positive correlations among labels in "Emotions" dataset

Rule's Number	Rule	Accuracy
1	If amazed then angry	0.53
2	If happy then relaxing	0.44
3	If Quite-still then sad	0.71
4	If Sad then Relaxing	0.57
5	If angry then Relaxing	0.03
6	If Relaxing then Relaxing	1.00

The third step in the proposed model is to apply a rule based classification algorithm on the transformed dataset that has been achieved from the first step. Table 3.6 shows some of the

learning rules discovered after applying "PART" classifier on the transformed "Emotions" dataset.

Table 3.7 learning rules discovered after applying "PART" classifier on the transformed "Emotions" dataset

Rule's Number	Rules
1	IF AQ > 0.217678 AND B <= 0.090652 AND V > 0.580398 AND AZ > 3.787686 AND AX > 0.060033 AND BD <= 0.173826 Then Sad.
2	IF AQ <= 0.215792 AND BJ <= 0.108461 AND J <= 1.021892 AND BO <= 0.066288 Then Angry
3	IF AS > 0.206592 AND AI > 0.010202 AND D > -76.700621 Then Amazed
4	IF AS > 0.206592 AND AI > 0.010202 AND B <= 0.191563 Then Quit-Still
5	IF AS > 0.208738 AND B <= 0.119991 AND AP > 0.213677 AND BN <= 102 AND D > -75.367339 Then Relaxing
6	IF G > 2.024609 AND E > 3.112653 Then Happy

The last step in the proposed model is to build multi-label classifier based on correlations among labels and rules discovered from applying a rule based algorithm on the

transformed dataset. Table 3.7 summarizes some of the multi-label rules discovered from "Emotions" dataset.

Table 3.8 multi-label rules discovered from "Emotions" dataset.

Rule's Number	Multi-Label Rules
1	IF AQ > 0.217678 AND B <= 0.090652 AND V > 0.580398 AND AZ > 3.787686 AND AX > 0.060033 AND BD <= 0.173826 Then {Sad, Relaxing}.
2	IF AQ <= 0.215792 AND BJ <= 0.108461 AND J <= 1.021892 AND BO <= 0.066288 Then {Angry, Relaxing}
3	IF AS > 0.206592 AND AI > 0.010202 AND D > -76.700621 Then {Amazed, Angry}
4	IF AS > 0.206592 AND AI > 0.010202 AND B <= 0.191563 Then {Quite-Still, Sad}
5	IF AS > 0.208738 AND B <= 0.119991 AND AP > 0.213677 AND BN <= 102 AND D > -75.367339 Then {Relaxing}
6	IF G > 2.024609 AND E > 3.112653 Then {Happy, Relaxing}

3.8 Distinguishing Features for the Proposed Model

The proposed model has some of distinguished features over other multi-label classification methods such as:

- Merging between two different learning tasks, the first task is an unsupervised learning task, which is the task of finding positive association among labels. The second task is a supervised learning task, which is the task of applying any rule-based classifier on the transformed dataset.
- Getting benefits from finding the correlations among labels, in the process of generating multi-label rules. Transforming multi-label dataset into single label dataset causes too loss in information, and by finding correlations among labels, the proposed model tries to substitute this information loss.
- The proposed model has much flexibility, since any rule-based classifier could be used in the process of classifying the transformed data set.

3.9 Summary

In this chapter we have proposed a multi-label classification algorithm based on correlations among labels. And give extra details for every step in the proposed algorithm. We have used Predictive Apriori for discovering positive correlations among labels and PART algorithm has been applied on the transformed dataset. We have used the least frequent label criteria as a transformation method to solve the problem of imbalance class distribution.

Chapter 4

Data and Experiments

4.1 Data

In this thesis, we use two different application domains which they are: Biological, and Music. For each application domain, one multi-label dataset has been used, as shown in table 4.1. Both datasets and many others datasets are available at http://mulan.sourceforge.net/datasets.html

The first dataset is called "Emotions" and it is concerned about songs according to the emotions they evoke. This data set contains six labels, with label cardinality (LC) and label density (LD) equal to 1.869, 0.311 respectively. There are 27 distinct labelsets (DLS) in a total number of 593 examples in this dataset.

As previously mentioned label cardinality (LC) is the average number of labels per example, while label density is the same number (LC) divided by number of labels in the dataset (6 in the emotion dataset as an example).

The second dataset is called "Yeast" which is concerned about protein function classification. This dataset contains 2417 examples with 198 distinct labelsets. Yeast has 14 different labels with cardinality equals to 4.327 and density equals to 0.303.

Table 4.1 Multi-label datasets statistics

Dataset	# of Examples	# of Labels	DLS	LC	LD
Emotions	593	6	27	1.869	0.311
Yeast	2417	14	198	4.327	0.303

From all the statistics mentioned in Table 4.1 we are more interested in LC to determine the association's cardinality according to the following equation:

$$\text{Association rule's cardinality} = \text{Label Cardinality} - 1 \quad (4.1)$$

The next two tables summarize the labels that could be found in the datasets which will be used in the evaluation process and the frequency of each label.

Table 4.2 "Emotions" Dataset Labels Frequency

Label	**Amazed**	**Happy**	**Relaxing**	**Quite-still**	**Sad**	**Angry**
Frequency	**173**	**166**	**264**	**148**	**168**	**189**

Table 4.3 "Yeast" Dataset Labels Frequency

Label	C1	C2	C3	C4	C5	C6	C7
Frequency	762	1038	983	862	722	597	428
Label	C8	C9	C10	C11	C12	C13	C14
Frequency	480	178	253	289	289	1799	34

4.2 Experiments on "Emotions" Dataset

An extensive evaluation process has been done, using three evaluation measures, five problem transformation methods, two algorithm adaptation methods. All experiments were conducted on Intel core i3, 2.10 GHz (4 CPU) PC under Windows 7 Ultimate 32-bit.

All of multi-label classification methods and also all supervised learning algorithms which are used in this thesis are implemented using Mulan. Mulan is a Weka-based Java package for multi-label classification.

All experiments were conducted using the 10-fold cross validation measure. Data were divided into two parts learning part and testing part. Learning part is nearly 25%, while testing part is nearly 75% of the complete dataset.

4.2.1 Accuracy

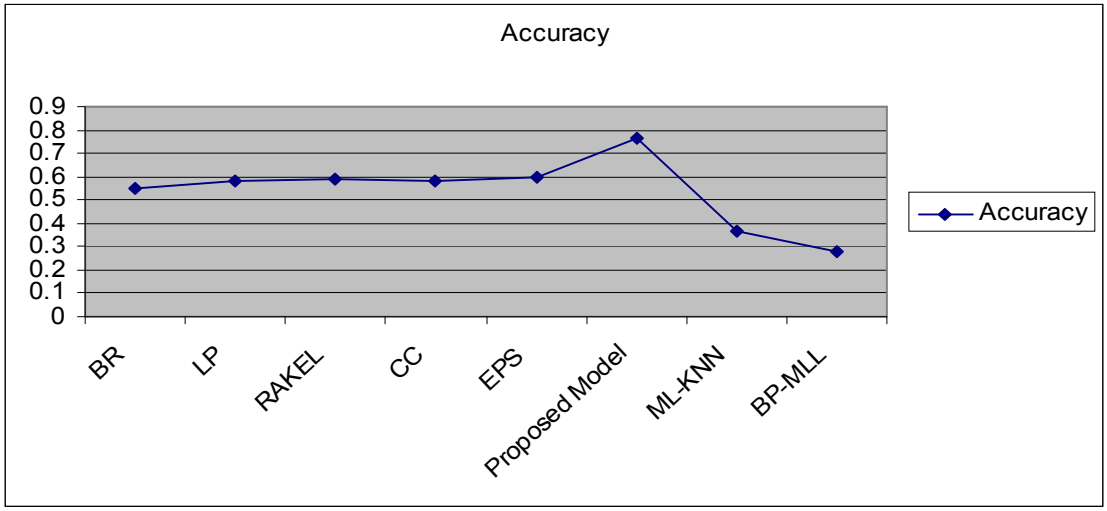

Figure 4.1 Difference in accuracy between the proposed model and different methods

As we can see from figure 4.1, the proposed model has the highest accuracy (0.767) among all the multi-label classification methods. The second best accuracy is 0.592 achieved by RAKEL. This indicates that using correlations among labels increase accuracy in a great way.

4.2.2 Hamming Loss

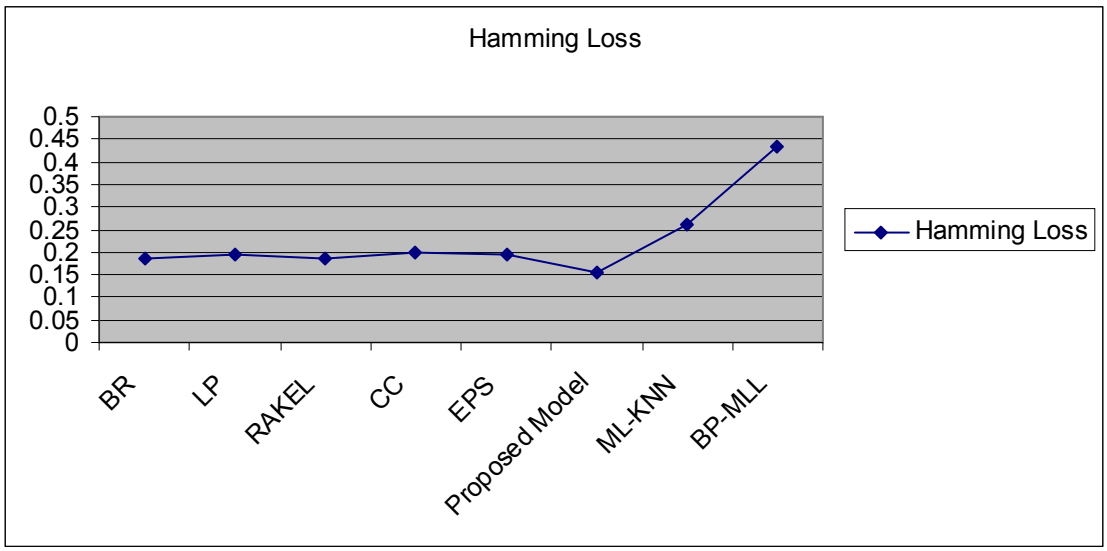

Figure 4.2 Difference in Hamming Loss between the proposed model and different methods

As we can see from figure 4.2, the proposed model has the lowest Hamming Loss (0.155) among all the multi-label classification methods. The second best hamming lost is achieved by RAKEL method (0.186), which indicates that the proposed model decreases both incorrect labels classification and missing labels classification in a good way.

4.2.3 Harmonic Mean (F1 Measure)

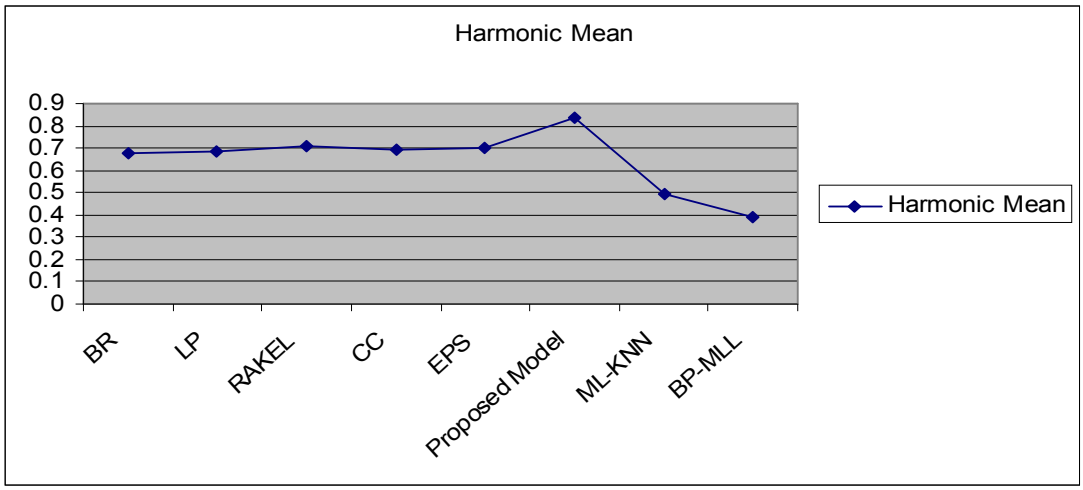

Figure 4.3 Difference in Harmonic Mean between the proposed model and different methods

As we can see from figure 4.3, the proposed model has the highest Harmonic Mean (0.837) among all multi-label classification methods.

4.3 Experiments on "Yeast" Dataset

Table 4.4 contains the best correlations among labels after applying Predictive Apriori on "Yeast" dataset.

Table 4.4 Positive Association Rules of "Yeast" dataset

Rule #	Rule	Accuracy
1	If C1 then C2	0.49
2	If C2 then C12	0.43
3	If C3 then C12	0.50
4	If C4 then C12	0.51
5	If C5 then C12	0.53
6	If C6 then C12	0.54
7	If C7 then C8	0.63
8	If C8 then C13	0.50
9	If C9 then C8	0.81
10	If C10 then C11	0.82
11	If C11 Then C12	0.76
12	If C12 then C12	1.00
13	If C13 then C12	0.80
14	If C14 then C4	0.99

Table 4.4 summarizes the results of the evaluation measures on "Yeast" dataset. Five problem transformation methods and two algorithm adaptation methods are used. Table 4.5 shows that the proposed model has the highest accuracy (0.554), and EPS method has the second highest accuracy (0.537). The proposed model has the best value for Hamming loss (0.161), while BR and ML-KNN have the second best value (0.193). Finally, the proposed model has the best value (0.672) of Harmonic mean measure, and ML-KNN has the second best value (0.654) of Harmonic mean.

Table 4.5 Evaluation results of "Yeast" dataset

Method	**Accuracy**	**Hamming Loss**	**Harmonic Mean**
BR	0.522	0.193	0.652
LP	0.530	0.206	0.643
RAKEL	0.493	0.207	0.559
CC	0.521	0.211	0.633
EPS	0.537	0.207	0.654
Proposed Model	**0.554**	**0.161**	**0.672**
ML-KNN	0.520	0.193	0.654
BP-MLL	0.185	0.322	0.210

4.3.1 Accuracy

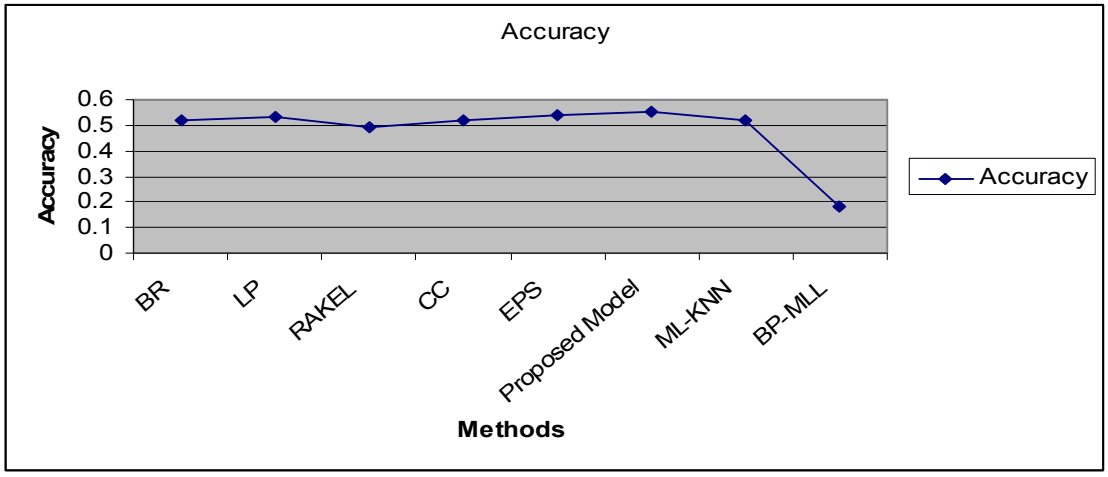

Figure 4.4 Difference in accuracy between the proposed model and different methods

As we can see from figure 4.4, the proposed model has the highest accuracy among all the multi-label classification methods.

4.3.2 Hamming Loss

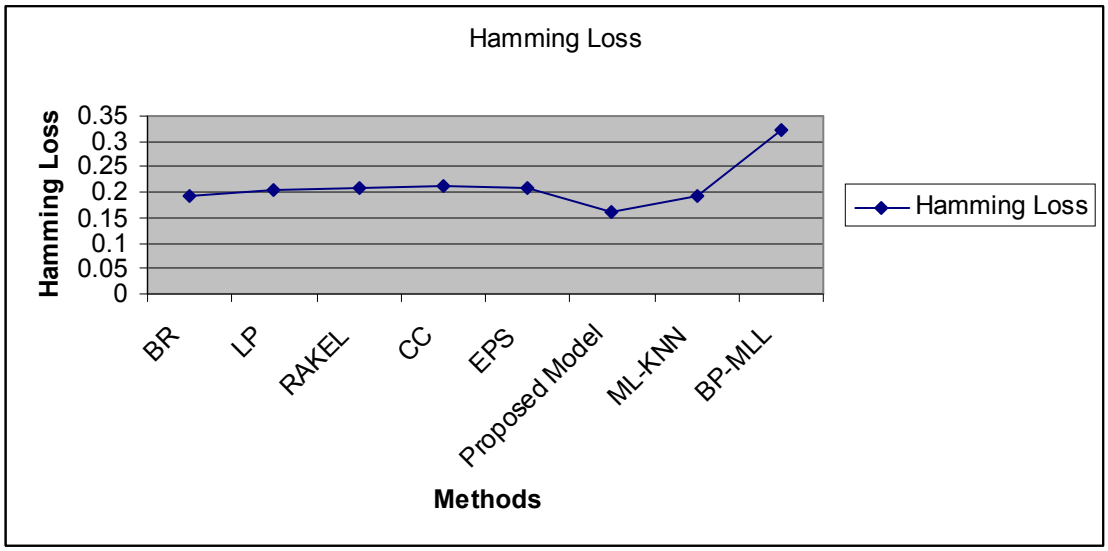

Figure 4.5 Difference in Hamming Loss between the proposed model and different methods

As we can see from figure 4.5, the proposed model has the minimum Hamming Loss among all the multi-label classification methods.

4.3.3 Harmonic Mean (F1 Measure)

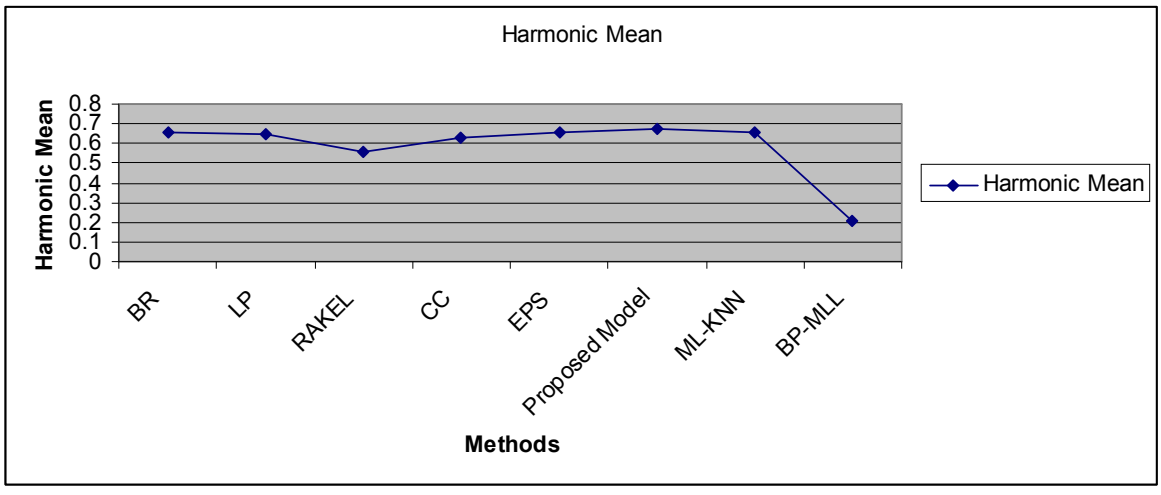

Figure 4.6 Difference in Harmonic Mean between the proposed model and different methods

As we can see from figure 4.6, the proposed model has the highest Harmonic Mean among all the multi-label classification methods.

4.4 Summary

In this chapter, we have introduced the evaluation process of multi-label classification algorithm based on correlations among labels. The proposed algorithm was evaluated using two different multi-label datasets, and contrasted with seven different classification methods of both types: problem transformation methods and algorithm adaptation methods. Further, three evaluation measures including: Accuracy, Hamming Loss, and Harmonic Mean. Final results indicate that our proposed model is effective, consistent and has a higher classification rate than many other multi-label classification methods, and this is due to using correlations among labels to build multi-label classifier.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we have investigated the problem of multi-label classification, and the benefits from having the correlations among label in building multi-label rules. The output is an algorithm for multi-label classification based on correlations among labels. Unlike previous approaches, this algorithm combines between problem transformation methods with the criteria of selecting least frequent label and unsupervised learning method (Predictive Apriori). We summarize our contributions in this section.

5.1.1 Issue 1: Benefits of Discovering Correlation among Labels in Multi-Label Classification Problem

There are many methods for handling multi-label classification problem. These methods fall into two groups: problem transformation methods and algorithm adaptation methods. The first group transforms multi-label data into single label data, and then applies any single label classification algorithm. This causes much information loss, especially in correlations among labels. The second group adapts single label classification algorithm to handle multi-label dataset. This leads to increase complexity and inherits all of the single label classification algorithm drawbacks.

The proposed model employs problem transformation methods because of its simplicity, and defeats information loss caused by problem transformation methods through correlations among labels. The idea is new and simple but it has a great impact on solving the problem of multi-label classification. Experiments on "Emotions" dataset show that: using least frequent label as transformation criteria is more suitable and has better accuracy than using most frequent label as transformation criteria, when applying the proposed model.

5.1.2 Issue 2: How label's cardinality and diversity distinguish multi-label data set from each other?

Perhaps, the most important characteristics of the multi-label dataset is the average number of labels per example (LC). In general, Dataset with high LC is more complex to classify than one with low LC, and as LC increases, the accuracy of the classifier decreases and vise versa. Other important factor that has a great influence in multi-label classification problem is the activity for each label in the label set. The label is said to be active, if it has more than one strong correlation with other labels such as the following case that has been discovered after applying predictive Apriori on labels of "Emotions" dataset:

1- If Quite Then Sad (Accuracy = 0.71)
2- If Quite Then Relaxing (Accuracy = 0.70)
3- If Quite Then Happy (Accuracy = 0.44)

Label "Quite" is an active label, since it has three strong correlations with "Sad", "Relaxing", and "Happy". In such a case, the proposed model, fires rule "1" since it has the highest accuracy, and neglect rule "2" and rule "3", which leads to information loss. Solution for this case is to enhance the proposed model to consider disjunction case like the following rule:
If Quite Then {Quite, Sad} or {Quite, Relaxing} or {Quite, Happy}

High LC and active labels increase the complexity of solving multi-label classification problem, but at the same time help to design a hierarchical structure for multi- label to manage label correlationships.

In general, the proposed model gives a great success when LC is greater than or equal 2, while it is useless to use the proposed model when LC is close to 1, such as in "Scene" dataset, where LC is nearly 1.07.

5.2 Future Work

5.2.1 Proposing New Problem Transformation Method based on Accuracy of correlations among labels

We may adapt the proposed model as following:

Step1: Discovery of positive correlations among labels

Step2: Apply problem transformation method based on association among labels and using the highest accuracy criteria, which means to select the label that produces the highest accuracy as being antecedent of the association rule.

Step3: Applying a rule based classifier on the transformed data set and producing the rules set.

Step4: Generating the multi-label rules set, using the single rules set produced by the classifier in step 3, and the associative rules for each instance that has been discovered in step 1.

Experiment on "Emotions" dataset shows that the adapted model is promising and need to be studied more. When applying the adapted model in "Emotions" dataset, the accuracy was (0.752) which is really close to the accuracy of the proposed model (0.767).

5.2.2 Disjunction Case

In "Emotions" dataset, we found the following positive association rules.

1- If Quite Then Sad (Accuracy = 0.71)

2- If Quite Then Relaxing (Accuracy = 0.70)

3- If Quite Then Happy (Accuracy = 0.44)

The proposed model select the rule with highest accuracy, and ignore others , so rule "1" is selected, and rule "2", rule"3" are ignored. This leads to information loss, and could be solved by considering all rules that has accuracy greater than some user predefined threshold. Table 5.1 shows the predicted labels for "Emotions" dataset using all association rules that have accuracy greater than (0.40).

Table 5.1 disjunction case for "emotions" dataset

amazed	happy	relax	quite	sad	angry	Least	Predicted Labels
0	0	1	1	1	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
1	0	0	0	0	1	amazed	amazedangry
0	0	0	0	1	0	sad	sadrelax
0	1	1	0	0	0	happy	happyrelax
0	0	0	0	1	0	sad	sadrelax
0	0	1	0	1	0	sad	sadrelax
1	0	0	0	0	1	amazed	amazedangry
0	0	1	1	1	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
0	0	0	0	1	1	sad	sadrelax
0	0	1	1	1	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
0	1	1	1	0	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
0	0	1	1	1	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
1	1	0	0	0	0	happy	happyrelax
1	0	0	0	1	1	sad	sadrelax
1	0	0	0	0	1	amazed	amazedangry
0	0	1	1	1	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
0	0	1	0	0	0	relax	relax
0	0	0	1	1	0	quite	{Quite,Sad} or {Quite,Relaxing} or {Quite,Happy}
1	1	0	0	0	0	happy	happyrelax

5.2.3 Enhancement of LP using correlations among labels

In this section, we are proposing an idea to convert multi-label classification problem to multi-class problem classification, and then use an algorithm such as MMAC to produce the classifier. The idea is based on using correlations among labels to find frequent label set and then transform multi-label data into single class problem which reflects composite label class as in table 5.2.

Table 5.2 Enhancement of LP using correlations among labels

Amazed	Happy	Relaxing	Quite	Sad	Angry	Class
0	0	1	1	1	0	{Quite-Sad}
1	0	0	0	0	1	{Amazed-Angry}
0	0	0	0	1	0	{Sad-Relaxing}
0	1	1	0	0	0	{Happy-Relaxing}

Table 5.3 shows some statistics about frequent labels sets and its frequency in the "Emotions" dataset.

Table 5.3 statistics about frequent labels sets and its frequency in the "Emotions" dataset.

Label Set	Frequency Percentage
{ Amazed ,Angry }	15%
{ Quite ,Sad }	18%
{ Quite ,Relaxing }	16%
{Sad , Relaxing }	16%
{ Relax , Quite , Sad}	13%

References

1. Agrawal, R., Amielinski, T., and Swami, A. (1993) Mining association rule between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data,* (pp. 207-216). Washington, DC.

2. Boutell, M., Shen, X., Luo, J. & Brown, C, Multi-label semantic scene classification. Technical Report 813, Department of Computer Science, University of Rochester, NY and Electronic Imaging Products R & D, Eastern Kodak Company, 2003.

3. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37 (2004) 1757–1771

4. Cendrowska, J. (1987) PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(1987): 349-370.

5. Clare, A., King, R.: Knowledge discovery in multi-label phenotype data. In: Proceedings of the5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany (2001) 42–53

6. Cohen, W. (1995) Fast effective rule induction. *Proceedings of the 12th International Conference on Machine Learning*, (pp. 115-123). CA, USA.

7. De Carvalho, A.C.P.L.F. and Freitas, A.A. A tutorial on multi-label classification techniques. In Abraham,A., Hassanien, A. E., and Snael, V. (eds.),Foundations of Computational Intelligence, Vol. 5, Springer, September 2009 pp. 177-195.

8. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005), Volos, Greece (2005) 448–456

9. J. F¨urnkranz and E. H¨ullermeier. Pairwise preference learning and ranking. In N. Lavra˘c,D. Gamberger, H. Blockeel, and L. Todorovski, editors, Proceedings of the 14th EuropeanConference on Machine Learning (ECML-03), volume 2837 of Lecture Notes in Artificial Intelligence, pages 145–156, Cavtat, Croatia, 2003. Springer-Verlag.

10. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the 2005 ACM Conference on Information and Knowledge Management (CIKM '05), Bremen,Germany (2005) 195–200

11. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004). (2004) 22–30

12. Hilage, T., Kulkarni, R, Review of Literature on Data Mining, IJRRAS Journal Volume 10, 107-114, .(2012).

13. Jabez Christopher J , A Statistical Approach for Associative Classification , European Journal of Scientific Research ISSN 1450-216X Vol.58 No.2 (2011), pp.140-147

14. J.Arunadevi and Dr.V.Rajamani , An Evolutionary Multi Label Classification using Associative Rule Mining for Spatial Preferences . IJCA Special Issue on "Artificial Intelligence Techniques – novel Approach and practical Applications " AIT , 2011

15. McCallum, A, Multi-label text classification with a mixture model trained by EM .Proceeding of the AAAI'99 Workshop on Text Learning, 1999.

16. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining,1st Edition, Addison-Wesley,2005

17. Purvi Prajapati , Amit Thakkar , Amit Ganatra , A Survey and Current Research Challenges in Multi-Label Classification Methods , International Journal of Soft Computing & Engineering ISSN 2231-2307 Volume: 2; Issue: 1; Start page: 248; Date: 2012.

18. Read, Jesse, Pfahringer, Bernhard, Holmes, Geo_rey,and Frank, Eibe. Classifier chains for multi-label classification. In Proceedings of the European Con-ference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp 254-269, 2009.

19. Read, Jesse, Bernhard Pfahringer, Geoff Holmes, Multi-label Classification using Ensembles of Pruned Sets**,** Eighth IEEE International Conference on Data Mining,2008

20. Thabtah, F., Cowling, P. & Peng, Y. MMAC: A new multi-class, multi-label associative classification approach. In Proceedings of the 4th IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, pp. 217–224, 2004.

21. Thabtah, Fadi Abdeljaber, A review of associative classification mining. Knowledge Engineering Review, 22 (1). pp. 37-65. ISSN 0269-8889, (2007).

22. Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas,I. Multilabel classification of music into emotions.In Proceedings of the 9th International Conference on Music Information Retrieval, 2008.

23. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3 (2007) 1–13

24. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: Proceedings of the 18th European Conference on Machine Learning (ECML 2007), Warsaw, Poland (2007) 406–417

25. Vens, C., Struyf, J., Schietgat, L., Dˇzeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73 (2008) 185–214

26. Sang-hyeun Park , Johannes Fürnkranz , Multi-Label Classification with Label Constraints , In Technical Report TUD-KE-2008-04 (2008)

27. Schapire, R.E. Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning 39 (2000) 135–168

28. Scheffer, Tobias, Finding Association Rules that Trade Support Optimally Against Confidence, Proc of the 5th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'01), pp. 424-435. Freiburg, Germany: Springer-Verlag.2001

29. Sorower, Mohammad S. [Ph. D Qualifying Review Paper] A Literature Survey on Algorithms for Multi-label Learning. Corvallis, OR, Oregon State University. December 2010.Major Professor: Thomas G. Dietterich, Ph.D, Computer Science, Oregon State University.

30. Witten, I. H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, second edition, Elsevier: San Francisco, ISBN 0-12-088407-0

31. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40 (2007) 2038–2048

Appendix

Number	Feature Name in "Emotion" Dataset	Feature Symbol
1	Mean_Acc1298_Mean_Mem40_Centroid	A
2	Mean_Acc1298_Mean_Mem40_Rolloff	B
3	Mean_Acc1298_Mean_Mem40_Flux	C
4	Mean_Acc1298_Mean_Mem40_MFCC_0	D
5	Mean_Acc1298_Mean_Mem40_MFCC_1	E
6	Mean_Acc1298_Mean_Mem40_MFCC_2	F
7	Mean_Acc1298_Mean_Mem40_MFCC_3	G
8	Mean_Acc1298_Mean_Mem40_MFCC_4	H
9	Mean_Acc1298_Mean_Mem40_MFCC_5	I
10	Mean_Acc1298_Mean_Mem40_MFCC_6	J
11	Mean_Acc1298_Mean_Mem40_MFCC_7	K
12	Mean_Acc1298_Mean_Mem40_MFCC_8	L
13	Mean_Acc1298_Mean_Mem40_MFCC_9	M
14	Mean_Acc1298_Mean_Mem40_MFCC_10	N
15	Mean_Acc1298_Mean_Mem40_MFCC_11	O
16	Mean_Acc1298_Mean_Mem40_MFCC_12	P
17	Mean_Acc1298_Std_Mem40_Centroid	Q
18	Mean_Acc1298_Std_Mem40_Rolloff	R
19	Mean_Acc1298_Std_Mem40_Flux	S
20	Mean_Acc1298_Std_Mem40_MFCC_0	T
21	Mean_Acc1298_Std_Mem40_MFCC_1	U
22	Mean_Acc1298_Std_Mem40_MFCC_2	V

23	Mean_Acc1298_Std_Mem40_MFCC_3	W
24	Mean_Acc1298_Std_Mem40_MFCC_4	X
25	Mean_Acc1298_Std_Mem40_MFCC_5	Y
26	Mean_Acc1298_Std_Mem40_MFCC_6	Z
27	Mean_Acc1298_Std_Mem40_MFCC_7	AA
28	Mean_Acc1298_Std_Mem40_MFCC_8	AB
29	Mean_Acc1298_Std_Mem40_MFCC_9	AC
30	Mean_Acc1298_Std_Mem40_MFCC_10	AD
31	Mean_Acc1298_Std_Mem40_MFCC_11	AE
32	Mean_Acc1298_Std_Mem40_MFCC_12	AF
33	Std_Acc1298_Mean_Mem40_Centroid	AG
34	Std_Acc1298_Mean_Mem40_Rolloff	AH
35	Std_Acc1298_Mean_Mem40_Flux	AI
36	Std_Acc1298_Mean_Mem40_MFCC_0	AJ
37	Std_Acc1298_Mean_Mem40_MFCC_1	AK
38	Std_Acc1298_Mean_Mem40_MFCC_2	AL
39	Std_Acc1298_Mean_Mem40_MFCC_3	AM
40	Std_Acc1298_Mean_Mem40_MFCC_4	AN
41	Std_Acc1298_Mean_Mem40_MFCC_5	AO
42	Std_Acc1298_Mean_Mem40_MFCC_6	AP
43	Std_Acc1298_Mean_Mem40_MFCC_7	AQ
44	Std_Acc1298_Mean_Mem40_MFCC_8	AR
45	Std_Acc1298_Mean_Mem40_MFCC_9	AS
46	Std_Acc1298_Mean_Mem40_MFCC_10	AT
47	Std_Acc1298_Mean_Mem40_MFCC_11	AU

48	Std_Acc1298_Mean_Mem40_MFCC_12	AV
49	Std_Acc1298_Std_Mem40_Centroid	AW
50	Std_Acc1298_Std_Mem40_Rolloff	AX
51	Std_Acc1298_Std_Mem40_Flux	AY
52	Std_Acc1298_Std_Mem40_MFCC_0	AZ
53	Std_Acc1298_Std_Mem40_MFCC_1	BA
54	Std_Acc1298_Std_Mem40_MFCC_2	BB
55	Std_Acc1298_Std_Mem40_MFCC_3	BC
56	Std_Acc1298_Std_Mem40_MFCC_4	BD
57	Std_Acc1298_Std_Mem40_MFCC_5	BE
58	Std_Acc1298_Std_Mem40_MFCC_6	BF
59	Std_Acc1298_Std_Mem40_MFCC_7	BG
60	Std_Acc1298_Std_Mem40_MFCC_8	BH
61	Std_Acc1298_Std_Mem40_MFCC_9	BI
62	Std_Acc1298_Std_Mem40_MFCC_10	BJ
63	Std_Acc1298_Std_Mem40_MFCC_11	BK
64	Std_Acc1298_Std_Mem40_MFCC_12	BL
65	BH_LowPeakAmp	BM
66	BH_LowPeakBPM	BN
67	BH_HighPeakAmp	BO
68	BH_HighPeakBPM	BP
69	BH_HighLowRatio	BQ
70	BHSUM1	BR