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

List of Figures

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

$\overrightarrow{v}$	Physical vector, see equation ...

Subscripts

a	Adiabatic
a	Coordinate

Chapter 1

Introduction

1.1 Motivation

The motivation for this thesis is two-fold:

1. Multi-label learning is a highly relevant field in machine learning and statistics because of its wide range of applications. To varying degrees of success, it has been applied to problems in text categorisation, multimedia, biology, chemical data analysis, social network mining and e-learning among others (review list). Despite the rapid increase in multi-label learning literature (see Figure 1.1), the field is nowhere near the maturity level of its single-label counterpart. Consistently effective and efficient multi-label learning strategies are scarce. Researchers in the field have not yet reached consensus on many of the aspects when learning from multi-labelled data such as how to handle dependent labels or how to apply dimension reduction techniques. The field can gain from an up-to-date review of the literature (latest thorough review in 2014), more statistical perspectives on some of the challenges, additional benchmark datasets and quality empirical evaluations of the theory.

2. Deforestation is a massive global problem[1]. It contributes to reduced biodiversity, habitat loss, climate change and other devastating effects. It is said that the world loses an area of forest the size of 48 football fields per minute and the area most affected is in the Amazon basin (cite Kaggle). This problem can be fought more effectively by governments and local stakeholders if better data about the location of deforestation and human invasion on forests are continuously available to them - an ideal task for machine learning! Planet[2] and SCCON[3] constructed a dataset

[1] I saw in another paper that the rate of decline is decreasing (cite)

[2] Designer and builder of the world's largest constellation of Earth-imaging satellites - www.planet.com

[3] Remote sensing experts - www.sccon.com.br/eng

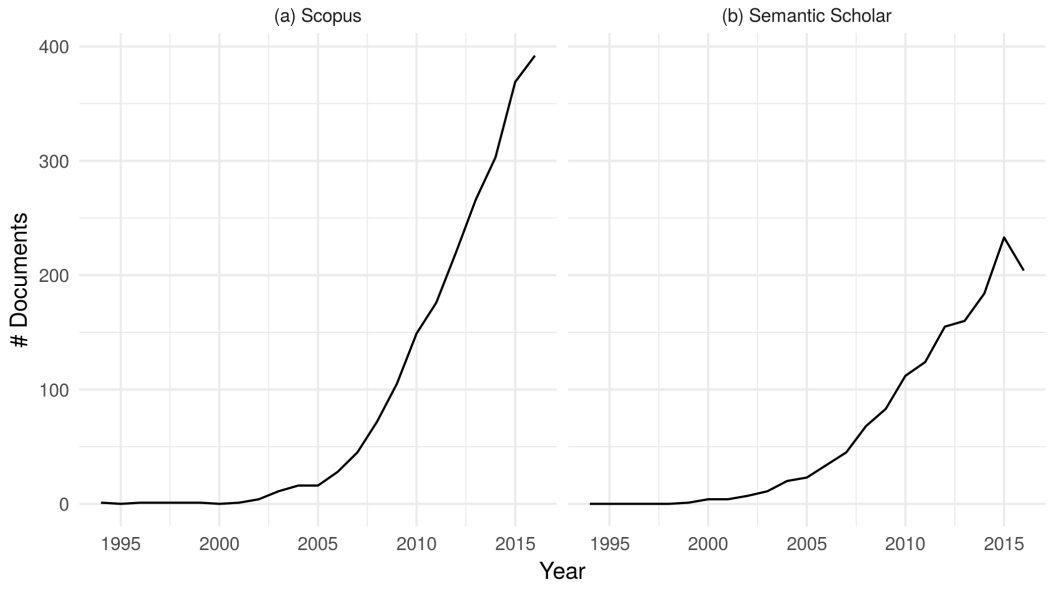

Figure 1.1: Line graphs illustrating the rise in multi-label learning publications per year for two databases. The database searches were done on 24-03-2017. The searches were not identical since they were limited to the search features of the databases. (a) The search on Scopus (cite) was for all documents (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. (b) The search on Semantic Scholar was 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.

of labelled satellite images taken of the Amazon basin and released it as part of a competition on Kaggle[4], challenging competitors to build algorithms that can automatically label these images with atmospheric conditions and various classes of land use/cover[5]. Resulting algorithms will help the global community better understand where, how, and why deforestation happens all over the world - and ultimately how to respond.

[4]Runs programming contests to crowd source machine learning solutions - www.kaggle.com

[5]Land cover indicates the physical land type such as forest or open water whereas land use documents how people are using the land.

1.2 Thesis Objectives

This thesis works towards building a multi-label learner that can label satellite images of the Amazon as accurately as possible. The method thought best to achieve this is to:

1. Identify the most important and latest developments in the multi-label literature, as well as in satellite image classification.
2. Provide an extensive review and discussion of these methods and how they compare to each other.
3. Empirically evaluate and compare them on the satellite image data in order to find the best strategies for our labeling task.

The main focus points for this thesis are:

- Label dependence - What is it; can it be used to improve a learner's accuracy and/or complexity; and when?
- Resampling - What are effective resampling techniques for multi-label data to deal with the class imbalance problem and to estimate errors and standard deviations?
- Dimension reduction - How to reduce the number of dimensions of the input and output space in order to build more effective and efficient algorithms.

  This is still a rough list and should be updated as progress is made with the following chapters.

  Make sure this is how an introduction is allowed to look.

  should I have a section on contributions?

1.3 Data

This section covers an initial introduction to the data. The elements of the data important to know before moving on will be discussed here and the rest will be addressed throughout the thesis, as it becomes relevant to the discussion.

1.3.1 Image Format

The data for this task comes from a set of images (also referred to as chips). Each chip is a small excerpt from a larger image of a specific scene in the Amazon taken by satellites. The chip size in pixels is 256×256, representing roughly 90 hectares of land, and is taken from a larger scene of 6600×2200 pixels. All of the satellite images were taken between January 1, 2016 and

February 1, 2017. The format of these images differ from the standard image format. Each image contains four bands of data: red (R), green (G), blue (B) and near infrared (NIR), where the standard format images usually only contain R, G and B. The additional NIR colour channel is common in remote sensing[6] applications and supposedly allows for clear distinction between water and vegetation in satellite images, for example.

Another difference between these images and the usual format is that these have pixel intensities in 16-bit digital number format as opposed to the usual 8-bit of standard RGB images. This allows the colours in the images to have a much higher range since 16-bit pixel intensities have 65536 (2^16) levels, compared to 256 levels of 8-bit images. This becomes useful, for example, to distinguish between different levels of darkness in an image. Thus each chip can be represented by a vector of size 262144 ($256 \times 256 \times 4$). This might prove to require to much computational power but strategies to reduce the size of such a vector exist, *e.g.* filtering or resizing of the image.

1.3.2 Collection and Labelling of the Images

The image collection was created by first specifying a "wish list" of scenes containing the phenomena the creators wanted to be included and also a rough estimate of the number of such scenes that are necessary for a sufficient representation in the final collection. This set of scenes was then searched for manually on Planet Explorer[7]. From these scenes the 4-band chips were created. The chips were labelled manually by crowd sourcing. The utmost care was taken to get a large and well-labelled dataset, but that does not mean the labels all correspond to the ground-truth, *i.e.* the data will contain some inherent error. The creators believe that the data has a reasonable high signal to noise ratio.

Note, the training and test splits was determined by the Kaggle competition creators. The training chips are labeled but the test chips are not. Predicted labels for the test chips can be submitted for Kaggle to evaluate in terms of an evaluation metric. This setup prevents competitors from using the test chips for training a classifier. There are 40479 training chips and 40669 test chips.

1.3.3 Class Labels

The class labels for the images can be broken into three groups: atmospheric conditions, common land cover/use phenomena and rare land cover/use phenomena. Each chip will have one atmospheric label and zero or more common

[6]The use of satellite- or aircraft-based sensor technologies to detect and classify objects on Earth [https://en.wikipedia.org/wiki/Remote_sensing].

[7]A web based interactive map of Earth consisting of satellite images, similar to Google Earth - www.planet.com/explorer

and rare labels. Chips that are labeled as cloudy should have no other labels, but there are some labeling errors.

The atmospheric condition labels are: *clear*, *haze*, *partly cloudy* and *cloudy*. They are relevant to a chip when:

- clear: there are no evidence of clouds.
- haze: clouds are visible but they are not so opaque as to obscure the ground.
- partly cloudy: scenes show opaque cloud cover over any portion of the image but the land cover/use phenomena are still visible.
- cloudy: 90% of the image is obscured with opaque cloud cover.

Examples of chips with atmospheric labels can be found in Figure 1.2. Each chip should only have one atmospheric label and therefore this classifying task simplifies to a multiclass problem. This allows for the option to break up the labeling task of all the labels into two tasks: a multiclass classification problem for the atmospheric labels and a multi-label classification problem for the land cover/use labels. This approach might save some computational time and give extra information to the multi-label learners for classifying the land cover/use labels. We will experiment with these approaches in Chapter ??.

The common land cover/use labels are: *primary*, *agriculture*, *water*, *habitation*, *road*, *cultivation* and *bare ground*. They are relevant to a chip when:

- primary: it is primarily consisting of rain forest (virgin forest), *i.e.* dense tree cover.
- agriculture: it contains any land cleared of trees that is being used for agriculture or range land.
- water: it contains any one of the following: rivers, reservoirs, or oxbow lakes.
- habitation: it contains human homes or buildings.
- road: it contains any type of road.
- cultivation: it shows signs of smaller-scale/informally cleared land for farming.
- bare ground: it contains naturally (not the caused by humans) occurring tree-free areas.

Examples of chips with common land cover/use labels are found in Figure 1.3. According to the competition page on Kaggle, small, single-dwelling habitations are often difficult to spot but usually appear as clumps of a few pixels that are bright white. Roads sometimes look very similar to rivers and therefore these two labels might be noisy. The NIR band might give a classifier additional information to help distinguish between the two. Cultivation is a subset of agriculture and is normally found near smaller villages, along major rivers or at the outskirts of agricultural areas. It typically covers very small areas.

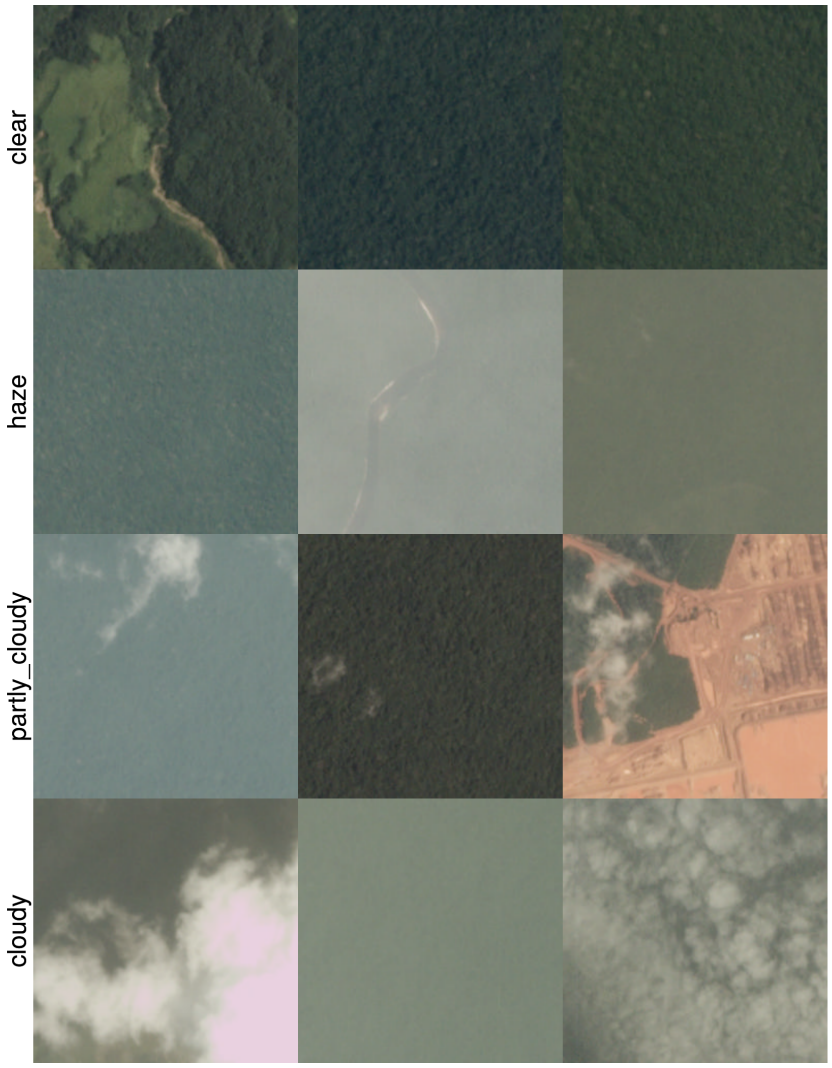

Figure 1.2: Examples of chips with atmospheric labels. These (along with all the other chips plotted throughout the thesis) are the JPEG conversions of the original 4-band, 16-bit images.

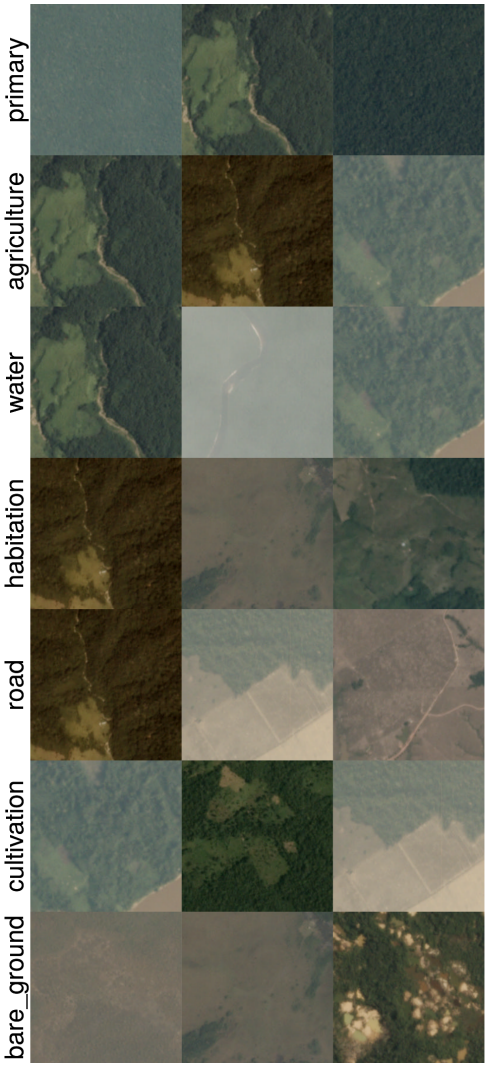

Figure 1.3: Examples of chips with common land cover/use labels.

The less common land cover/use labels are: *slash and burn*, *selective logging*, *blooming*, *conventional mine*, *artisinal mine* and *blow down*. Chips are tagged with these labels when:

- slash and burn: there are signs of the farming method that involves the cutting and burning of the forest to create a field. These look like cultivation patches with black or dark brown areas.
- selective logging: winding dirt roads are present adjacent to bare brown patches in otherwise primary rain forest. Selective logging is the practice of selectively removing high values tree species from the rainforest.
- blooming: there are signs of trees flowering. Blooming is a natural phenomena where particular species of flowering trees bloom, fruit and flower at the same time. These trees are quite big and the phenomena

can be seen in the chips. They usually appear as white dots.

- conventional mine: it contains signs of large-scale legal mining operations.
- artisinal mine: it contains signs of small-scale (sometimes illegal) mining operations.
- blow down: there are signs of trees uprooted or broken by wind. High speed winds (~160km/h) in the Amazon are generated when the cold dry air from the Andes settles on top of the warm moist air in the rainforest and then sinks down with incredible force, toppling larger rainforest trees. These open areas are visible from space.

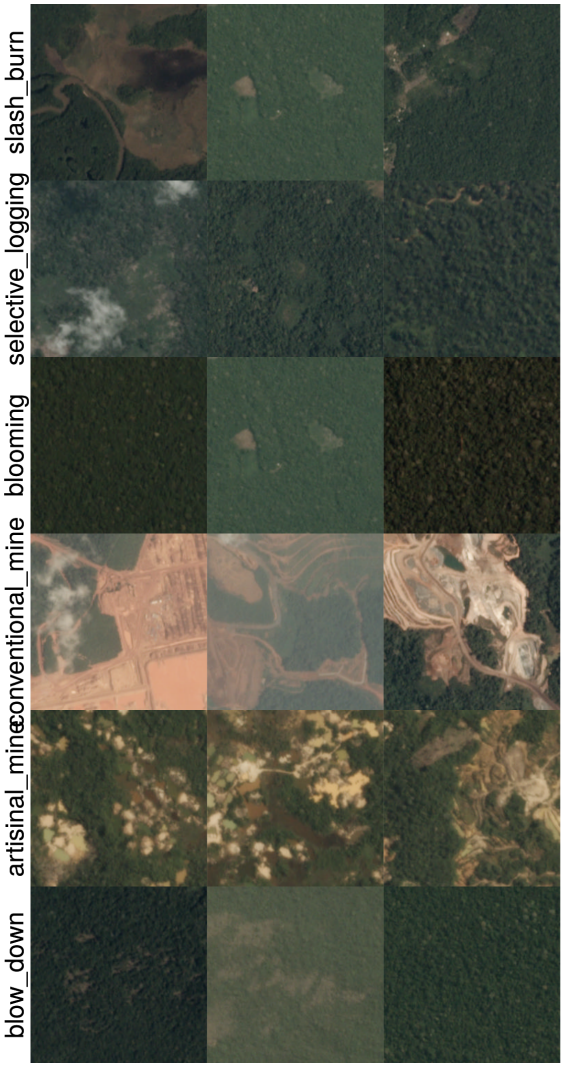

Figure 1.4: Examples of chips with less common land cover/use labels.

Examples of chips with these less common land cover/use labels are given in Figure 1.4. These labels are more challenging to identify in the chips and

since they also appear less frequently, it might be difficult for the classifier to learn these labels.

1.4 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, why it is important, etc.

1.5 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.

1.6 Outline

The structure of this thesis is built to mimic the workflow of fitting a supervised learning model to data. At each step, the relevant literature will be critically reviewed and discussed. Thereafter the proposed and recommended strategies will be applied to our data to see if the results match the literature and to find the best methods for our application.

In any supervised learning problem, it is essential to become familiar with the data and the task at hand before moving on to the training process. The background information of the data has already been discussed. In Chapter ?? the unique proppertes of multi-label data will be investigated and what steps are recommended to follow for data with certain properties. It is very important to clearly define the objective of the supervised learning task. For this thesis, prediction accuracy is more important than making inferences on the data (models also giving insight into the data is a bonus). The evaluation metric for our task will be introduced and discussed in Chapter ?? along with other ways to evaluate multi-label classifiers.

Other things still to mention:

- basic ML strategies
- ML resampling strategies for class imbalance and error estimates
- orders of complexity
- label dependence
- input space reduction

- output space reduction
- final predictions and evaluations (maybe MDS of actual vs predicted)
- might want to include short history/timeline of ML
- might want to do a meta analysis of the literature on main topics

Chapter 2

The Multi-Label Framework

The new plan for this chapter is to introduce all the ml concepts necessary to understand before fitting the model. First we need to define the ML learning objective. Then we will look at ML data set properties. Important: class imbalance, label correlation (should I talk about exploiting it here?). Then we will discuss evaluation metrics. This chapter might become very long.

2.1 Introduction

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 data set 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 $\mathcal{L} = \{l_1, l_2, \ldots, l_K\}$ 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 $L \subseteq \mathcal{L}$ 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.

As should be expected, the ML framework has a few concepts novel to the single-label case which should be reviewed before looking at the algorithms for ML learning. In this chapter, the core notation for the thesis will be introduced and a clear definition of the task of ML learning will be given. Then, a deep look is taken into the unique properties of ML data and how these might affect the performance of classifiers. The concepts of label correlation and class imbalance will also be introduced, however, how to deal with these will be discussed in the next chapter (for now). Finally, we will get to the evaluation metrics of ML algorithms. This is an important topic in ML learning, often neglected in the literature [cite]. After completing this chapter, the reader will have a good basis to be able to move on to the discussion of ML learning algorithms.

2.2 Notation

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 or output matrix as

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

where K is the size of the label set $\mathcal{L}$. Y only contains zeros and ones, *i.e.* $y_{ik} = 1$ if label l_k, $k = 1, \dots, K$, 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 data set will be defined as $D = \begin{bmatrix} X & Y \end{bmatrix}$, which contains the n input-output pairs, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \,|i = 1, \dots, n\}$. Note that, $\boldsymbol{y}_i = (y_1, y_2, \dots, y_K)$, $y_k \in \{0, 1\}$, used here is the label vector, however, it is also common to use the label set notation, *i.e.* $L_i \subseteq \mathcal{L}$, where $\mathcal{L}$ is the complete label set and L_i is the set of relevant labels for observation i.

2.3 The Task of Multi-Label Learning

A more formal definition of the ML learning task will be given in the following chapter. However, it is important to note that we will define the utlimate task of ML learning as the 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. (see (Zhang and Zhou, 2014) for a more brief description)

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

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., $h : X \to Y$, 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 $f : X \to G$, 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?)

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.

2.4 Multi-Label Indicators

As with all supervised learning problems, no one ML algorithm performs optimally on all problems. It is common practice in classical single output supervised learning to first consider, for example, the number of features (p) and the number of observations (n) in a data set before deciding on which model(s) to fit to the data. The same naturally holds for a ML problem but with added complexity. The multiple outputs of the data introduces many more factors to consider before continuing to the modelling phase. Some ML data sets have only a few labels per observation, while others have plenty. In some ML data sets the number of label combinations is small, whereas in others it can be very large. Some labels appear more frequently than others. Moreover, the labels can be correlated or not. These characteristics can have a serious impact on the performance of a ML classifier. This is the reason why several specific indicators have been designed to assess ML data set properties.

The two standard measures for the multi-labeledness of a data set are *label cardinality* and *label density*, introduced by (Tsoumakas and Katakis). The label cardinality of a ML data, D, set is the average number of labels per observation:

$$LCard(D) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}.$$

This measure can be normalised to be independent of the label set size, which results in the label density indicator:

$$LDens(D) = \frac{1}{K}LCard(D) = \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}.$$

According to (Tsoumakas and Katakis) it is important to distinguish between these two measures, since two data sets with the same label cardinality but with a great difference in the number of labels might not exhibit the same properties and cause different behaviour to the ML classification methods. These two measures give a good indication of the label frequency of a data set, but we are also interested in the uniformity and regularity of the labeling scheme. The authors of (Read *et al.*, 2011*b*) suggested measuring the proportion of distinct label sets and the proportion of label sets with the maximum frequency. Consider the number of distinct label sets, also referred to as the label diversity (Zhang and Zhou, 2014), which can be defined as:

> there are multiple ways this is defined in the literature - still need to decide on which one I want to use

$$LDiv(D) = |\{Y|\exists \boldsymbol{x} : (\boldsymbol{x}, Y) \in D\}|,$$

by (Zhang and Zhou, 2014). ((Read *et al.*, 2011*b*) uses $\exists!$ instead of $\exists$ and Y as a vector $\boldsymbol{y}$. I want to consider a way of defining it in matrix notation. Maybe with an indicator function. Some papers define it as DL instead of $LDiv$.) The proportion of distinct label sets in D is then

$$PLDiv\{/PUniq/PDL\}(D) = \frac{1}{n}LDiv(D).$$

The proportion of label sets with the maximum frequency is defined by (Read *et al.*, 2011*b*) as:

$$PMax(D) = \max_{\boldsymbol{y}} \frac{\text{count}(\boldsymbol{y}, D)}{n},$$

where $\text{count}(\boldsymbol{y}, D)$ is the frequency that label combination $\boldsymbol{y}$ is found in data set D. This represents the proportion of observations associated with the most frequently occurring label sets. High values of $PLDiv$ and $PMax$ indicate an irregular and skewed labeling scheme, respectively, *i.e.* a relatively high number of observations are associated with infrequent label sets and a relatively high number of observations are associated with the most common label sets. (*think about this again*) When this is the case, and the labels are modelled separately,

the classifiers will suffer from the class imbalance problem, a common problem in supervised classification tasks. More detail about this will be adressed shortly.

Very little research has been done on how all these ML indicators affect the performance of a ML classifier. (Chekina *et al.*, 2011) made a worthy attempt. Their goal was to find a way of determining which ML algorithm to use given a data set with specific properties and with a specific evaluation metric to optimise. They approached this problem by training a so called meta-learner on a meta-data set containing the performance of multiple ML algorithms on benchmark data sets with different properties. This trained meta-learner is then able to predict which ML algorithm is most likely to give the best results in terms of a specific evaluation metric, given the properties of the data set to be analysed. Although we will not use their meta-learner for this thesis, we will consider some of the additional findings in their research. They found that the following properties (among others) of a ML data set was important to their trained meta-model (which was based on classification trees) in predicting which ML algorithm is most appropriate: K; $LDiv(D)$; $LCard(D)$; the standard deviation, skewness and kurtosis of the number of labels per observation in D; number of unconditionally dependent label pairs (based on what?); average of χ^2 -scores of all dependent label pairs; number of classes with less than 2, 5 and 10 observations; ratio of classes with less than 2, 5, 10 and 50 observations; average, minimal and maximal entropy of labels (def of entropy?); average observations per class. This strengthens the argument that it is important to take ML indicators into account before the training process.

Some rules that they found that I might refer to later:

- for micro-AUC target evaluation measure if label cardinality of training data is above 3.028 then the 2BR method (among the single-classifiers) should be used.
- Another example for an extracted rule is for ranking loss evaluation measure: if minimum of label entropies is zero (i.e. there is at least one certain label in the training set), number of labels is less than 53 and skewness of label cardinality is below or equal to 2.49 then the EPS method (among ensembles) should be used.

2.5 Benchmark Data Sets

The progress of areas in machine/statistical learning is highly dependent on the availability of quality and diverse benchmark data sets. This enables researchers to compare their methods in a wide variety of environments. Recently, a decent amount of ML data sets has been published, but not without critique. (Luaces

et al.) argues that the MULAN[1] ML data set repository does not have data sets that are truely ML and that most of the data sets are very similar to each other. Most of the data sets have low cardinality and low label dependence. The problem with this is that these data sets may not show the true performance of ML algorithms. In (Gibaja and Ventura, 2015*a*) the authors also comments on the lack of thorough, comparative empirical studies on these benchmark sets.

Some of the most popular and recent ML benchmark data sets will be introduced here along with their properties. This will give us some form of a reference to compare our data set of satellite images against.

- (Read *et al.*, 2011*b*) defines a complexity measure as $n \times p \times K$
- (Gibaja and Ventura, 2015*b*) long list of datasets. Other than MULAN: Plant and Human, Slashdot, LangLog, IMDB
- (Sorower)
- https://manikvarma.github.io/downloads/XC/XMLRepository.html
- yelp dataset: http://www.ics.uci.edu/~vpsaini/
- also new yt8m

2.6 Sampling and Resampling

- Simulating (Tomás *et al.*, 2014) (also gives citations to other papers)
- partitioning mentioned in (Gibaja and Ventura, 2015*a*) - referred to (Sechidis *et al.*, 2011)
- (Luaces *et al.*) Therefore they created a ML data generator to simulate ML data on which algorithms can be evaluated.

2.7 Class Imbalance

- (Charte *et al.*, 2015)

Maybe include the following headers here:

2.8 Learning Objective

2.9 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

[1] A Java library for ML learning - http://mulan.sourceforge.net/datasets-mlc.html.

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.

2.9.1 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 2.1 is just an example. The image quality is lacking.

2.9.2 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

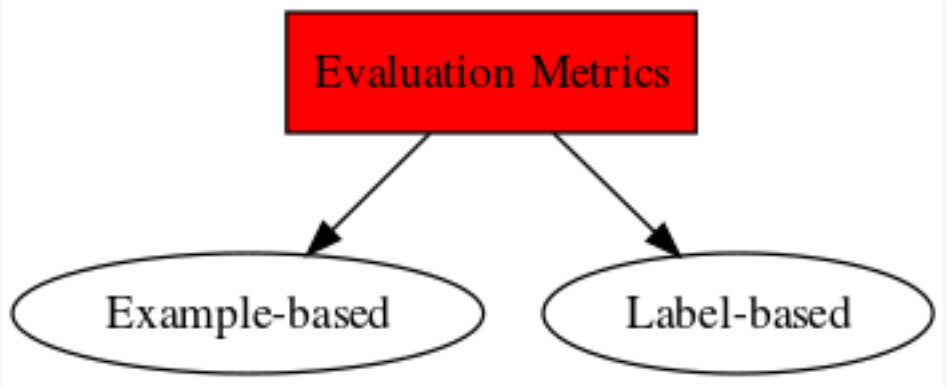

Figure 2.1: Categorisation of the taxonomy of MLL evaluation metrics

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:

2.9.3 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).

2.9.4 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.

2.10 Label Dependence

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*a*), (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

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 $k = 1, 2, \ldots, K$. $h_k : \boldsymbol{X} \to \{0, 1\}$ 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:

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

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:

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

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:

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

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:

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

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 2ˆ{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 $Y_1, \ldots, Y_K$ 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 $\boldsymbol{x}^{(n)} = [x_1^{(n)}, \ldots, x_p^{(n)}]$, where $x_j \in \mathcal{R}$, $j = 1, \ldots, p$.
- 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 $\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_K] = h(\boldsymbol{x})$.
- 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

$$\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_K] = h(\tilde{\boldsymbol{x}}) = \arg \max_{\boldsymbol{y} \in \{0,1\}^p} p(\boldsymbol{y}|\tilde{\boldsymbol{x}})$$

  This corresponds to minimizing the susbet 0/1 loss.

- $h_{BR}(\tilde{\boldsymbol{x}}) := [h_1(\tilde{\boldsymbol{x}}), \ldots, h_K(\tilde{\boldsymbol{x}})]$
- 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:

$$h_{CC}(\tilde{\boldsymbol{x}}) := [h_1(\tilde{\boldsymbol{x}}), h_2(\tilde{\boldsymbol{x}}, h_1(\tilde{\boldsymbol{x}})), \ldots, h_K(\tilde{\boldsymbol{x}}, \ldots, h_{K-1}(\tilde{\boldsymbol{x}}))]$$

- PCC formulates CC as the joint distribution using the chain rule,

$$h_{CC}(\boldsymbol{x}) := \arg\max_{\boldsymbol{y}} p(y_1|\boldsymbol{x}) \prod_{k=2}^{K} p(y_k|\boldsymbol{x}, y_1, \ldots, y_{K-1})$$

  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).

From the Bayesian point of view, the problem of multi-label learning can be reduced to modeling the conditional joint distribution of $P(\boldsymbol{y}|\boldsymbol{x})$. This can be done in various ways. First-order approaches solve the problem by decomposing it into a number of independent task through modelling $P(y_k|\boldsymbol{x})$, $k = 1, \ldots, K$. Second-order approaches solve the problem by considering interactions between a pair of labels through modelling $P((y_k, y_{k'})|\boldsymbol{x})$, $k \neq k'$. High-order approaches solve the problem by adressing correlations between a subset of labels through modelling $P((y_{k_1}, y_{k_2}, \ldots, y_{k_{K'}})|\boldsymbol{x})$, $K' \leq K$. Our goal is to find a simple and efficient way to improve the performance of multi-label learning by exploiting the label dependencies (Zhang and Zhang, 2010). Propose LEAD approach.

- (Tsoumakas *et al.*, 2009) use the ϕ coefficient to estimate label correlations.

3.9 Exploiting Label Dependence

- (Sorower)
- mention the holy grail comment
- comment on what 'exploitation' means. Since many authors claim that exploiting label dependence structures is the only way to effectively handle multiple labels, I would assume this means that we can make use of label correlations to spare time and increase accuracy.
- we need to think about how observations are labelled, when will it be useful to take label dependence into account and how.
- Such a solution, however, ne- glects the fact that information of one label may be help- ful for the learning of another related label; especially when some labels have insufficient training examples, the label correlations may provide helpful extra information (Huang and Zhou, 2012)

3.10 Theoretical Results

3.10.1 Intuitive Perspective

3.10.2 Two Types of Label Dependence

- estimating label correlations are difficult when we assume that some labels are missing (Zhu *et al.*, 2017)

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

- is local the same as conditional? and global unconditional?
- (Zhu *et al.*, 2017)

Existing approaches to exploiting label correlations either assume the the label correlations are global and shared by all instances, or that the label correlations are local and shared only by a subset of the data. It may be that some label correlations are globally applicable and some share only in a local group of observations.

- give example
- mention GLOCAL (Zhu *et al.*, 2017)
- (Huang and Zhou, 2012)

Existing approaches typically exploit label correlations globally by assuming that the label correlations are shared by all observations. In the real-world, however, different observations may share different label correlations and few correlations are globally applicable.

- propose ML-LOC approach
- mentions that by assuming global correlations may be hurtful to the performance (Huang and Zhou, 2012) in empirical discussion

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)
- few results for ML FS (Spolaôr *et al.*, 2013)

4.1.1 Single-Label Framework

4.2 Feature Selection

- Feature Selection (FS) plays an important role in machine learning and data mining, and it is often applied as a data pre-processing step. FS aims to find a small number of features that describes the dataset as well as the original set of features does [14], providing support to tackle the "curse of dimensionality" problem when learning from high-dimensional data. Feature selection can effectively reduce data dimen- sionality by removing irrelevant and/or redundant features, speeding up learning algorithms and sometimes improving their performance. In fact, various studies show that features can be removed without performance deterioration. (Spolaôr *et al.*, 2013)
- individual evaluation vs subset evaluation (Spolaôr *et al.*, 2013)
- standard to transform into single lable problems and then selecting features

4.2.1 Filter Approaches

- The filter approach filters out irrelevant features independently of the learning algorithm
- may not choose the best features for specific learning algorithms
- fast and simple

- most frequently used
- several feature importance measures
- Information gain; ReliefF, fisher score, Gini index, CFS, Rough set

4.2.2 Wrapper Approaches

- The wrapper approach requires a specific learning algorithm to evaluate and to determine which features are selected.
- high computational cost

4.2.3 Embedded Approaches

- incorporate feature selection as part of the training process

4.3 Feature Extraction

4.4 Meta-analysis

- see (Spolaôr *et al.*, 2016) for systematic review

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

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

Charte, F., Rivera, A.J., del Jesus, M.J. and Herrera, F. (2015). Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, vol. 163, pp. 1–14. ISSN 09252312.
Available at: http://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/1790{_}2015-Neuro-Charte-MultiLabel{_}Imbalanced.pdfhttp://linkinghub.elsevier.com/retrieve/pii/S0925231215004269

Chekina, L., Rokach, L. and Shapira, B. (2011). Meta-learning for selecting a multi-label classification algorithm. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 220–227. ISBN 9780769544090. ISSN 15504786.

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

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*). 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

Gibaja, E. and Ventura, S. (2015*b*). 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

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

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

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. 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*a*). Classifier chains for multi-label classification. *Machine Learning*, vol. 85, no. 3, pp. 333–359. ISSN 08856125. arXiv:1207.6324.

Read, J., Pfahringer, B., Holmes, G., Frank, E., Brodley Read, C.J., Pfahringer, B., Holmes, G. and Frank, E. (2011*b*). Classifier chains for multi-label classification. *Mach Learn*, vol. 85, no. 85.
Available at: http://download.springer.com/static/pdf/44/art{%}253A10.1007{%}252Fs10994-011-5256-5.pdf?originUrl=http{%}3A{%}2F{%}2Flink.springer.com{%}2Farticle{%}2F10.1007{%}2Fs10994-011-5256-5{&}token2=exp=1490608886{~}acl={%}2Fstatic{%}2Fpdf{%}2F44{%}2Fart{%}25253A10.1007{%}25252Fs10994-011-5256-

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{%}25252F978-3-64

Sorower, M.S. (). A Literature Survey on Algorithms for Multi-label Learning.

Spolaôr, N., Cherman, E.A., Monard, M.C. and Lee, H.D. (2013). A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135–151. ISSN 15710661.

Spolaôr, N., Monard, M.C., Tsoumakas, G. and Lee, H.D. (2016). A systematic review of multi-label feature selection and a new method based on label construction. *Neurocomputing*, vol. 180, pp. 3–15. ISSN 18728286.
Available at: http://dx.doi.org/10.1016/j.neucom.2015.07.118

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

Tomás, J.T., Spolaôr, N., Cherman, E.A. and Monard, M.C. (2014). A framework to generate synthetic multi-label datasets. *Electronic Notes in Theoretical Computer Science*, vol. 302, pp. 155–176. ISSN 15710661.
Available at: http://ac.els-cdn.com/S1571066114000267/1-s2.0-S1571066114000267-main.pdf?{_}tid=207a475a-25c4-11e7-9d47-00000aacb35d{&}acdnat=1492691174{_}037266698571d8a927f3feb0eb432995

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://lpis.csd.auth.gr/publications/tsoumakas-mld09.pdfhttp://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/learning-from-multi-label-data.pdf{#}page=102

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

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

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

Zhu, Y., Kwok, J.T. and Zhou, Z.-H. (2017). Multi-Label Learning with Global and Local Label Correlation. 1704.01415.
Available at: https://arxiv.org/pdf/1704.01415.pdfhttp://arxiv.org/abs/1704.01415