Multi-Label Learning with Feature Selection for Video Classification

by

Jan André Marais

Thesis presented in partial fulfilment of the requirements for the degree of Master of Commerce (Mathematical Statistics) in the Faculty of Economic and Management Sciences at Stellenbosch University

Supervisor: Dr. S. Bierman

December 2017

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: . . . .

Abstract

Multi-Label Learning with Feature Selection for Video Classification

J. A. Marais

Thesis: MCom (Mathematical Statistics)
December 2017

English abstract

Uittreksel

Multi-Etiket leer met Veranderlikeseleksie vir Videoklassifikasie

("Multi-Label Learning with Feature Selection for Video Classification")

J. A. Marais

Tesis: MCom (Wiskundige Statistiek)
Desember 2017

Afrikaans abstract

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

Contents

Declaration **i**

Abstract **ii**

Uittreksel **iii**

Acknowledgements **iv**

Contents **v**

List of Figures **viii**

List of Tables **ix**

Nomenclature **x**

1 Introduction **1**

1.1 Motivation and Thesis Objectives 1

1.1.1 Motivation 1

1.1.2 Thesis Objectives 1

1.2 Data, Code and Reproducibility 1

1.3 Important Concepts and Terminology 1

1.3.1 Supervised Learning 2

1.4 Outline 2

2 Multi-Label Learning **3**

2.1 Introduction 3

2.1.1 Basic Idea 6

2.1.2 Applications 6

2.1.3 In the Literature 6

2.1.4 Challenges and Ongoing Research 6

2.1.5 Aim of this Chapter 7

2.2 The Paradigm 7

2.2.1 Learning Framework 7

2.2.2 Multi-Label Data and Repositories 8

2.2.3 Key Challenge . . . 8
2.2.4 Threshold Calibration . . . 9
2.3 Evaluation Metrics . . . 9
2.3.1 Brief Taxonomy . . . 9
2.3.2 Example-based Metrics . . . 10
2.3.3 Label-based Metrics . . . 11
2.3.4 Theoretical Results . . . 13
2.4 Learning Algorithms . . . 13
2.4.1 Simple Catergorisation . . . 13
2.4.2 Problem Transformation Methods . . . 14
2.4.3 Algorithm Adaption Methods . . . 20
2.4.4 Summary . . . 20
2.5 Related Learning Strategies . . . 21
2.6 Pitfalls and Guidelines . . . 21
2.7 Conclusion . . . 21

3 Label Dependence 23
3.1 My thoughts (remove later) . . . 23
3.2 Introduction . . . 23
3.3 Two types of label dependence . . . 24
3.3.1 The task of multi-label classification . . . 25
3.3.2 Marginal vs. conditional dependence . . . 25
3.4 Link between label dependence and loss minimization . . . 29
3.4.1 The individual label view . . . 29
3.4.2 The joint label view . . . 30
3.4.3 The joint distribution view . . . 31
3.5 Previous attempts to ‘exploit’ label dependence . . . 32
3.6 Improved attempts to ‘exploit’ label dependence . . . 32
3.7 Empirical Ideas . . . 40
3.8 Introduction . . . 40
3.9 Exploiting Label Dependence . . . 41
3.10 Theoretical Results . . . 41
3.10.1 Intuitive Perspective . . . 41
3.10.2 Two Types of Label Dependence . . . 41
3.10.3 Link with Loss Function . . . 41
3.10.4 Symmetry . . . 41
3.10.5 Locality . . . 41
3.11 Empirical Analysis . . . 41
3.11.1 Previous Findings . . . 42
3.11.2 Simulation Study . . . 42
3.12 Conclusion . . . 42

4 Input Space Reduction 43
4.1 Introduction . . . 43

4.1.1 Single-Label Framework 43
4.2 Feature Selection 43
4.3 Feature Extraction 43
4.4 Meta-analysis 43
4.5 Summary 43

5 Output Space Reduction 44
5.1 Introduction 44

6 Video Tagging 45
6.1 Introduction 45
6.2 General Approaches 45
6.3 Mutli-Label Tagging 45
6.4 Introduction 45
6.5 Definition 45
6.6 Existing datasets 46
6.7 Other approaches 46
6.8 Scalabiltiy 46
6.8.1 Active/Online Learning 46
6.8.2 Output reduction 46

7 YouTube-8m Challenge 47
7.1 Describe the challenge 47
7.2 Results 47

8 Conclusion 48

Appendices 49

A Benchmark Datasets 50

B Software 51

Bibliography 52

List of Figures

2.1 Line graph per database of the number of relevant ML documents released per year for. 5
2.2 Categorisation of the taxonomy of MLL evaluation metrics 10
2.3 Categorisation of multi-label learning taxonomy (this is just an example) 14

List of Tables

Nomenclature

Constants

g =	$9.81\,\mathrm{m/s^2}$	

Variables

Re_D	Reynolds number (diameter)	[ ]
x	Coordinate	[m]
$\ddot{x}$	Acceleration	[$\mathrm{m/s^2}$]
θ	Rotation angle	[rad]
τ	Moment	[N·m]

Vectors and Tensors

$\vec{\boldsymbol{v}}$	Physical vector, see equation ...

Subscripts

a	Adiabatic
a	Coordinate

Chapter 1

Introduction

1.1 Motivation and Thesis Objectives

give short aim of the study

1.1.1 Motivation

desribe why this thesis is created. Mention things like relevance of field / wide applicability, lack of statistical perspective, lack of recent review, unsolved problems, YouTube Challenge.

1.1.2 Thesis Objectives

What I want to achieve with this thesis/ readers to take away. Mention things like: understanding the basics and scope of MLC, being able to apply it to large problems.

maybe say something of generating ML data and comment on ML data repositories.

maybe comment on statistical learning vs machine learning.

1.2 Data, Code and Reproducibility

Got this header from Arnu's thesis - not sure if I will include this. But it may be appropriate to indicate here where to find the code for the thesis, etc.

1.3 Important Concepts and Terminology

Briefly introduce the important concepts to be grasped in order to follow the main thread of the thesis. It seems reasonable to introduce the problem of supervised learning here. The rest still needs to be decided on.

1.3.1 Supervised Learning

1.4 Outline

Give an overview of the outline of the thesis, which is (thus far): review of current ML literature, investigation of label dependence, ways of selecting features for ML problems, considerations for extreme MLC, introduction to video classification and tagging and finally the YT8M challenge.

Chapter 2

Multi-Label Learning

2.1 Introduction

- why this chapter
- aim is to cover the core of MLL literature
- update of previous reviews
- to highlight areas to focus on for further studies

Multi-label (ML) learning belongs to the supervised learning paradigm and can be viewed as a generalisation of the traditional single-label learning problem. Suppose the dataset to be analysed consists of a set of observations each representing a real-world object such as an image or a text document. In the single-label context each object is restricted to belonging to a single, mutually exclusive class, *i.e.* each observation is associated with a single label. One can quite effortlessly come up with tasks that will not fit into this framework: an image annotation problem where each image contains more than one semantic object, a text classification task where each document has multiple topics or an acoustic classification task where the recordings contain the sounds of multiple bird species. Therefore the need for a ML learner that can assign a set of labels to an observation. Let $L = \{l_1, l_2, \ldots, l_K\}$ denote the complete set of possible labels that can be assigned to an observation. Whereas a single-label learner aims to find which single label l_k, $k = 1, 2, \ldots, K$, belongs to a given observation, a ML learner is capable of assigning a set of labels $l \subseteq L$ to the observation.

According to (Zhang and Zhou, 2014), ML learning can be considered a sub-problem of a wider framework, called multi-target learning, covering all problems where an observation is associated with multiple outputs. When the output variables are binary, it is a ML learning problem. But problems also exist where the output variables are multi-class or numerical and in these settings the problems are respectively known as multi-dimensional learning and multi-output regression. It is also possible that the output variables are combinations of the aforementioned types.

cannot find good literature on multi-target learning - might not want to add the above paragraph

ML learning is still an emerging research area of statistical learning. Figure 2.1 illustrates the rise of ML learning research. A search (24-03-2017) for ML literature was done on two bibliographic databases - Scopus and Semantic Scholar. Details of the search strategies can be found at (appendix ??). The birth of the ML field (around 1999) came from the need to assign multiple labels to text documents. Contributions in (Schapire and Singer, 1999) and (Schapire and Singer, 2000) adapted a boosting algorithm to handle ML data. (Elisseeff and Weston, 2001) defined a ranking based SVM to deal with ML problems in the areas of text mining and also bioinformatics. (Lewis *et al.*, 2004) released an important benchmark collection for ML text classification. Another highly cited ML SVM implementation is (Boutell *et al.*, 2004), with application in scene/image classification. (Zhang *et al.*, 2006) showed how to apply neural networks to a ML problem and (Zhang and Zhou, 2007) adapted the KNN algorithm for ML input. The first overview on the subject was given in (Tsoumakas and Katakis, 2007*b*) where the author discussed the most relevant ML learning approaches. Then came applications to music, (Trohidis and Kalliris, 2008) and (Turnbull *et al.*, 2008). (Vens *et al.*, 2008) showed how to use decision trees for hierarchical ML classification. Important papers introducing unique ML approaches are (Tsoumakas and Vlahavas, 2007), (Fürnkranz *et al.*, 2008) and [Read2011a]. A crucial step for ML learning was to make it accesible and useable to more reasearchers. The authors of (Tsoumakas *et al.*, 2011) developed a Java library for ML learning. Later on (Madjarov *et al.*, 2012) did a empirical study on the most important ML algorithms up to that date, comparing 12 multi-label learning methods using 16 evaluation measures over 11 benchmark datasets. More recent extensive reviews of ML learning are given in (Zhang and Zhou, 2014) and (Gibaja and Ventura, 2014).

not sure which papers to include here. The aim is to highlight the most important. Difficult to determine. Should I visualise timeline of contributions?

what does recent research focus on? Extreme ML learning?

The rapid growth of the ML learning is probably owed to the vast and expanding range of ML application domains, the biggest being text and multimedia categorisation especially those generated and/or stored on the web. Other application domains common to ML learning are: biology, chemical data analysis, social network mining and E-learning amongst others. A thorough list of applications and their citations can be found in (Gibaja and Ventura, 2014).

Say something of video classification? Create my own list of applications?

The searches were not identical since they were limited to the search features of the databases. On Scopus the search was for all documents (conference papers, articles, conference, articles in press, reviews, book chapters and books) in any subject area with either the words *multi-label* or *multilabel* and either the words *learning* or *classification* found in either their titles, abstracts or keywords. The Semantic Scholar search is based on machine learning principles and thus automatically decides which research documents are relevant to a specific search query. The query used was *multilabel multi-label learning classification.* The search only returns research in the computer science and neuroscience fields of study. More technical details can be found on the respective engine's websites.

probably need to reference the engines

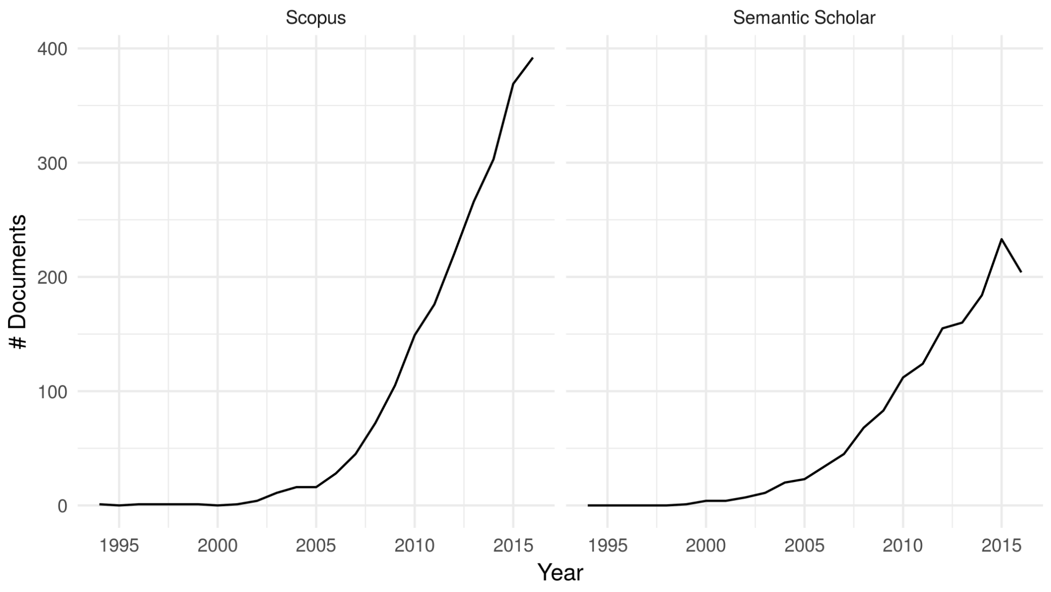

Figure 2.1: Line graph per database of the number of relevant ML documents released per year for.

The key challenge in ML learning is to exploit dependecies amongst labels, *e.g.* using the information on the relevance of label l_i to predict label l_j, $i, j \in \{1, 2, \ldots, K\}$, $i \neq j$. This is especially difficult for a ML learner when there are many labels. It is not uncommon for ML datasets to have hundreds of thousands of labels. Proof of this can be found at this ML data repository. Algorithms that can accurately and efficiently model label dependence on these datasets are rare (Sorower, 2010). This is a focus area of recent ML research, called extreme multi-label learning (Xu *et al.*, 2016). A more formal definition of label dependence will be given later on. An in-depth discussion on the unique challenges (thorough list by (Gibaja and Ventura, 2014)) that arise from dealing with label dependence and some of the possible strategies to follow will also be covered.

It is important to note that we will define the utlimate task of ML learning as assigning of multiple labels to an observation. ML learning covers to very similar approaches, namely, ML classification and ML ranking. ML classification algorithms output whether or not labels are relevant to an observation (binary) and ML ranking algorithms outputs a real-valued score assigned to each label indicating its relative importance to an observation. Thus with ML ranking, for each observation we seek a list of labels ordered by their scores representing the confidence in how relevant they are to the specific observation. Many classifiers base their final (categorical) prediction on the thresholding of the real-valued output of the algorithm and thus can also be used for ranking. Similarly, ranking algorithms can also be used for classification if a thresholding function is applied to the real-valued output.

2.1.1 Basic Idea

2.1.2 Applications

- thorough list by (Gibaja and Ventura, 2014)
- text; multimedia; biology; chemical data analysis; social network mining; e-learning; other
- examples and citations
- mention importance

2.1.3 In the Literature

- history
- database analysis
- show trend
- most cited papers
- mention important reviews:

(Tsoumakas and Katakis, 2007*a*), (Sorower, 2010), (Zhang and Zhou, 2014), (Gibaja and Ventura, 2014), Tutorial: (Carvalho, André C P L F de, 2009), (Gibaja and Ventura, 2015)

- see (Gibaja and Ventura, 2015) for different MLL resources

2.1.4 Challenges and Ongoing Research

- key challenge = output space

- mention challenges casused by key challenges
- from (Gibaja and Ventura, 2014)
- worth highlighting that recent research has shifted its focus to problems on large scale labels - focus on scalability
- Label Dependence: conditional vs marginal; mentions order categorisation; asymetric and local
- Dimensionality Reduction
- Reduction of the Input Space: feature selection; feature extraction
- Reduction of the Output Space: HOMER; Label reduction with association rules; CCA; compressed sensing; principal label space transform; compressed labelling
- MIML
- Semi-supervised and Active Learning
- Online MLL and Data Streams
- Hierachical MLC
- Dealing with Class Imbalance: label skew
- Mention extreme MLC and maybe add a section somewhere in chapter
- list also in (Alazaidah and Ahmad, 2016)

2.1.5 Aim of this Chapter

- the goal of this chapter: review + highlight limitations to be studied further
- outline

2.2 The Paradigm

- consider adding the settings described by Read and (Gibaja and Ventura, 2015)

2.2.1 Learning Framework

- mathematical definition with notation of the task of ML learning.
- real-valued output + thresholding function (ranking vs classification)

The following notation will be used throughout the thesis. Define the input matrix as

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^\intercal & \boldsymbol{x}_2^\intercal & \dots & \boldsymbol{x}_n^\intercal \end{bmatrix},$$

where n is the number of observations and p is the number of features. $\boldsymbol{x}_i^\intercal$ represents the p-dimensional vector that forms the i-th row of X. For a text classification problem, x_{ij} might indicate the number of times a word j appeared in document i. Define the label (/output) matrix as

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1K} \\ y_{21} & y_{22} & \cdots & y_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nK} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_1^\intercal & \boldsymbol{y}_2^\intercal & \ldots & \boldsymbol{y}_K^\intercal \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{(1)} & \boldsymbol{Y}_{(2)} & \ldots & \boldsymbol{Y}_{(K)} \end{bmatrix},$$

where K is the size of the labelset L. Y only contains zeros and ones, i.e., $y_{ik} = 1$ if label l_k, $k = 1, \ldots, K$, is present for observation i and $y_{ik} = 0$ if it is absent. Thus $\boldsymbol{Y}_{(k)}$ is a n-dimensional binary vector indicating which observations are associated with label l_k.

A ML dataset will be defined as $D = \begin{bmatrix} X & Y \end{bmatrix}$, which contains the n input-output pairs, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \,|i = 1, \ldots, n\}$. The task of ML classification is to find a function h that accurately maps the observations contained in X to the label matrix Y, i.e., $h : X \rightarrow Y$, so that given a new observation, h can determine which labels belong to it. The accuracy aforementioned is a topic that will be discussed shortly. The measurement thereof is another unique problem for ML classification.

On the other hand, the goal of ML ranking is to find a function $f : X \rightarrow G$, where G is a similar matrix to Y, but with the g_{ij} a real value representing the relative confidence score that label j is relevant to observation i. f is found by optimising a ranking metric, also discussed shortly. From the confidence scores of observation i, $f(\boldsymbol{x}_i)$, a ranking $\boldsymbol{r}_i$ can be obtained, giving the rank of labels in descending order of $f(\boldsymbol{x}_i)$.

mention the calibration factor of (Zhang and Zhou, 2014). Finding z_i from r_i

h will be referred to as the ML classifier and f as the ML ranker. When ML learner will be a collective term covering both h and f. Before different ML learners can be discussed, an understanding of how the output of these algorithms are evaluated is necessary, since fitting f of h envolves optimising an evaluation metric. (always?)

2.2.2 Multi-Label Data and Repositories

- properties of a ML data set: label cardinality, label density, label diversity
- Simulating
- partitioning mentioned in (Gibaja and Ventura, 2015) - referred to (Sechidis *et al.*, 2011)
- (Sorower, 2010)
- details of benchmark datasets in appendix (learn how to link)

2.2.3 Key Challenge

- key challenge is the size of the output space: label sets grows exponentially with increase in labels
- solution is to exploit label correlations efficiently
- (Sorower, 2010) has a section on exploiting label dependence
- mention label dependence chapter
- first-order, second-order, high-order strategies grouping

2.2.4 Threshold Calibration

- calibrate real-valued output against thresholding function output in order to determine labels of unseen instances.
- constant vs induced from training data + ad hoc specific to certain learning algorithms
- alternative to choose top k labels

2.3 Evaluation Metrics

The evaluation of the performance of ML algorithms is another distinct problem to this setting. Compared to the single-label case, many more evaluation metrics exist, with subtle or obvious differences in their measurement. According to (Madjarov *et al.*, 2012) it is essential to evaluate a ML algorithm on multiple and contrasting measures because of the additional degrees of freedom introduced by the ML setting. In addition, care should be taken when reporting multiple measures and with their interpretation. Since some of the measures are contrasting it is dangerous to report multiple metrics and conclude that on average one learner is better than the other. This was highlighted in (Dembcz *et al.*, 2012), where the authors suggested that when evaluating the performance of a ML learner, it should be made clear which metric(s) it is aiming to optimise, otherwise the results can be misleading. It is impossible (?) for a learner to have superior performance over others in terms of all the multi-label evaluation metrics simultaneously.

The evaluation measures of predictive performance of multi-label learnerss can be divided into two groups: example-based and label-based measures. Example-based measures compares the actual versus the predicted labels for each observation and then computes the average across all the observations in the dataset. Where label-based measures computes the predictive performance on each label separately and then averages across all labels (Madjarov *et al.*, 2012). For both groups the measures can further be partitioned into metrics from a classification persepective and measures from a ranking perspective, *i.e.* metrics for h and metrics for f respectively. The most commonly used metrics in each of the groups will be introduced here.

2.3.1 Brief Taxonomy

- more complicated than single label metrics
- introduce example based vs label based
- for classification and ranking
- diagram / table + where they are used

```
grViz('figures/eval-tax.gv') %>% export_svg() %>%
  charToRaw %>% rsvg %>% png::writePNG('figures/eval-tax.png')
```

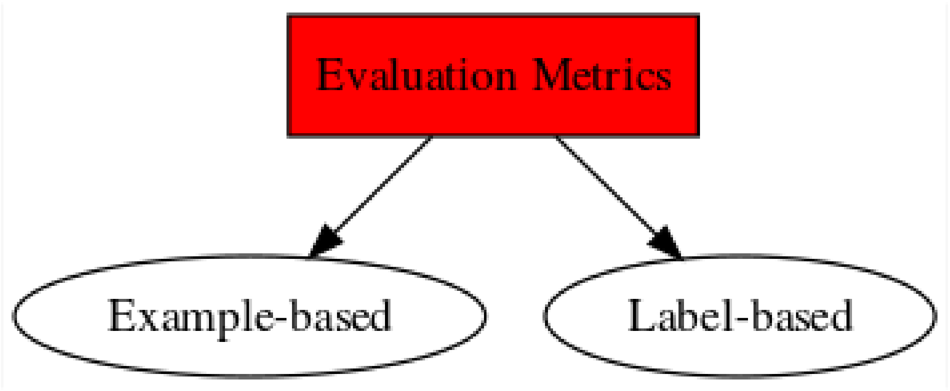

Figure 2.2: Categorisation of the taxonomy of MLL evaluation metrics

- Figure 2.2 is just an example. The image quality is lacking.

2.3.2 Example-based Metrics

- subset accuracy; hamming loss; accuracy; precision; recall; one-error; coverage; ranking loss; average precision
- definition + brief interpretaion where it is unclear

For the following definitions, let y_i be the set of true labels for observation $\boldsymbol{x}_i$ and z_i the set of predicted labels for the same observation, obtained from the predicted indicator vector of $\hat{h}(\boldsymbol{x}_i)$. The Hamming loss is then defined as

$$\text{hloss}(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{K}|z_i \triangle y_i|,$$

where $\triangle$ stands for the symmetric difference and $|.|$, the size of the set. For example, $|\{1,2,3\}\triangle\{3,4\}| = |\{1,2,4\}| = 3$. Thus the Hamming loss counts the number of labels not in the intersection of the predicted subset of labels and the true subset of labels, as a fraction of the total size of the labelset, averaged across each observation in the dataset. When h returns perfect predictions for each observation in the dataset, $\text{hloss}(h) = 0$, and if h predicts for each observation that it belongs to all the labels except for its the true labels, $\text{hloss}(h) = 1$.

Accuracy is defined as

$$\text{accuracy}(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{|z_i \cap y_i|}{|z_i \cup y_i|}.$$

Thus for each observation the number of correctly predicted labels is calculated as a proportion of the sum of the correctly and incorrectly predicted labels. These quantities are then averaged over each observation in the dataset. If the h perfectly predicts the relevant subset of labels for each observations, $\text{accuracy}(h) = 1$. If h does not manage to predict a single correct label for any observation, $\text{accuracy}(h) = 0$.

The precision and recall are respectively defined as

$$\text{precision}(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{|z_i \cap y_i|}{|z_i|},$$

and

$$\text{recall}(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{|z_i \cap y_i|}{|y_i|}.$$

Precision calculates the average proportion of correctly predicted labels in terms of the number of labels predicted, across all the observations in the dataset. Recall calculates a similar average, with the only difference that the proportion is calculated in terms of the number of true labels per observation. Both these metrics lie in the range $[0, 1]$ with larger values desirable.

The harmonic mean between the precision and the recall is called the F_1-score and is defined as

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|z_i \cap y_i|}{|z_i| + |y_i|}.$$

The perfect classifier will result in a F_1-score of 1 and the worst possible score is zero.

The subset accuracy or classification accuracy is defined as

$$\text{subsetacc}(h) = \frac{1}{n}\sum_{i=1}^{n} I(z_i = y_i),$$

where $I(.)$ is the indicator function. This the subset accuracy is the proportion of observations that were perfectly predicted by h.

The above are all performance measures of ML classifiers. If the ML learner outputs real-valued confidence scores, these ranking metrics can be used to evaluate the learner's performance:

One-error:
Coverage:
Ranking Loss:
Average Precision:

2.3.3 Label-based Metrics

- micro vs macro ito tp, tn, fp, fn
- auc example

The idea with label-based measures is to compute a single-lable metric for each label based on the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) made by the classifier on a dataset and then obtaining an average of the values (Gibaja and Ventura, 2014). Note, TN_k, TP_k, FN_k and FP_k denote the quantities for label l_k, $k = 1, 2, \ldots, K$. Thus $TP_k + TN_k + FP_k + FN_k = n$. Let B be any binary classification metric, i.e. $B \in \{\text{accuracy}, \text{precision}, \text{recall}, F_1\}$. B can be written in terms of TN_k, TP_k, FN_k and FP_k, for example

$$\text{accuracy}(TN_k, TP_k, FN_k, FP_k) = \frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k}.$$

B is then calculated for each label and then an average is calulated. The averaging can be done either by the micro or the macro approach. The micro approach considers predictions of all observations together and then calculates the measure across all labels, i.e.

$$B_{micro} = B\left(\sum_{k=1}^{K} TP_k, \sum_{k=1}^{K} TN_k, \sum_{k=1}^{K} FP_k, \sum_{k=1}^{K} FN_k\right).$$

Whereas the macro approach computes one metric for each label and then the values are averaged over all the labels, i.e.

$$B_{macro} = \frac{1}{K}\sum_{k=1}^{K} B(TP_k, TN_k, FP_k, FN_k).$$

Note, also that $\text{accuracy}_{micro}(h) = \text{accuracy}_{macro}(h)$ and that $\text{accuracy}_{micro}(h) + \text{hloss}(h) = 1$, since Hamming loss is the average binary classification error.

Again, all of the above mentioned metrics are from a classification perspective. An example of a label-based metric from a ranking persepective is the macro- and micro-averaged AUC:

Most multi-label classifiers learn from the training observations by explicity or implicitly optimising one specific metric (Zhang and Zhou, 2014). That is why in (Dembcz *et al.*, 2012) the authors reccomended specifying which of the metrics a new proposed algorithm aims to optimise in order to show if it is succesful. But at the same time it is important to test the algorithm on numerous metrics for fair comparisons against other algorithms (Zhang and Zhou, 2014), (Madjarov *et al.*, 2012). It might be that a algorithm does very well in terms of the Hamming loss, but performs poorly according to the subset accuracy, or vice versa, as shown in (Dembcz *et al.*, 2012). In (Tsoumakas and Vlahavas) they claim that the Hamming loss reported together with the micro-average F-measure gives a good indication of the performance of a multi-label classifier.

These multi-label metrics are usually non-convex and discontinuous (Zhang and Zhou, 2014). Therefore multi-label classifiers resort to considering surrogate metrics which are easier to optimise.

probably should add an example or maybe later

Other than predictive performance, are there other aspects on which multi-label classifiers can be evaluated, such as efficiency and consistency. Multi-label algorithms should be efficient in the sense that it takes the least amount of computational power for a given level of predictive performance (Madjarov *et al.*, 2012). These classifiers can take a considerable amount of time to train when complicated ensembles are being implemented on datasets with huge labelsets. In cases where live updating and predictions are needed, this may be a problem [reference]. The other desirable attribute of multi-label classifiers are that they are consistent. This means that the expected loss of the classifier converges to the Bayes loss when the number of observations in the training set tends to infinity. Actually only a very few number of multi-label classifiers satisfy this property (Gao and Zhou, 2011), (Koyejo *et al.*, 2015).

2.3.4 Theoretical Results

- evaluate perfomance on many metrics for fairness
- something on label dependence link that will be discussed in next chapter
- minimisation of surrogate loss functions and consistency
- consistency (Gao and Zhou, 2011):

They were the first to do a theoretical study on the consistency of multi-label learning algorithms, focusing on the ranking loss and the hamming loss. A learning algorithm is said to be consistent if its expected risk converges to the Bayes risk as the size of the training data increases. They found that any convex surrogate loss is inconsistent with the ranking loss and therefore proposed a partial ranking loss (which is consistent with some surrogate loss functions) as an alternative. They also show how some recent multi-label algorithms are inconsistent in terms of the hamming loss and provides a discussion on the consistency of approaches which transforms the multi-label problem into a set of binary classification tasks.

2.4 Learning Algorithms

2.4.1 Simple Catergorisation

There are numerous multi-label learning algorithms. It is difficult to keep up with the all the latest proposed methods. These algorithm can be categorised in a number of ways, *e.g.* the review (Zhang and Zhou, 2014) and the tutorials (Gibaja and Ventura, 2015) and (Carvalho, André C P L F de, 2009), all have different ways of grouping the algorithms. The categorisation for this thesis is chosen to satisfy the criteria of being common, simple and intuitive. Nevertheless, the characteristics of the algorithms leading to the other grouping variants will still be given in the remarks of the algorithms.

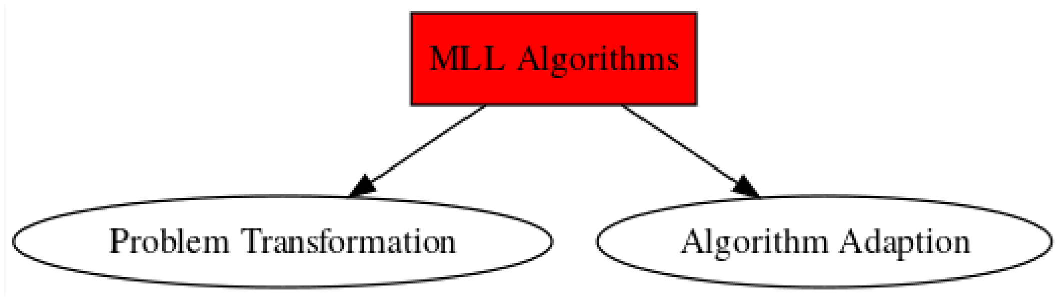

Figure 2.3: Categorisation of multi-label learning taxonomy (this is just an example)

- still need to edit Figure 2.3
- want to keep it simple and representative but also give table with full list of methods
- many proposals
- scrutinise 8 representative algorithms for feasibility concerns
- representativeness criteria: broad spectrum; primitive impact; favourable impact
- introduce PT vs AA
- diagram of categorisation
- very thorough one in (Gibaja and Ventura, 2015)
- mention ensemble category

2.4.2 Problem Transformation Methods

Problem transformation methods consist of first transforming the multi-label problem into one or more single-label problem(s) and then fitting any standard supervised learning algorithm(s) to the single-label data. For that reason, problem transformation methods are called algorithm independent, i.e. once the data is transformed, any single-label classifier can be used (Tsoumakas and Vlahavas).

The two main problem transformation algorithms are the binray relevance and label powerset transformations. Both methods suffer from several limitations but they form the basis of arguably any problem transformation method. The state-of-the-art problem transformations algorithms are most of the times extensions of either the standard binary relevance or label powerset algorithms (Alazaidah and Ahmad, 2016). Therefore the understanding of these two basic methods are crucial in dealing with the more complex, modern problem transformation methods.

2.4.2.1 Binary Relevance

- basic idea
- notation
- cross-training
- T-criterion for avoiding empty prediction
- psuedo-code
- remarks: first-order; parallel; straightforward; building block of state-of-the-art; ignores potential label correlations; may suffer from class-imbalance; computational complexity

The most common transformation method is binary relevance (BR). BR transforms the mutli-label into K single-label problems by modelling the presence of the labels separately. Typically K single-label binary data sets, $D_k = (X, \boldsymbol{Y}_k)$ for $k = 1, ..., K$, would be constructed from the multi-label data set, $D = (X, Y)$. To each D_k any single-label classifier can be applied. In the end, predictions $\hat{\boldsymbol{Y}}_1, ..., \hat{\boldsymbol{Y}}_K$ are obtained separately which can then be combined to allocate all the predicted relevant variables to each instance. Note, that it may occur that all of the single-label learners produces zeroes, which would imply that the instance belongs to an empty set. To avoid this (Zhang and Zhou, 2014) suggests following the T-criterion rule. The rule states, briefly, that in such a case the labels associated with the greatest output should be assigned to the instance. Clearly, this will only work if the base learners used gives continuous outputs and it will only make sense if all the base learners are of the same type. I suppose these rules are ad-hoc and I can think of alternatives.

> With this approach the standard single label feature selection procedures can be applied. The relevant subset of features can be identified for each label. This is convenient since it is not unlikely that the optimal subset of features will differ from label to label.

The biggest drawback for this approach is that it models each label separately and ignores the possible correlations between labels. Thus BR assumes that there are no correlations between the labels. However, these correlations can be very helpful in predicting the labels present. This is a first-order strategy. Also it can be time consuming since data sets with hundreds of labels is not rare. This would mean more than a hundred models should be fit and tuned separately. But this complexity scales linearly with increasing K, which is actually not so bad when comparing to other multi-label algorithms. Grouping the labels in a hierachical tree fashion may become useful when K is very large (Cherman *et al.*, 2011) (see also Incorporating label dependency into the binary relevance framework for multi-label classification by the same authors).

Another argument against BR from (Read *et al.*, 2011): The argument is that, due to this information loss, BR's predicted label sets are likely to contain either too many or too few labels, or labels that would never co-occur in practice.

Advantage of BR by (Read *et al.*, 2011): Its assumption of label independence makes it suited to contexts where new examples may not necessarily be relevant to any known labels or where label relationships may change over the test data; even the label set L may be altered dynamically - making BR ideal for active learning and data stream scenarios.

Nevertheless, BR remains a competitive ML algorithm in terms of efficiency and efficacy, especially when minimising a macro-average loss function is the goal (Luaces *et al.*). The most important advantage of BR is that it is able to optimise several loss functions (Luaces *et al.*) also see small proof. They also show empirically that BR tends to outperform ECC when there are many labels, high label dependency and high cardinality, i.e. when the multi-label data becomes more complicated.

Compared to label powerset (LP) which will be discussed later, BR is able to predict arbitrary combinations of labels (Tsoumakas *et al.*, 2009) not restricted only to those in the training set.

(Cherman *et al.*, 2011) also proposes a variation of BR called BR+. Its aim is to keep the simplicity of BR but also to consider the possible label correlations. It does so by also creating K binary data sets but this time each of these data sets treat all the label columns not to be predicted by the current single-label classifier as features to the classifier. Thus each sinlge-label classifier will have $p + K - 1$ inputs. So now when predicting label l, all of the original features in X and the remaining variables $\boldsymbol{Y}_k$, $k \neq l$, are used as inputs for classifier l. (second order strategy?)

The problem arises when predicting unseen instances for which the labels are unknown. Thus the input needed for each binary classifier is not available. One workaround is to obtain an initial prediction of the labels using an ordinary BR approach and then using these predictions as inputs to the BR+ algorithm. The BR+ algortihm will most likely produce different predictions to the initial predicitons or BR which can then also be used in a next round of BR+. These steps can be continued until convergence but this seems like the classifier chains approach. (to be investigated).

(Tsoumakas *et al.*, 2009) mentions the 2BR strategy that seems similar/identical to BR+. They describe the 2BR method as follows: first train a binary classifier on each of the K binary data sets and then use their predictions (and or probabilities) as so called meta-features for a second round of BR. They mention that it might be better to train the base and meta learners on separate parts of the training data to avoid biased predictions. They suggest using a cross-validation approach for both learners to also avoid size constraints of the training data. They describe this approach as a stacked generalisation, also mentioned in (Tsoumakas *et al.*, 2006), (Godbole and Sarawagi, 2004), (Pachet and Roy, 2009) calls it classifier fusion.

The adding of all the base learner predicitions as meta-feature to the meta-learners is not necessarily desirable. Some label pairs might have no correlation and adding predictions for those labels as inputs to the meta-learner will add noise to the model and waste computation time. (Tsoumakas *et al.*, 2009) suggests a solution called corerlation-based pruning. They calculate the pairwise correlations between labels, ϕ, and only add base learner prediction of label i as a meta-feature to meta-learner j if ϕ_{ij} is greater than some threshold. In this way only label-pairs that are highly correlated will be used in the final prediction of each other.

- BR performs well for Hamming loss, but fails for subset 0/1 loss.
- It is not clear, in general, whether the meta-classifier b should be trained on the BR predictions h(x) alone or use the original features x as additional inputs. Another question concerns the type of information provided by the BR predictions. One can use binary predictions, but also values of scoring functions or probabilities, if such outputs are delivered by the classifier @(Dembcz *et al.*, 2012).

2.4.2.2 Classifier Chains

- basic idea
- notation
- importance of ordering
- ECC brief explanation
- psuedo-code

- remarks: high-order; considers label correlations in a random manner; not parallel; computational complexity

Another extension of BR, similar to 2BR and BR+, is the classifier chains (CC) approach introduced by (Read *et al.*, 2011). It also consists of transforming the mutli-label data set D to K single-label data sets but the transformations are done sequentially in the sense that the label previously treated as a response will be added as a feature for predicting the next label. This will give data sets similar to $D_1 = (X, \boldsymbol{Y}_1), D_2 = (X, \boldsymbol{Y}_1, \boldsymbol{Y}_2), ...D_K = (X, \boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_K)$, where the last column of each is the response that needs to be predicted. To each of these single-label data sets a classifier can be trained and then their predictions are combined in the same fashion as BR. CC keeps the simplicity of BR but has that additional capacity to model label dependencies by passing label information between classifiers. This should raise the question of what order of labels should the chain consist of and should it stop after one cycle?

In a response to this, the ensembles of classifier chains (ECC) was suggested by (Read *et al.*, 2011). Here the term ensemble refers to an ensemble of multi-label classifiers instead of an ensemble of binary classifiers already mentioned before. ECC trains m classifier chains, each with a random chain ordering and a random subset of instances. These parameters of ECC contributes to the uniqueness of each classifier chain which helps with variance reduction when their predictions are combined. These predictions are 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 predicted multi-label set (Read *et al.*, 2011) (copied from). More details still to cover in article.

CC and ECC has an advantage over the ensemble methods of BR, that it is not necessary for an initial step of training to obtain predictions of labels that can later be used as features, it does this simultaneously.

Paper still need to look at for CC (Sucar *et al.*, 2013).

2.4.2.3 Calibrated Label Ranking

- basic idea
- notation
- process for unseen instance
- threshold function
- enriched version of pairwise comparison
- psuedo-code
- remarks: second-oder; one-vs-one; mitigates class imbalance issue; quadratic increase in classifiers and improvements thereof; computational complexity

2.4.2.4 Random k-Labelsets

- basic idea

- notation
- LP
- limitations of LP: incompleteness and inefficiency
- why rakel improves it
- psuedo-code
- remarks: high-order; ensembling; computational complexity

The other widely known problem transformation approach is the label powerset (LP) algorithm. Each combination of the labels is seen as a distinct class and then a standard multiclass classification learner can be applied. More formally, the transformation $h : L \to P(L)$ is applied (Tsoumakas and Vlahavas). Thus label correlations are taken into account but LP has other limitations. The number of possible classes increase exponentially with the increase in K and some of the classes/combinations are under-represented (if represented at all) in the training set. This leads to the difficult problem of learning from unbalanced classes and also restricts the algortihm to only predict combinations of labels present in the training set. Labels (or labelsets) that only occur a limited number of times are called tail labels. These are generally the ones difficult to model and a classifier can easily neglect their importance (Xu *et al.*, 2016).

One way to reduce the number of resulting classes after a label powerset transformation is to create meta-labels (not to be confused with meta in the stacking sense) (Read *et al.*, 2014). Meta-labels represent partitions of the label set, but I still do not fully understand the concept. Seems like after the transformation we still end up with a multi-label problem. Investigate further.

Another option is to throw away the combinations that appear infrequently in the training set. This obviously limits the possible output of the mutli-label algorithm even more. Sounds like PPT (Read, 2008).

The best way, according to the literature (Zhang and Zhou, 2014), to overcome the two main limitations of LP is another ensemble based approach (similar idea to stacking) called random k-labelsets (RAkEL). It was first introduced in (Tsoumakas and Vlahavas). It reduces the number of classes to predict from and allows each class to have more training instances compared to the LP method (Lo *et al.*, 2013). Each learner in the ensemble constructs an LP classifier from a random subset of k labels, referred to as k-labelsets. The randomness of this approach can also be its downfall, since the chosen subsets may not cover all labels and inter-label correlations (Systems and Aviv-yafo, 2014). We will use the term L^k to denote the set of all distinct k-labelsets on L. The size of L^k is given by $|L^k| = \binom{|L|}{k}$. The RAkEL algorithm iteratively constructs m LP classifiers from a randomly select k-labelset, where each k-labelset is sampled from L^k without replacement. The parameters to be tuned for RAkEL is the size of the ensemble (number of iterations), m, and the size of the labelsets, k. If $k = 1$ and $m = |L| = K$ this is just an ensemble of BR classifiers. If $k = |L|$ then it becomes the ordinary LP method. (Tsoumakas and Vlahavas) suggests a small k with a sufficient number of ensembles, m, which they show to manage label correlations efficient- and effectively.

To make a prediction on a new instance, the new instance is feeded to each classifier in the ensemble and then their outputs are combined. To determine if label j is relevant for this new instance, the proportion of classifiers in the ensemble indicating that label l is relevant is obtained and then the final output is 1 if this proportion is greater than some threshold t. Intuitively $t = 0.5$ makes sense, since this is equivalent to a majority vote, but RAkEL has been shown to work for a wide range of t-values.

Another factor to consider is the value of k. We do not want it to big. Also how many models should be included in the ensemble?

There are other ways of choosing subsets of the labelset, references in (Systems and Aviv-yafo, 2014).

Note, with all these ensemble extensions, we can still try different ways of ensembling/stacking, especially with RAkEL. Not only taking the average but also by assigning weights to each model or by fitting a model to the predictions. Think (Lo *et al.*, 2013) is an example of this with generalised k-labelsets ensemble.

- LP takes the label dependence into account, but the conditional one: it is well-tailored for the subset 0/1 loss, but fails for the Hamming loss.
- LP may gain from the expansion of the feature or hypothesis space.
- One can easily tailor LP for solving the Hamming loss minimization problem, by marginalization of the joint probability distribution that is a by-product of this classifier.

2.4.3 Algorithm Adaption Methods

These are methods tackling the multi-label learning task by adapting, extending and/or customising an existing supervised learning algorithm (Madjarov *et al.*, 2012).

The main weakness of algorithm adaption methods is that they are mosty tailored to suit a specific model, whereas problem transformation methods are more general and allows for the use of many well-known and effective single-label models (Systems and Aviv-yafo, 2014) (algorithm independent).

2.4.3.1 Multi-Label k-Nearest Neighbour (ML-kNN)

- basic idea
- procedure
- psuedo-code
- remarks: first-order; merits of lazy learning and Bayesian reasoning; mitigate class-imbalance; extensions/variations; computational complexity

2.4.3.2 Multi-Label Decision Tree (ML-DT)

- basic idea
- procedure
- psuedo-code
- remarks: first-oders; efficient; improve with pruning and or ensembling; computational complexity

2.4.3.3 Ranking Support Vector Machine (Rank-SVM)

- basic idea
- procedure
- psuedo-code
- remarks: second-order; variants; computational complexity

2.4.3.4 Collective Multi-Label Classifier (CML)

- basic idea
- procedure
- psuedo-code
- remarks: second-order; conditional random field model; DAG; computational complexity

2.4.4 Summary

- table: basic idea; order; complexity (test/train); domains; optimised metric
- interpretation
- comment on applicability to data sets
- first-order AA is not BR

2.5 Related Learning Strategies

- multi-instance
- ordinal classification
- multi-task learning
- data streams classification
- multi-task learning; multiple-labels learning; multi-intance learning; reverse multi-label learning; preferential text classification; weak label problem; multi-valued multi-label; graded multi-label; multi-view learning

2.6 Pitfalls and Guidelines

- choosing algo is hard
- more empirical evidence is needed; with with wide range of data sets, algos and measures; compare with statistical tests and consider computation time (training and test)
- part on statistical tests in (Gibaja and Ventura, 2015)
- gives reccomendations from empirical study
- mentions label dependence modelling from dembcsz
- trees for efficiency, ensembles for predictive performance, transformation methods for flexibility

2.7 Conclusion

- what was done in the paper
- no formal charaterisation on the underlying concept or any principled mechanism on the appropriate usage of label correlations
- correlations might be asymmetric and or local
- label correlation understanding is holy grail of ML
- complement of this paper would be a broad empirical study
- what follows in next chapter(s)
- highlight challenges again and mention those we are going to focus on: label dependence, high dimensionality
- consider the addition of empirical study
- will actually contribute empirically but with specific questions in mind (end of each chapter)

Chapter 3

Label Dependence

3.1 My thoughts (remove later)

With this chapter I want to investigate the need for approaches in multi-label classification which model the dependence structure between labels. For this we need a sound theoretical definition and analysis of label dependence and then we might want to investigate it empirically with synthetic datasets (or real world). The main papers inspiring this chapter are (Dembcz *et al.*, 2012) and (Read and Hollmén, 2015), and some content will be taken from (Read and Hollmen, 2014), (Madjarov *et al.*, 2012) (for empirical evidence maybe), (Read *et al.*, 2011), (Dembczy, 2010), (Dembczynski *et al.*, 2012). My main hypothesis is that modelling the input-output pairs individually should have just as good, if not better performance compared to approaches trying to model label dependence, since all the available information of the labels should be contained in X and by the assumption that label y_i can be determined with the help of the knowledge of label y_j, it should also be possible to find y_i from X since y_j is found from X. This argument probably only holds for approaches trying to "correct" binary relevance (BR) with regards to its lack of modelling label dependence, such as classifier chains (CC), stacking like MBR/2BR/BR+, etc. Reformulate hypothesis later.

3.2 Introduction

It is essentially a given in multi-label classification literature that in order to obtain competitive results, a learner should be able to model the dependence structure between labels in some way. Whenever a new MLC algorithm is proposed, it will be compared to independent label learning (BR) and if it has superior empirical performance, it is usually ascribed to its ability of modelling label dependence in some ad-hoc way (examples?). The authors of (Dembczy, 2010), (Dembczynski *et al.*) and (Dembczyński *et al.*, 2010) were the first to point out this lack of understanding of the term *label dependence* in the literature (later on a comprehensive and extended discussion of the topics covered in the aforementioned papers was given in (Dembcz *et al.*, 2012)). They argued that *label dependence* is only understood and used by most in the literature in a purely intuitive manner, and that in order to build a better understanding of multi-label classifiers, theoretical backing is essential.

Modelling each label independently, *i.e.* using the binary relevance (BR) approach, is one of the simplest and most intuitive approaches to tackling the multi-label problem. But it has been criticized and overlooked by the majority because it does not take into account the possible dependence between labels. However, BR has many advantages. (Dembcz *et al.*, 2012) shows that BR is the risk minimizer of the Hamming Loss and (Read and Hollmen, 2014) pointed out that it is very rare for 'improved' methods to achieve significantly better results than BR in terms of this measure (also visible in (Madjarov *et al.*, 2012) (make sure)). In addition, BR is highly resistant to overfitting label combinations, since it does not expect samples to be associated with previously-observed combinations of labels [Read2011a]. It can naturally handle data streaming or other dynamic scenarios where the addition and removal of labels are quite common. BR's biggest strength is its low computational complexity compared to other multi-label classification methods. It scales linearly with increasing number of labels and it is easily parallelizable - desirable properties, especially working with large label sets.

Recently, (Read and Hollmén, 2015) has gone so far as to claim that BR can perform just as well as methods supposedly modelling label dependence, and if it does not, it is usually because of the inadequacy of the base learners used. In other words, if the base learner can extract the right features, BR will be as good as any other multi-label classifier, without the need to model label dependence. Some theoretical justifications were given but the empirical evidence was not convincing. This is what motivated the writing of this chapter - to answer the question, "is it essential for a multi-label classifier to take label correlations into account in order to be optimal?". To investigate this one needs a thorough, theoretical understanding of *label dependence*, how to possibly exploit it and how to evaluate it. This is what this chapter aims to do. Most of the work is based on the papers (Dembcz *et al.*, 2012) and (Read and Hollmén, 2015). We will also attempt to back up the theory with empirical results.

3.3 Two types of label dependence

As mentioned, most mutli-label learning papers display merely an intuitive understanding of *label dependence*, in the sense that in predicting a specific label, the information on the rest of the labels may be helpful. For example in an image recognition problem, if a picture is labelled with *beach* and *ocean*, *sand* will most likely be a relevant label. Clearly, this understanding is insufficient to gain advances in the multi-label learning literature (later on it will also be pointed out why this may indeed not make intuitive sense). In this section, a formal statistical definition of the two types of label dependence will be given. First, we briefly revisit the task of multi-label classification (MLC), in mathematical(?) terms.

3.3.1 The task of multi-label classification

deurmekaar met wanneer simbole moet hoof- of kleinletters wees.

Let $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ define a multi-label dataset. $\boldsymbol{x}$, is the feature/input/instance vector of an observation and is given by a p-dimensional real-valued vector, $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$, *i.e.* $\boldsymbol{x} \in \mathcal{R}^p$. Each instance, $\boldsymbol{x}$ is associated with a subset of labels $L \in 2^{\mathcal{L}}$, where $2^{\mathcal{L}}$ represents the powerset of the full set of labels, $\mathcal{L} = \{l_1, l_2, \ldots, l_K\}$. The subset L is represented as an indicator vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_K)$, where $y_k = 1$ if $l_k \in L$ or else $y_k = 0$, for $k = 1, 2, \ldots, K$. We assume examples in $\mathcal{D}$ to be independently and identically distributed (*i.i.d.*) from $P(\boldsymbol{X}, \boldsymbol{Y})$. Let h define a multi-label classifier, which is a mapping,

$$h : \boldsymbol{X} \to \boldsymbol{Y}$$

(not sure about this notation). The risk of h is defined as the expected loss over the joint distribution $P(\boldsymbol{X}, \boldsymbol{Y})$:

$$R_L(h) = E_{\boldsymbol{XY}}\left[L\left(\boldsymbol{Y}, h(\boldsymbol{X})\right)\right],$$

where $L(.)$ is a multi-label loss function. The MLC task boils down to given training data, $\mathcal{D}$, drawn independently from $P(\boldsymbol{X}, \boldsymbol{Y})$, learn a classifier h that minimizes the risk with respect to a specific loss function, *i.e.*

$$h^* = \arg\min_h E_{\boldsymbol{XY}}\left[L\left(\boldsymbol{Y}, h(\boldsymbol{X})\right)\right] = \arg\min_h E_{\boldsymbol{X}}\left[E_{\boldsymbol{Y}|\boldsymbol{X}}\left[L\left(\boldsymbol{Y}, h(\boldsymbol{X})\right)\right]\right],$$

where h^* is the so-called risk-minimizing model and can be determined in a pointwise way by the risk minimizer,

$$h^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} E_{\boldsymbol{Y}|\boldsymbol{X}}\left[L(\boldsymbol{Y}, \boldsymbol{y}))\right].$$

Note, here we allow $h(\boldsymbol{x})$ to take on real values, *i.e.* $h(\boldsymbol{x}) \in \mathcal{R}^K$, for the sake of generality. This is to cover multi-label ranking functions and multi-label classifiers that output real real values before thresholding.

3.3.2 Marginal vs. conditional dependence

First note that we denote the conditional distribution of $\boldsymbol{Y} = \boldsymbol{y}$ given $\boldsymbol{X} = \boldsymbol{x}$ as

$$P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x}) = P(\boldsymbol{y}|\boldsymbol{x})$$

and the corresponding conditional marginal distribution of Y_k (conditioned on $\boldsymbol{x}$) as

$$P(Y_k = b|\boldsymbol{x}) = \sum_{y_i=b} P(\boldsymbol{y}|\boldsymbol{x}).$$

(can probably also write as $P(Y_k|\boldsymbol{x})$ since b is either 0 or 1?)

(Dembcz *et al.*, 2012) defines two types of dependence among lables, namely, conditional dependence and marginal dependence. Their definitions follow:

Definition 1 *A random vector of labels* $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_K)$ *is called marginally independent if*

$$P(\boldsymbol{Y}) = \prod_{k=1}^{K} P(Y_k). \tag{3.3.1}$$

Marginal dependence is also known as unconditional dependence and can be thought of as a measure of the frequency of co-occurrence among labels. Conditional dependence captures the dependence of the labels given a specific observation $\boldsymbol{x}$.

Definition 2 *A random vector of labels is called conditionally independent, given* $\boldsymbol{x}$ *if*

$$P(\boldsymbol{Y}|\boldsymbol{x}) = \prod_{k=1}^{K} P(Y_k|\boldsymbol{x}). \tag{3.3.2}$$

The conditional joint distribution of a random vector $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_K)$ can be expressed by the product rule of probability ($P(AB) = P(A|B)P(B)$):

$$P(\boldsymbol{Y}|\boldsymbol{x}) = P(Y_1|\boldsymbol{x}) \prod_{k=2}^{K} P(Y_k|Y_1, \ldots, Y_{k-1}, \boldsymbol{x}). \tag{3.3.3}$$

A similar expression can be given for $P(\boldsymbol{Y})$. If $Y_1, Y_2, \ldots, Y_K$ are conditionally independent, then Equation 3.3.3 will simplify to Equation 3.3.2.

Marginal and conditional dependence are closely related - it can be written as:

$$P(\boldsymbol{Y}) = \int_{\mathcal{X}} P(\boldsymbol{Y}|\boldsymbol{x}) d\mu(\boldsymbol{x}), \tag{3.3.4}$$

where μ is the probability measure on the input space $\mathcal{X}$ induced by the joint probability distribution P on $\mathcal{X} \times \mathcal{Y}$. Marginal dependence can roughly be viewed as an 'expected dependence' over all instances. Nevertheless, marginal dependence does not imply conditional independence, or *vice versa.* Two examples from (Dembcz *et al.*, 2012) are given to illustrate this.

Example 1 *Suppose two labels, Y_1 and Y_2, are independently generated from $P(Y_k|\boldsymbol{x}) = (1 + \exp(-\phi f(\boldsymbol{x}))^{-1}$, where ϕ controls the Bayes error rate. Thus, by definition, the two labels are conditionally independent with conditional joint distribution, $P(\boldsymbol{Y}|\boldsymbol{x}) = P(Y_1|\boldsymbol{x}) \times P(Y_2|\boldsymbol{x})$. However, as $\phi \to \infty$, the Bayes error tends to zero and the marginal dependence increases to an almost deterministic case of $y_1 = y_2$. Showing, conditional independence does not imply marginal independence.*

Example 2 *Suppose two labels, Y_1 and Y_2, are to be predicted by using a single binary feature, x_1. Let the joint distribution $P(X_1, Y_1, Y_2)$ be given by the following table:*

x_1	y_1	y_2	P
0	0	0	0.25
0	0	1	0.00
0	1	0	0.00
0	1	1	0.25
1	0	0	0.00
1	0	1	0.25
1	1	0	0.25
1	1	1	0.00

Thus, the labels are not conditionally independent,

$$P(Y_1 = 0, Y_2 = 0|x_1 = 1) = 0 \neq P(Y_1 = 0|x_1 = 1)\times P(Y_2 = 0|x_1 = 1) = 0.25\times 0.25,$$

but it can be shown that they are indeed marginally independent. For example,

$$P(Y_1 = 0, Y_2 = 0) = 0.25 = P(Y_1 = 0) \times P(Y_2 = 0) = 0.5 \times 0.5.$$

This holds for all the combination of labels, showing that marginal independence does not imply conditional independence.

This distinction between marginal and conditional dependence is crucial in the attempt to model label dependence in multi-label classification. We describe a multi-output model with the following notation, similar to (Hastie *et al.*, 2009):

$$Y_k = h_k(\boldsymbol{X}) + \epsilon_k(\boldsymbol{X}), \tag{3.3.5}$$

for all $k = 1, 2, \dots, K$. $h_k : \boldsymbol{X} \to \{0, 1\}$ will be referred to as the structural part and $\epsilon_k(\boldsymbol{x})$ as the stochastic part of the model. Note that a common assumption in multi-variate regression (real-outputs) is that

$$E[\epsilon_k(\boldsymbol{x})] = 0. \tag{3.3.6}$$

for all $\boldsymbol{x} \in \boldsymbol{X}$ and $k = 1, 2, \ldots, K$. This is not a reasonable assumption in mutli-label classification (Dembcz *et al.*, 2012) - the distribution of the noise terms can depend on $\boldsymbol{x}$ and two or more noise terms can depend on each other. Classifier h_k might also be very similar to h_l, $l \neq k; l = 1, 2, \ldots, K$. Thus there are two possible sources of label dependence: the structural part and the stochastic part of the model.

It seems that marginal dependence between labels is caused by the similarity between the structural parts. This assumption is made since it is reasonable to assume that the structural part will dominate the stochastic part. Suppose there exists a function $f(.)$ such that $h_k \approx f \circ h_l$, *i.e.*

$$h_k(\boldsymbol{x}) = f(h_l(\boldsymbol{x})) + g(\boldsymbol{x}), \tag{3.3.7}$$

with $g(.)$ being negligible in the sense that $g(\boldsymbol{x}) = 0$ with high probability. Then this $f(.)$-*dependence* between the classifiers is likely to dominate the averaging process in Equation 3.3.4, compared to $g(.)$ and the stochastic parts. This is what happens in Example 1 when $\phi \to \infty$. Thus we see that even if the dependence between h_k and h_l is only probable, it can still induce a dependence between the labels Y_k and Y_l (verstaan nie presies wat hier bedoel word nie). Another example illustrating idea is given from (Dembcz *et al.*, 2012).

Example 3 *Consider a problem with a 2-dimensional input $\boldsymbol{x} = (x_1, x_2)$, where x_i is uniformly distributed in $[-1, 1]$ for $i = 1, 2$, and two labels, Y_1, Y_2, determined as follows. Y_1 is set to 1 for all positve values of x_1, i.e. $Y_1 = I(x_1 > 0)$. The second label is generated similarly but with the decision boundary of Y_1 ($x_1 = 0$) rotated by an angle of $\alpha \in [0, \pi]$ (give illustration). In addition, let the two error terms of the model be independent and both flip the label with a probability of 0.1. If α is close to zero, the labels will almost be identical and a high correlation will be observed between them. But if $\alpha = \pi$, the decision boundaries of the labels are orthogonal and a low correlation will be observed.*

With regards to Equation 3.3.7, in Example 3, $f(.)$ is the identity function and $g(.)$ given by the ± 1 in the regions between the decision boundaries. From this point of view, marginal dependence can be seen as a kind of soft constraint that a learning algorithm can exploit for the purpose of regularization (Dembcz *et al.*, 2012). (verstaan nie wat dit beteken nie)

For the conditional dependence, it seems that the stochastic part of the model is the cause. In Example 3, Y_1 and Y_2 is conditionally independent because the error terms are assumed to be independent. However, if there is a close relationship between ϵ_1 and ϵ_2, this conditional independence will be lost. (Dembcz *et al.*, 2012) proves the proposition that a vector of labels is conditionally dependent given $\boldsymbol{x}$ if and only if the error terms in Equation 3.3.5

are conditionally dependent given $\boldsymbol{x}$, *i.e.*

$$E\left[\epsilon_1(\boldsymbol{x}) \times \cdots \times \epsilon_K(\boldsymbol{x})\right] \neq E\left[\epsilon_1(\boldsymbol{x})\right] \times \cdots \times E\left[\epsilon_K(\boldsymbol{x})\right].$$

(Include proof?) It should also be noted that conditional independence can also cause marginal dependence because of Equation 3.3.4. Thus the similarity between models is not the only source of of marginal dependence.

What we have learned thus far is that there is a difference between marginal and conditional label dependence. The presence of marginal dependence does not imply conditional label dependence and *vice versa*. If label correlations are observed it can only be assumed that marginal dependence between the labels exist. It does not necessarily imply that there are any dependencies among the error terms (although it could be the cause). On the other hand, if conditional dependence is observed, one can safely assume that there are dependencies among the error terms. Next, we see how to exploit both types of label dependence to improve predictive accuracy.

3.4 Link between label dependence and loss minimization

One can view the MLC task from different persepectives in terms of loss minimizations. (Dembcz *et al.*, 2012) describes three such views, determined by the type of loss function to be minimized, the type of dependence taken into account and the distinction between marginal and joint distribution estimation. The three views and the main questions to consider for each of them are:

1. The individual label view: How can we improve the predictive accuracy of a single label by using information about other labels?
2. The joint label view: What type of non-decomposable MLC loss functions is suitable for evaluating a multi-label prediction as a whole and how to minimize such loss functions?
3. The joint distribution view: Under what conditions is it reasonable to estimate the joint conditional probability distribution over all label combinations?

3.4.1 The individual label view

With this view, the goal is to minimize a loss function that is label-wise decomposable and we want to determine whether or not it will help taking label relationships into account. The most common and intuitive label-wise decomposable loss function is the Hamming loss, which is defined as the fraction of labels whose relevance is incorrectly predicted:

$$L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{K}\sum_{k=1}^{K} I(y_k \neq \hat{y}_k). \tag{3.4.1}$$

Equation 3.4.1 is only the Hamming loss for one observation. To compute the Hamming loss over an entire dataset, Equation 3.4.1 is averaged over all the observations.

It is easy to see that the Hamming loss is minimized when

$$\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_K),$$

where

$$\hat{y}_k = \arg\max_{y_k \in \{0,1\}} p(y_k|\boldsymbol{x}),$$

for $k = 1, 2, \ldots, K$. This shows that it is enough to take only the conditional marginal distribution $P(Y_k|\boldsymbol{x})$ into account to solve the problem, at least on a population level. Thus the Hamming loss is minimized by BR. (Dembcz *et al.*, 2012) also gives a similar result for label-wise decomposable loss functions in general (thus also relevant for F-measure, AUC, *etc.*). This result implies that the multiple single label predictions problem can be solved on the basis of $P(Y_k|\boldsymbol{x})$ alone. Hence, with a proper choice of base classifiers and parameters for estimating the conditional marginal probabilities, there is in principle no need for modelling conditional dependence between the labels. However, in cases where the base classifiers are inadequate, dependence between the errors will exist and BR will give a suboptimal solution (make sure this statement is used correctly). Methods exist to improve BR in these situations and will be discussed shortly.

3.4.2 The joint label view

Here we are interested in non-decomposable (label-wise) MLC loss functions such as rank loss and the subset 0/1 loss. We discuss when they are appropriate and how to minimize them. First, consider the rank loss. Suppose the true labels constitute a ranking in which all relevant labels ideally precede all irrelevant ones and $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \ldots, h_K(\boldsymbol{x}))$ is seen as a ranking function representing a degree of label relevance sorted in a decreasing order. The rank loss simply counts the number of label pairs that disagree in these two rankings:

$$L_r(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \sum_{(k,l);y_k>y_l} \Big(I(h_k(\boldsymbol{x}) < h_l(\boldsymbol{x})) + \frac{1}{2} I(h_k(\boldsymbol{x}) = h_j(\boldsymbol{x})) \Big). \tag{3.4.2}$$

This function is not convex nor differentiable, thus an alternative would be to minimize a convex surrogate like the hinge or exponentional function. However, (Dembcz *et al.*, 2012) proves that it is enough to minimize Equation 3.4.2 by sorting the labels by their probability of relevance:

Theorem 1 *A ranking function that sorts the labels according to their probability of relevance, i.e. using the scoring function $\boldsymbol{h}(.)$ with $h_k(\boldsymbol{x}) = P(Y_k = 1|\boldsymbol{x})$, minimizes the expected rank loss.*

(include proof?) This implies again (just like in the case for the label-wise decomposable loss functions) that, in principle, it is not necessary to know the joint label distribution $P(\boldsymbol{Y}|\boldsymbol{x})$ when training a multi-label classifier, *i.e.* risk-minimizing predictions can be made without any knowledge about the conditional dependency between labels. Thus, to minimize the rank loss, one can simply use any approach minimizing the single label losses. Note this results does not hold for the normalized version of rank loss.

Next, we look at the extremely stringent multi-label loss function, the subset 0/1 loss:

$$L_S\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = I\left(\boldsymbol{y} \neq \hat{\boldsymbol{y}}\right). \tag{3.4.3}$$

Although most would agree that this is not a fair measure for MLC performance, since it does not disinguish between almost correct and completely wrong, it is still interesting to study with regards to exploiting label dependence. The risk-minimizing prediction for Equation 3.4.3 is given by the mode of the distribution:

$$h_s^*(\boldsymbol{x}) = \arg\max_{\boldsymbol{y}} P(\boldsymbol{Y}|\boldsymbol{x}). \tag{3.4.4}$$

This implies that the entire distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$ is needed to minimize the subset 0/1 loss. Thus a risk minimizing prediction requires the modelling of the joint distribution and hence the modelling of the conditional dependence between labels. Later on we will show an important results that under independent outputs, minimizing the Hamming loss and the subset 0/1 loss is equivalent, implying that BR will indeed also minimize the subset 0/1 loss (consider to show it here).

The cases for F-measure loss and the Jaccard distance is a bit more complicated and will not be discussed here. (give citation of where this can be found)

3.4.3 The joint distribution view

> not sure if I want to mention the joint distribution view. Maybe only distinguish between single label and joint label prediction approach.

We just saw that minimzing the subset 0/1 loss requires the estimation of the entire conditional joint distribution, $P(\boldsymbol{Y}|\boldsymbol{X})$. Generally, if the joint distribution is known, a risk-minimizing prediction can be derived for any loss function in an explicit way:

$$h^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} E_{\boldsymbol{Y}|\boldsymbol{x}}\left[L(\boldsymbol{Y}, \boldsymbol{y})\right].$$

In some applications modelling the joint distribution may result in using simpler classifiers, potentially leading to a lower cost and a better performance compared to directly estimating marginal probabilities by means of more complex classifiers. Nevertheless, it remains a difficult task. One has to estimate 2^{K} values to estimate for a given $\boldsymbol{x}$.

...

3.5 Previous attempts to 'exploit' label dependence

3.6 Improved attempts to 'exploit' label dependence

Theoretical insights into MLC

- when new MLC algorithm is introduced, it should be specified which loss functions it intends to minimize. Otherwise it may give misleading results (like that it is optimal for many loss functions).
- a classifier supposed to be good in solving one problem may perform poorly on a different problem and vice versa
- restricts attention to hamming loss and subset 0/1 loss.
- hamming is representative of single label scenario and subset 0/1 for the mutli-label loss.
- assumes unconstrained hypothesis space.
- proposition (with proof in paper): The hamming loss and subset 0/1 loss have the same risk-minimizer, *i.e.* $\boldsymbol{h}^*_H(\boldsymbol{x}) = \boldsymbol{h}^*_s(\boldsymbol{x})$, if one of the following conditions holds: (1) Labels $Y_1, \dots, Y_K$ are conditionally independent, *i.e.* $P(\boldsymbol{Y}|\boldsymbol{x}) = \prod_{k=1}^{K} P(Y_k|\boldsymbol{x})$. (2) The probability of the mode of the joint probability is greater than or equal to 0.5, *i.e.* $P(\boldsymbol{h}^*_S(\boldsymbol{x})|\boldsymbol{x}) \geq 0.5$.
- corollary (with proof in paper): In the separable case (*i.e.* the joint conditional distribution is deterministic, $P(\boldsymbol{Y}|\boldsymbol{x}) = I(\boldsymbol{Y} = \boldsymbol{y})$), the risk minimizers of the hamming loss and subset 0/1 loss coincide.
- Then 3 propositions on upper bounds of these losses. Ponder its relevance.

MLC algorithms for exploiting label dependence

- proposed algorithms improve predictive performance by supposedly modelling label dependence.

- type of dependence and loss to be optimized is omitted
- leads to poor designs and misleading results.
- focus on PT methods
- discussion on BR:
- simplest. does not take marginal or conditional dependence into account.
- in general not able to yield risk-mininizing predictions for multi-label losses but is well suited for loss functions whose risk-minimizer can solely be expressed in terms of marginal (conditional) distributions.
- may be sufficient, but exploiting marginal dependencies may still be beneficial especially for small-sized problems.
- moves discussion to single label predictions
- several methods that exploit similarities between structural parts of the label models.
- general scheme:

$$\boldsymbol{y} = \boldsymbol{b}(\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{x}), \tag{3.6.1}$$

where $\boldsymbol{h}(\boldsymbol{x})$ is the binary relevance learner and $\boldsymbol{b}(.)$ is an additional classifier that shrinks or regularizes the solution of BR. Or

$$\boldsymbol{b}^{-1}(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}), \tag{3.6.2}$$

where the output space is first transformed and then the BR classifiers are trained and then transformed back to original. + Stacking follows first scheme. Form of regularization or feature expansion. Not clear which inputs should all be use for second level. + multivariate regression + kernel dependency estimation + compressive sensing + next section on methods that seek to estimate the joint distribution $P(\boldsymbol{Y}|\boldsymbol{x})$. + LP. Largest drawback the number of label combinations + the literature usually claims LP is generally the rigth approach. FALSE. LP takes conditional dependence into account but usually fails for losses like Hamming. + can improve with RAKEL, but it is still not well understood from a theoretical point of view. + PCC. computationally more manageable. ECC to reduce importance of label chain order.

Experimental evidence

- real and synthetic data
- BR, SBR, CC, LP
- hamming loss and subset 0/1
- MULAN
- logistic regression for base classifier.
- marginal independence: stacking does improve on BR, CC similar to SBR, LP also bad. Error increases with number of labels. hamming and subset 0/1 coincide.

- conditional independence: again loss functions coincide. SBR improves over BR, even higher when structural parts are more similar.Supports theoretical claim that the higher the structural similarties the more prominent effect of stacking. Study rest of results.
- conditional dependence:
- xor problem

Conclusions

- study

Nou opsomming van (Read and Hollmén, 2015) - sodra klaar, probeer in hoofstuk inkorporeer.

Introduction

- n-th feature vector $\boldsymbol{x}^{(n)} = [x_1^{(n)}, \dots, x_p^{(n)}]$, where $x_j \in \mathcal{R}$, $j = 1, \dots, p$.
- in the traditional binary classification task we are intersted in having a model h to provide a prediction for test instances $\tilde{\boldsymbol{x}}$, *i.e.* $\hat{y} = h(\tilde{\boldsymbol{x}})$. In MLC there are K binary output class variables (labels) and thus $\hat{\boldsymbol{y}} = [\hat{y}_1, \dots, \hat{y}_K] = h(\boldsymbol{x})$.
- probabilistic speaking h seeks the expecctation $E[\boldsymbol{y}|\boldsymbol{x}]$ of unknown $p(\boldsymbol{y}|\boldsymbol{x})$. This task is typically posed as a MAP estimate of the joint posterior mode
$$\hat{\boldsymbol{y}} = [\hat{y}_1, \dots, \hat{y}_K] = h(\tilde{\boldsymbol{x}}) = \arg \max_{\boldsymbol{y} \in \{0,1\}^p} p(\boldsymbol{y}|\tilde{\boldsymbol{x}})$$
This corresponds to minimizing the susbet 0/1 loss.
- $h_{BR}(\tilde{\boldsymbol{x}}) := [h_1(\tilde{\boldsymbol{x}}), \dots, h_K(\tilde{\boldsymbol{x}})]$
- entirety of ML literature point out that BR obtain suboptimal performance because it assumes labels are independent.
- several approaches attempt to correct/regularize BR, SBR.
- others attempt to learn the labels together, LP. $\hat{\boldsymbol{y}} = h_{LP}(\tilde{\boldsymbol{x}})$
- another example is CC done using a greedy search:
$$h_{CC}(\tilde{\boldsymbol{x}}) := [h_1(\tilde{\boldsymbol{x}}), h_2(\tilde{\boldsymbol{x}}, h_1(\tilde{\boldsymbol{x}})), \dots, h_K(\tilde{\boldsymbol{x}}, \dots, h_{K-1}(\tilde{\boldsymbol{x}}))]$$
- PCC formulates CC as the joint distribution using the chain rule,
$$h_{CC}(\boldsymbol{x}) := \arg\max_{\boldsymbol{y}} p(y_1|\boldsymbol{x}) \prod_{k=2}^{K} p(y_k|\boldsymbol{x}, y_1, \dots, y_{K-1})$$
and show that it is indeed possible to make a Bayes-optimal search with guarantees to the optimal solution for 0/1 loss. Several search techniques exist to make the seach optimal, but greedy is still popular.

- order and structure of chains in cc is the main focus point.
- although in theory the chain rule holds regardless of the order of variables, each $p(y_k|\boldsymbol{x}, y_1, \ldots, y_{K-1})$ is only an approximation of the true probability because it is modelled from finite data under a constrainded class of model, and consequently a different indexing of labels can lead to different results in practice.
- many approaches try to find the best order and show better empirical results, but the reason why is not quite clear
- LP can be viewed as modelling the joint probability directly,

$$h_{LP}(\boldsymbol{x}) := \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}, \boldsymbol{x})$$

- two main points from previous papers: (1) the best label order is impossible to obtain from observational data only. (2) the high performance of classifier chains is due to leveraging earlier labels in the chain as additional feature attributes.

The role of label dependence in multi-label classification

- marginal dependence: frequency of co-occurence among labels
- conditional dependence: after conditioning on the input
- modelling complete dependence is intractable
- rather attempt pairwise marginal dependence or use of ensemble.
- many new methods do not outperform each other over a reasonable amount of datasets.
- improvements of prediction on standard multi-label datasets reached a plateau (maybe investigate).
- question the logic, if the ground truth label dependence could be known and modelled, multi-label predictive performance would be optimal and therefore as more technique and computational effort is invested into modelling label dependence, the lead of the new methods over BR and other predecessors will widen.
- BR might be underrated
- modelling label dependence is a compensation of lack of training data and one could only assume that given infinite data two separate binary models on labels y_k and y_l could achieve as good performance as one that models them together.
- the 'intuitive' understanding actually seems quite flawed: if we take two labels and wish to tag images with them, the assumption that label dependence is key to optimal multi-label accuracy is analogous to assuming that an expert trained for visually recognising one label will make optimum classifications only if having viewed the classiication of an expert trained on the other label.

- in reality, modelling label dependence only helps when a base classifier behind one or more labels is inadequate.
- depends on the base classifier
- there is no guarantee that an ideal structure based on label dependence can be found at all given any amount of training data.
- see XOR problem
- take the view that BR can perform as well as any other method when there is no dependence among the outputs given the inputs.
- not to say that BR should perform as well as other methods if there is no dependence *detected*. Due to noisy data or insufficient model dependence may be missed or even introduced.
- if a ML method outperforms BR under the same base classifier then we can say that it using label dependence to compensate for the inadequacy in its base classifiers.
- attempt to remove the dependence among the labels
- dependence generated by inadequate base classifiers

Binary relevance as a state-of-the-art classifier

- CC and LP are representative of PT problems. Succesful on many fronts and can be builded on. Still has some drawbacks. Discusses them.
- BR less parameters to tune.
- multi-label classifiers can be comprised of individual binary models that perform equally as well as models explicitly linked together based on label dependence or even a single model that learns labels together (intrinsic label dependence modelling).
- claim this is the case for example and label based metrics. (not what the previous paper found)
- proposition with proof: given $X = x$, there exists a classifier $h'_2(x) \approx \arg\max_{y_2 \in \{0,1\}} p(Y_2|X)$ that achieves at least as small error as classifier $h_2(x) \approx \arg\max_{y_2 \in \{0,1\}} p(Y_2|Y_1, X)$, under loss $L(y_2, \hat{y}_2) = I(y_2 \neq \hat{y}_2) = I(y_2 \neq h_2(x))$. Instances of X, Y_1, Y_2 are given in the training data but only $\tilde{x}$ is given at test time. (see proof in paper)
- This means that if we are interested in a model for any particular label, best accuracy can be obtained in ignorance of other labels.
- proposition and proof: under observations $X = x$, there exists two individually constructed classifiers $h'_1 \approx \arg\max_{y_1} p(Y_1|X)$ and $h'_2 \approx \arg\max_{y_2} p(Y_2|X)$ such that under 0/1 loss, $[h_1(x), h_2(x)] \equiv \hat{\boldsymbol{y}} \equiv \boldsymbol{h}(x)$ are equivalent, where $\boldsymbol{h} \approx \arg\max_{[y_1,y_2]} p(Y_1, Y_2|X)$ models labels together. Instances of X, Y_1, Y_2 are given in the training data but only x (tilde) is given at test time. (see proof in paper)
- following examples, X represents some document and Y_1, Y_2 represent the relevance of two subject categories for it. Latent variable Z represents

the unobservable current events which may affect both the observation X and the decisions for labelling it. (illustration of all of the scenarios)

- ignore case where input and all labels are independent.
- case of conditional independence - a text document is given independently to two human labelers who each independently identify if the document is relevant to their expert domain.

$$\begin{aligned} p(\boldsymbol{y}, x) &= p(y_1, y_2) \\ &= p(y_1|x)p(y_2|y_1, x) \\ &= p(y_1|x)p(y_2|x) \end{aligned}$$

  which obviously can be solved with BR, where $h_k(\tilde{x}) := \arg\max_{y_k} p(y_k|\tilde{x})$.
- a text document is labelled by the first labeller and afterwards by the second expert - potentially biasing the decision to label relevance or not with this second label. If we do not impose any restriction on any $h_k(x)$, it is straightforward to make some latent $z \equiv h_1(x)$ such that $h_2(x, z) \equiv h_2(x, h_1(x))$. We speak of equivalence in the sense that given Z we can recover Y_2 to the same degree of accuracy (probably compared to case without Z). In this analogy the second labeller must learn also the first labeller's knowledge and thus makes the first labeller redundant. If we drop Y_1 we return to the original structure.
- two experts label a document X but both are biased by each other and - possibly to alternate degrees - by an external source of information Z. Can also introduce latent variables Z_1, Z_2 to break the dependence between the labels.
- note the dependence between any variable can be broken by introducing hidden variables not just the label variables. Hence we can further break dependence between X and Y_1 in the same way - if we desire.
- universal approximation: with a finite number of neurons, even with even with a linear output layer, a network can approximate any continuous function. Implies for ML - given a large enough but finite feature representation in the form of a middle layer, any of the labels can be learned independently of the others, *i.e.* a linear BR layer can suffice for optimal classification performance.
- to summarise: if we find dependence between labels it can be seen as a result of marginalizing out hidden variables that generated them. Also, we can add hidden variables to remove the dependence between labels.
- this does not mean we have a method to learn this structure. Which is learning latent variables powerful enough.
- EM and MCMC sampling under energy models to learn latent variables by minimizing the energy and thus maxmimizing the joint probability with observed variables. (iterative procedures).
- unsupervised part more difficult than supervised

- **existing methods to obtain conditional independence among labels.**
- task: making outputs independent of each other by using a different input space to the original such that a simpler classifier can be employed to predict outputs.
- deep learning to learn a powerful higher-level feature representations of the data. (uses multiple hidden layers)
- in MLC the labels can be seen as high-level feature representations.
- **the equivalence of loss metrics under independent outputs**
- if outputs are independent of each other given the input, then minimizing Hamming loss and 0/1 loss is equivalent.
- the risk of Hamming loss is minimized by BR

  $$\hat{y}_k = \arg \max_{y_k \in \{0,1\}} p(y_k|\boldsymbol{x})$$

  for each label. The 0/1 loss on the other hand, is minimized by taking the mode of the distribution,

  $$\hat{\boldsymbol{y}} = \arg \max_{\boldsymbol{y} \in \{0,1\}^K} p(\boldsymbol{y}|\boldsymbol{x})$$

  equivalently written as

  $$\hat{\boldsymbol{y}} = \arg \max_{\boldsymbol{y} \in \{0,1\}^K} p(y_1|\boldsymbol{x}) \prod_{k=2}^{K} p(y_k|\boldsymbol{x}, y_1, \ldots, y_{K-1}).$$

- Noting that when all outputs are independent of each other given the input ($p(y_k|\boldsymbol{x}, y_l) \equiv p(y_k|\boldsymbol{x})$), then for all k, l it becomes

  $$\begin{aligned}\hat{\boldsymbol{y}} &= \arg \max_{\boldsymbol{y} \in \{0,1\}^K} \prod_{k=1}^{K} p(y_k|\boldsymbol{x}) \\ &= \left[\arg \max_{y_1 \in \{0,1\}} p(y_1|\boldsymbol{x}), \ldots, \arg \max_{y_K \in \{0,1\}} p(y_k|\boldsymbol{x})\right].\end{aligned}$$

- here input refers to the input into the model and not the original features.
- we can replace the input with hidden variables derived from the original feature space in order to make them independent. If this is successful, the above holds, and using BR will achieve the same result as CC on either measure.
- suppose only the third of three outputs is successfully made independent, then prediction of independent models is optimizing

  $$\hat{\boldsymbol{y}} = \left[\arg \max_{y_1, y_2 \in \{0,1\}^2} p(y_1, y_2|\boldsymbol{x}), \arg \max_{y_3 \in \{0,1\}} p(y_3|\boldsymbol{x})\right].$$

- if this is the case it could be handled elegantly by RAkELd - disjoint labelset segmentations RAkEL. But detecting these mixed dependence sets is difficult.

- RAkEL and ECC benefit from the ensemble effect of reducing variance of estimates but it is not clear what loss measure is being optimized.

Classifier chains augmented with synthetic labels (CCASL)

- difficult to search for good order in CC
- if 'difficult' label is at start of chain, all other labels may suffer.
- present a method that adds synthetic labels to the beginning of the chain and builds up a non-linear representation, which can be leveraged by other classifiers further down the chain. CCASL
- create H synthetic labels.
- many options - they used threshold linear unit (TLU) to make binary, can also try others like ReLU with continuous output. or sigmoid and radial basis.
- the synthetic labels can be interpreted as random cascaed basis functions, except that at prediction time the values are predicted and thus we refer to them as synthetic labels.
- synthetic label $z_k = I(a_k > t_k)$ with activation values

  $$a_k = \left([B * W]^T_{k,1:(p+(k-1))} \cdot \boldsymbol{x}'_k\right)$$

  where W is a random weight matrix (sampled from multivariate normal) with identically sized masking matrix B where $B_{i,j} \sim Bernoulli(0.9)$, input $\boldsymbol{x}'_k = [x_1, \ldots, x_p, z_1, \ldots, z_{k-1}]$ (not the same k as label index), and threshold $t_k \sim \mathcal{N}(\mu_k, \sigma_k \cdot 0.1)$
- want to use synthetic labels at beginning of chain to improve prediction of the real labels.
- $\boldsymbol{y}' = [z_1, \ldots, z_H, y_1, \ldots, y_K]$ and from the predictions $\hat{\boldsymbol{y}}'$ we extract the real labels $\hat{\boldsymbol{y}} = [\hat{y}'_{H+1}, \ldots, \hat{y}_{H+K}] = [\hat{y}_1, \ldots, \hat{y}_K]$.
- $\hat{y}_j = \arg\max_{y_j \in \{0,1\}} p(y_j | x_1, \ldots, x_p, z_1, \ldots, z_H, y_1, \ldots, y_{j-1})$
- use LR as base classifier
- label order less of an issue.
- does well on complex non linear synthetic data - overfits on simple linear synthetic data.
- lots of tunable parameters
- few hidden labels are necessary for CCASL, empirical suggests $H = K$.
- **CCASL + BR**
- guards against overfitting, removes connections among the output
- advantages of BR, stacking and CC
- no back prop necessary.
- **CCASL+AML**
- CCASL strucutre is powerful for modeling non-linearities. CCASL+BR regularizes but otherwise does not offer a more powerful classifier.
- whereas we created synthetic labels from feature space, we can do the same from the label space.

- layer of binary nodes which are feature functions created from the label space for each subset
- see rest in paper.
- section on other network based literature
- back prop bad
- simply using a powerful non-linear base classifier may remove the need for transformations of the feature space altogether.

Experiments

- done in python and sklearn
- synthetic dataset and music, scene, yeast, medical, enron, reuters (max K = 103)
- 10 iterations for each datset 60/40 split
- report parameters
- all out-perform BR and CC
- BR_{RF} does best under hamming loss! RF are adequately powerful to model each layer
- CCASL are quite expensive
- the main advantage brought by modelling label dependence via connections among outputs is that of creating a stronger learner.
- did not investigate ensembles

3.7 Empirical Ideas

- simulate data with independent labels and see of say MBR still does better than BR
- see the effect of base learner on difference between BR and other 'label' methods

3.8 Introduction

- why this chapter
- aim: how to efficiently exploit label correlations
- methodology: look at what the literature says + do an empirical (simulation) study
- outline

It has been shown repeatedly in the literature that in order to achieve acceptable empirical results, the multi-label algorithm used must in some way or another exploit the dependence/correlation amongst the labels. Unfortunately very little theoretical evidence exists for this suggestion. To delve deeper into this topic an understanding of the evaluation metrics of multi-label classifiers is a fundamental step.

not sure if this should go here and to what extent. This is only an introduction and the rest will be continued after algorithms are introduced.

It has been mentioned here and many times in literature that the exploitation of label structures is essential to an effective multi-label algorithm. The problem is that the correlation/dependence/relationship between labels is not yet well defined in the literature (Zhang and Zhou, 2014). In (Dembcz *et al.*, 2012) the authors comment that researchers often use the term label dependence in an intuitive sense and not as a formally defined concept. Naturally this makes it a hard problem to solve, if it is not well defined. Some valiatnt attempts were made in (Dembczynski *et al.*), (Dembcz *et al.*, 2012) (and others). The following is an overview of them.

In (Zhang and Zhang, 2010) the existing strategies for multi-label classification are divided into categories based on the order of label correlations being considered by the algorithms. So-called first-order approaches are those that do not take label correlations into account. Second-order approaches consider the pairwise relationships between labels and high-order approaches allows for all interactions between labels and/or combinations of labels. First-order strategies simply ignore label correlations, but they are usally simpler. The latter two strategies are far more complex but also limited in some cases. Second-order strategies will not generalise well when higher-order dependencies exist amongst the labels and the the high-order strategies may ‘overfit’ if only subgroups of the labels are correlated (Zhang and Zhang, 2010).

3.9 Exploiting Label Dependence

- (Sorower, 2010)

3.10 Theoretical Results

3.10.1 Intuitive Perspective

3.10.2 Two Types of Label Dependence

3.10.3 Link with Loss Function

3.10.4 Symmetry

- (Huang *et al.*, 2012) claims that most of the time the label dependencies are asymmetric and suggest the MAHR algorithm. Also most of the existing methods exploit label correlations globally, which is not necessarily a good assumption if these correlations only exist for some instances

(Huang and Zhou, 2012). They suggest a ML-LOC algorithm (which seems to do very well).

3.10.5 Locality

3.11 Empirical Analysis

- maybe meta analysis on how others claimed to improved label correlation modelling

3.11.1 Previous Findings

3.11.2 Simulation Study

3.12 Conclusion

- limitations
- recommendations

Chapter 4

Input Space Reduction

4.1 Introduction

- Importance (use)
- Difficulties (different from typical contexts)

4.1.1 Single-Label Framework

4.2 Feature Selection

4.3 Feature Extraction

4.4 Meta-analysis

4.5 Summary

- (Tsoumakas and Vlahavas) mentions a feature selection approach under experimental setup.
- also for FS: Multi-label learning with label-specific feature reduction
- important for FS: A systematic review of multi-label feature selection and a new method based on label construction
- and other of spolaor
- See the LIFT approach by Zhang and Wu and in this paper they also refer to other FS papers (Zhang and Wu).

Chapter 5

Output Space Reduction

5.1 Introduction

Chapter 6

Video Tagging

6.1 Introduction

6.2 General Approaches

6.3 Mutli-Label Tagging

should I include single label papers?

6.4 Introduction

- why is it necessary/relevant now? (Qi *et al.*, 2007) for search, navigation and browsing. More videos than ever - yt
- mention how video classification is unsolved since not many large diverse datasets until recently (Abu-El-Haija *et al.*, 2016).
- can use many techniques from image classification but need to take temporal differences into account.
- briefly why it is difficult: large datasets
- maybe explain how video features are extracted (frame vs video)
- multi-instance learning?

6.5 Definition

- what are the labels representing? (Tang *et al.*, 2012) Content based Video Information Retrieval, Video Semantic Concept Classification

6.6 Existing datasets

- yt8m
- mediamill
- Sports-1M benchmarks

6.7 Other approaches

- (Abu-El-Haija *et al.*, 2016)
-
- (Tang *et al.*, 2012)
- (Ng, 2015)

6.8 Scalabiltiy

6.8.1 Active/Online Learning

6.8.2 Output reduction

Chapter 7

YouTube-8m Challenge

- chapter describing methods and results for yt8m

7.1 Describe the challenge

7.2 Results

Chapter 8

Conclusion

- summary
- contributions
- reccomendations
- limitations
- future work

Appendices

Appendix A

Benchmark Datasets

- (Gibaja and Ventura, 2015)
- MULAN: http://meka.sourceforge.net/
- https://manikvarma.github.io/downloads/XC/XMLRepository.html
- yelp dataset: http://www.ics.uci.edu/~vpsaini/
- (Luaces *et al.*) argues that the MULAN repository does not have a wide variety and truly multi-label data sets. Most of the data sets have low cardinality and low label dependence. Thus these data sets may not show the true performance of multi-label algorithms. Therefore they created a multi-label data generator simulate multi-label data on which algorithms can be evaluated.

Appendix B

Software

Bibliography

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B. and Vijayanarasimhan, S. (2016). YouTube-8M: A Large-Scale Video Classification Benchmark. *arXiv*. 1609.08675.
Available at: https://arxiv.org/pdf/1609.08675.pdfhttp://arxiv.org/abs/1609.08675

Alazaidah, R. and Ahmad, F.K. (2016). Trending Challenges in Multi Label Classification. *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 7, no. 10.
Available at: www.ijacsa.thesai.org

Boutell, M.R., Luo, J., Shen, X. and Brown, C.M. (2004). Learning multi-label scene classiÿcation. *Pattern Recognition*, vol. 37, pp. 1757–1771.
Available at: www.elsevier.com/locate/patcog

Carvalho, André C P L F de, A.A.F. (2009). A Tutoria on Multi-Label Classification Techniques. *Foundations of Computational Intelligence*, vol. 5, pp. 177–195.
Available at: http://www.icmc.usp.br/{~}andrehttp://www.cs.kent.ac.uk/{~}aafhttp://www.cs.kent.ac.uk/people/staff/aaf/pub{_}papers.dir/Found-Comp-Intel-bk-ch-2009-Carvalho.pdf

Cherman, E.A., Monard, M.C. and Metz, J. (2011). Multi-label Problem Transformation Methods: a Case Study. *CLEI Electronic Journal*, vol. 14, no. 1, pp. 4–4. ISSN 0717-5000.

Dembcz, K., Waegeman, W., Cheng, W., Hüllermeier, E., Tsoumakas, G., Zhang, M.-L., Zhou, Z.-H., Dembczyski, K., Waegeman, W., Cheng, W. and Hüllermeier, E. (2012). On label dependence and loss minimization in multi-label classification. *Mach Learn*, vol. 88, pp. 5–45.

Dembczy, K. (2010). Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 279–286.
Available at: http://machinelearning.wustl.edu/mlpapers/paper{_}files/icml2010{_}DembczynskiCH10.pdfhttp://www.uni-marburg.de/fb12/kebi/people/cheng/cheng-icml10c.pdf

Dembczynski, K., Waegeman, W., Cheng, W. and Hüllermeier, E. (). On Label Dependence in Multi-Label Classification.

Dembczyński, K., Waegeman, W., Cheng, W. and Hüllermeier, E. (2010). Regret analysis for performance metrics in multi-label classification: The case of hamming and subset zero-one loss. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6321 LNAI, pp. 280–295. ISBN 364215879X. ISSN 03029743.
Available at: https://biblio.ugent.be/publication/1155381/file/1210780.pdf

Dembczynski, K., Waegeman, W. and Hüllermeier, E. (2012). An analysis of chaining in multi-label classification. In: *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 294–299. ISBN 9781614990970. ISSN 09226389.
Available at: https://biblio.ugent.be/publication/3132158/file/3132170

Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. Available at: https://pdfs.semanticscholar.org/e925/33e4adca54e31d7b3726088469b493c2282e.pdf?{_}ga=1.194163567.379343330.1490351020

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K., Fawcett Fürnkranz, T.J., Loza Mencía, E., Hüllermeier, E. and Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Mach Learn*, vol. 73, no. 73, pp. 133–153.
Available at: http://download.springer.com/static/pdf/878/art{%}253A10.1007{%}252Fs10994-008-5064-8.pdf?originUrl=http{%}3A{%}2F{%}2Flink.springer.com{%}2Farticle{%}2F10.1007{%}2Fs10994-008-5064-8{&}token2=exp=1490609445{~}acl={%}2Fstatic{%}2Fpdf{%}2F878{%}2Fart{%}25253A10.1007{%}25252Fs10994-008-506

Gao, W. and Zhou, Z.-H. (2011). On the Consistency of Multi-Label Learning. *Annals of Statistics*. ISSN 00905364. 1204.1688.

Gibaja, E. and Ventura, S. (2014). Multi-label learning: A review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 6, pp. 411–444. ISSN 19424795.

Gibaja, E. and Ventura, S. (2015). A Tutorial on Multilabel Learning. *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, pp. 52:1—-52:38. ISSN 0360-0300.
Available at: https://www.researchgate.net/profile/Sebastian{_}Ventura/publication/270337594{_}A{_}Tutorial{_}on{_}Multi-Label{_}Learning/links/54bcd8460cf253b50e2d697b.pdfhttp://doi.acm.org/10.1145/2716262

Godbole, S. and Sarawagi, S. (2004). Discriminative Methods for Multi-labeled Classification. *Lecture Notes in Computer Science*, vol. 3056, pp. 22–30. ISSN 03029743. 978-3-540-24775-3{_}5.
Available at: http://link.springer.com/10.1007/978-3-540-24775-3{_}5

Hastie, T., Tibshiranie, R. and Friedman, J.H. (2009). *No Title*. 2nd edn. New York: Springer.

Huang, S.-j., Yu, Y. and Zhou, Z.-h. (2012). Multi-label hypothesis reuse. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, p. 525.
Available at: http://dl.acm.org/citation.cfm?id=2339530.2339615

Huang, S.-J. and Zhou, Z.-H. (2012). Multi-Label Learning by Exploiting Label Correlations Locally. *AAAI Conference on Artificial Intelligence*, pp. 949–955. ISSN 9781577355687.

Koyejo, O.O., Natarajan, N., Ravikumar, P.K. and Dhillon, I.S. (2015). Consistent Multilabel Classification. *Advances in Neural Information Processing Systems*, pp. 3303–3311. ISSN 10495258.
Available at: http://papers.nips.cc/paper/5883-consistent-multilabel-classification

Lewis, D.D., Yang, Y., Rose, T.G., Li, F. and Lewis, F.L. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, vol. 5, pp. 361–397.
Available at: http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf

Lo, H.-Y., Lin, S.-D. and Wang, H.-M. (2013). Generalized k-Labelsets Ensemble for Multi-Label and Cost-Sensitive Classification. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, , no. X.

Luaces, O., Díez, J., Barranquero, J., Del Coz, J.J. and Bahamonde, A. (). Binary Relevance Efficacy for Multilabel Classification.

Madjarov, G., Kocev, D., Gjorgjevikj, D. and Džeroski, S. (2012). Author's personal copy An extensive experimental comparison of methods for multi-label learning.
Available at: http://www.elsevier.com/copyright

Ng, J.Y.-H. (2015). Beyond Short Snippets : Deep Networks for Video Classification. ISSN 10636919. arXiv:1503.08909v2.
Available at: https://pdfs.semanticscholar.org/57ed/4de2c8ea9c865dcf4273f0576eb746263475.pdf?{_}ga=1.106149697.379343330.1490351020

Pachet, F. and Roy, P. (2009). Improving Multilabel Analysis of Music Titles: A Large-Scale Validation of the Correction Approach. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 17, no. 2.

Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., Mei, T. and Zhang, H.-J. (2007). Correlative multi-label video annotation. In: *Proceedings of the 15th international conference on Multimedia - MULTIMEDIA '07*, p. 17. ISBN 9781595937025. ISSN 15516857.
Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.4803{&}rep=rep1{&}type=pdfhttp://portal.acm.org/citation.cfm?doid=1291233.1291245

Read, J. (2008). A pruned problem transformation method for multi-label classification. *New Zealand Computer Science Research Student Conference, NZCSRSC 2008 - Proceedings*, , no. April, pp. 143–150.

Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84880106608{&}partnerID=40{&}md5=ccdd19f7d0f111a2fdda8df7af0a8075

Read, J. and Hollmen, J. (2014). A Deep Interpretation of Classifier Chains. *Advances in Intelligent Data Analysis Xiii*, vol. 8819, pp. 251–262. ISSN 0302-9743.
Available at: http://jmread.github.io/papers/Read,Holmen-ADeepInterpretationofClassifierChains.pdf

Read, J. and Hollmén, J. (2015). Multi-label Classification using Labels as Hidden Nodes. pp. 1–23. 1503.09022.
Available at: https://arxiv.org/pdf/1503.09022.pdfhttp://arxiv.org/abs/1503.09022

Read, J., Pfahringer, B., Holmes, G. and Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, vol. 85, no. 3, pp. 333–359. ISSN 08856125. arXiv:1207.6324.

Read, J., Puurula, A. and Bifet, A. (2014). Multi-label Classification with Meta-Labels. *2014 IEEE International Conference on Data Mining*, pp. 941–946.
Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7023427

Schapire, R.E. and Singer, Y. (1999). Improved Boosting Algorithms Using Confidence-rated Predictions.

Schapire, R.E. and Singer, Y. (2000). BoosTexter: A boosting- based system for text categorization. *Machine learning*, vol. 39, no. 23, pp. 135–168.
Available at: http://link.springer.com/article/10.1023/A:1007649029923

Sechidis, K., Tsoumakas, G. and Vlahavas, I. (2011). On the Stratification of Multi-label Data.
Available at: http://download.springer.com/static/pdf/229/chp{%}253A10.1007{%}252F978-3-642-23808-6{_}10.pdf?originUrl=http{%}3A{%}2F{%}2Flink.springer.com{%}2Fchapter{%}2F10.1007{%}2F978-3-642-23808-6{_}10{&}token2=exp=1489589222{~}acl={%}2Fstatic{%}2Fpdf{%}2F229{%}2Fchp{%}25253A10.1007{%}25252F978-3-642-23808-6{_}10.pdf{%}3ForiginUrl{%}3Dhttp{%}253A{%}252F{%}252Flink.springer.com{%}252Fchapter{%}252F10.1007{%}252F978-3-642-23808-6{_}10*{~}hmac=dd714044d8c51b87977e0b8f0c77b39ac65ee0d36a75766e63ca2ac0bb53e89c

Sorower, M. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, pp. 1–25.
Available at: http://people.oregonstate.edu/{~}sorowerm/pdf/Qual-Multilabel-Shahed-CompleteVersion.pdf

Sucar, L.E., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H. and Larrañaga, P. (2013). Author's personal copy Multi-label classification with Bayesian network-based chain classifiers.

Systems, E. and Aviv-yafo, T. (2014). Ensemble Methods for Multi-label Classification Ensemble Methods for Multi-label Classification. , no. July 2013.

Tang, T.Y., Alhashmi, S.M. and Jaward, M.H. (2012). Hamming Selection Pruned Sets (HSPS) for Efficient Multi-label Video Classification. *PRICAI*, , no. 1. Available at: https://pdfs.semanticscholar.org/6d49/8611e1d30c5d1928573c199db02fc538d8d4.pdf?{_}ga=1.107724480.379343330.1490351020

Trohidis, K. and Kalliris, G. (2008). Multi-Label Classification of Music Into Emotions. *Proc. ISMIR*, vol. 2008, pp. 325–330.
Available at: http://ismir2008.ismir.net/papers/ISMIR2008{_}275.pdf

Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I. and Vlahavas, I. (2009). Correlation-based pruning of stacked binary relevance models for multi-label learning. *Proceedings of the Workshop on Learning from Multi-Label Data (MLD'09)*, pp. 101–116. ISSN 1475-925X.
Available at: http://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/learning-from-multi-label-data.pdf{#}page=102

Tsoumakas, G., Gr, G.A., Gr, E.A., Vilcek, J. and Gr, V.A. (2011). MULAN: A Java Library for Multi-Label Learning Eleftherios Spyromitros-Xioufis Ioannis Vlahavas. *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414.
Available at: http://www.jmlr.org/papers/volume12/tsoumakas11a/tsoumakas11a.pdf

Tsoumakas, G. and Katakis, I. (2007*a*). Multi-Label Classification. *Database Technologies*, vol. 3, no. 3, pp. 309–319.
Available at: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60566-058-5.ch021

Tsoumakas, G. and Katakis, I. (2007*b*). Multi-Label Classification : An Overview. ISSN 1548-3924.

Tsoumakas, G., Katakis, I. and Vlahavas, I. (2006). A Review of Multi-Label Classification Methods. *Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD 2006)*, pp. 99–109.

Tsoumakas, G. and Vlahavas, I. (). Random k-Labelsets: An Ensemble Method for Multilabel Classification.

Tsoumakas, G. and Vlahavas, I. (2007). Random k-labelsets: An Ensemble Method for Multilabel Classification. *European Conference on Machine Learning*, pp. 406–417. ISSN 01681605.
Available at: http://link.springer.com/chapter/10.1007/978-3-540-74958-5{_}38

Turnbull, D., Barrington, L., Torres, D. and Lanckriet, G. (2008). Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 16, no. 2.

Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.2154{&}rep=rep1{&}type=pdf

Vens, C., Struyf, J., Schietgat, L., Džeroski, S. and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, vol. 73, no. 2, pp. 185–214. ISSN 08856125.
Available at: https://lirias.kuleuven.be/bitstream/123456789/186698/4/hmc.pdf

Xu, C., Tao, D. and Xu, C. (2016). Robust Extreme Multi-Label Learning. *KDD*, pp. 421–434. ISSN 0146-4833. arXiv:1602.05561v1.

Zhang, M.-L. and Wu, L. (). LIFT: Multi-Label Learning with Label-Specific Features.

Zhang, M.-L. and Zhang, K. (2010). Multi-label learning by exploiting label dependency. *Kdd*, pp. 999–1007. ISSN 9781577355687.
Available at: http://dl.acm.org/citation.cfm?doid=1835804.1835930

Zhang, M.-L. and Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, vol. 40, pp. 2038–2048.
Available at: www.elsevier.com/locate/pr

Zhang, M.L. and Zhou, Z.H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837. ISSN 10414347.

Zhang, M.-l., Zhou, Z.-h. and Member, S. (2006). Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. vol. 18, no. 10, pp. 1338–1351.