وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار– عنابـــة

Faculté des Sciences de L'Ingéniorat **Année 2017**

Département d'Informatique

THESE

Présentée en vue de l'obtention
du diplôme de Doctorat en Sciences

Semi-Supervised Multi-Label Feature Selection

Option
Intelligence Artificielle

par
Mr Abdelouahid ALALGA

DEVANT LE JURY

Président :		
Mme Labiba SOUICI-MESLATI	Professeur	Université Badji Mokhtar-Annaba
Directeur de Thèse		
Mme Nora TALEB	MCA	Université Badji Mokhtar-Annaba
Co-Directeur de Thèse		
Mr Khalid BENABDESLEM	MC, HDR	Université Claude Bernard, Lyon, France
Examinateurs :		
Mr Abdelkrim AMIRAT	Professeur	Université Med Cherif Messaadia-Souk Ahras
Mr Yacine LAFIFI	Professeur	Université 8 Mai 1945- Guelma

To those dear to me

"I shall tell you a great secret, my friend. Do not wait for the last judgment, it takes place every day."

Albert Camus

Preface

Acknowledgment

First and foremost, I would like to thank all those who worked so hard to set me obstacle in the road, they showed how far my limits could be pushed and helped me surpass them, without them this work would not have been accomplished. Their efforts have reached far beyond this work.

I would like to convey my utmost gratitude to my co-supervisor, mentor and friend Dr. Khalid Benabdeslem, Associate Professor at University of Lyon, for his helpful advice and continued support during my research and writing up. His broad knowledge in the field and his enthusiasm have been of great help to my research study. I am indebted to him for supporting me through my tough times. From the bottom of my heart, I thank him for hosting my internship and his warm welcome at LIRIS. Thanks again for the time we spent talking about every thing and nothing near the coffee machine, his down-to-earth attitude serves as a role model for me.

My deep feelings of gratefulness and respect are to Dr. Nora Taleb, Associate Professor at University of Annaba, this work would not have been possible without her unconditional help, encouragements and support. I would also like to express my profound thanks to the members of the Jury. Heartfelt gratitude to Pr. Labiba Souici, Professor at University of Annaba, to have honored me by presiding the jury. I owe my thanks to the Pr. Yacine Lafifi, Professor at University of Guelma, for accepting to take part of the jury as a reviewer. Special thanks go to Pr. Abdelkrim Amirat, Professor at University of Souk-Ahras, for reviewing this thesis.

I would also like to extend thanks to my friends, colleagues and anyone who

has contributed with their thoughts and opinions in successfully completing this project.

Last but not least, I would like to acknowledge from the bottom of my heart the endless love and support of my family, specially: my parents, my brothers and sisters. They all kept me going and without them I would not be capable of anything. All I have ever accomplished and I will ever be, I owe to them.

Abstract

NEVER before has there been a time where data is as easy to collect and record as it is now. With the rapid development of digital technologies, data has become both such a precious commodity and a very abundant one. With such abundance, however, come issues related to the quality of the data. Noisy, meaningless data has become the bottleneck of machine learning. The most likely source of this noise is due to irrelevant and redundant features. In this regard, feature selection, usually used as a preprocessing step, has become the mainstay of machine learning, especially when dealing with large-scale data. However, feature selection is continuously challenged by new emerging issues. Recently, different domain applications suggest data consisting of huge amount of unlabeled instances together with a small number of multi-labeled instances, which we refer to as semi-supervised multi-label learning. On the one hand, multi-labeled data arises in domains where instances are often classified into more than one category. On the other hand, difficulties to obtain such annotated data make it scarce. Multi-label data are hard and coslty to obtain, whereas unlabeled data is abundant. The abundance of this cheap and readily available unlabeled data can make up for the paucity of the annotated ones. Similar to single semi-supervised feature selection, its multi-label counterpart tends to capitalize on unlabeled data to fill the chasm between supervised and unsupervised feature selection.

In the same vein, seeking to address both issues, the present work deals with semi-supervised multi-label feature selection. Based on the premise of data similarity and the locality of data which we capture through the usage of spectral graph theory, we devised various solutions. More specifically, we firstly study and propose a new filter feature selection framework, that filter out, irrelevant uninformative features, *i.e.* features less related to the target concept (encoded by class label information and the geometric structure of the data). We present two transformation-based algorithms plus an algorithm developed based on algorithm adaptation, which is the main contribution of this work (named S-CLS). In addition, we also study the effect of variance on S-CLS and propose a three-fold ensemble learning solution. Finally, we demonstrate the benefits and advantages of our frameworks through various experimental studies. The experiments were carried out on a score of benchmark data sets from various application fields, and

results were promising, showing that our methods either outperform state-of-the-art algorithms or in the worst cases give comparable results.

Keywords: Feature selection, constraints, semi-supervised learning, multi-label classification, spectral graph theory, ensemble methods.

Résumé

Il n'y avait jamais eu un temps où les données sont aussi faciles à collecter et à enregistrer que maintenant. Avec l'avenement et le développement rapide des technologies numériques, les données sont devenues à la fois un bien précieux et très abondant. Cependant, avec une telle profusion, se posent des questions relatives à la qualité de ces données. Des données bruitées et sans un quelconque apport informationnel sont devenues le goulot d'étranglement de l'apprentissage automatique. La plus probable source de bruit est souvent due à des variables non pertinentes et/ou redondantes. À cet égard, la sélection de variables (caractéristiques), habituellement utilisée comme une étape de prétraitement, est devenue le pilier de l'apprentissage automatique, notamment lorsqu'il s'agit de données à grande échelle.

Néanmoins, cette sélection est constamment mise à l'épreuve par de nouveaux problèmes émergents. Récemment, diverses applications du monde réel ont mis à disposition des données constituées d'une large quantité d'instances non etiquetées ainsi qu'un nombre trés réduit d'instances multi-étiquettes, ce qui a donné naissance à l'apprentissage semi-supervisé multi-label. D'une part, les données multi-étiquettes apparaissent dans des domaines où les objets sont souvent classés dans plusieurs catégories à la fois sans aucune contrainte d'exclusion mutuelle. D'autre part, les difficultés de produire des données avec une telle annotation empêchent leur collecte en quantité suffisante pour qu'il y ait un apprentissage fiable. En effect, il est fastidieux et coûteux d'obtenir des données multi-étiquettes, alors que les données non étiquetées sont abondantes. Cette abondance peut compenser la pénurie des données multi-labels. D'une façon similaire à la selection semi-supervisée et standard de variables (mono-label), son homologue multi-label s'efforce de capitaliser sur les données non étiquetées pour combler l'abîme entre la sélection supervisée et non-superviseé.

Dans le même ordre d'idées, cherchant à concilier ces deux axes de recherche, cette thèse portent sur la sélection de variables dans un contexte multi-labels en mode semi-supervisé. En exploitant la prémisse de la similarité entre les données et de leur structure locale que nous captons grâce à l'utilisation de la théorie spectrale des graphes, nous proposons diverses solutions. Tout d'abord, un nouveau cadre de travail mettant en avant des méthodes "filtre" pour sélectionner les vari-

ables pertinentes. Plus particulièrement, nous présentons deux algorithmes basés sur la transformation de problèmes et surtout un autre algorithme développé en adoptant l'approche d'adaptation d'algorithmes. En outre, nous examinons l'effet de la diversité et proposons une solution de selection basée sur l'apprentissage ensembliste. Enfin, à travers de nombreuses expérimentations nous démontrerons l'efficacité des algorithmes proposés.

Mots-clés: sélection de variables, contraintes, apprentissage semi-supervisé, classification multi-label, théorie spectrale de graphe, méthodes ensemblistes.

ملخص

لم يحدث من قبل أن كانت البيانات أسهل للجمع والحفظ كما هو عليه الحال الآن. مع التطور السريع للتكنولوجيات الرقمية، أصبحت البيانات سلعة ثمينة و وفيرة جدا على حد سواء. و لكن، مع هذه الوفرة تأتي تحديات متعلقة بالجودة. حيث يمكن ان تقلص بعض الشوائب من القيمة المعرفية لتلك البيانات، وهكذا أصبحت البيانات التي لا معنى لها عنق الزجاجة بالنسبة لتعلم الآلة. ويرجع السبب الأكثر ترجيحا لهذه الشوائب إلى وجود متغيرات زائدة، غير مناسة أو مكررة. وفي هذا الصدد، تعتبر عملية اختيار المتغيرات المناسبة، و التى عادة ما تستخدم كخطوة معالجة مسبقة للبيانات، الدعامة الاساسية لتعلم الألة، لا سيما عندما يتعلق الامر بمعالجة بيانات ضخمة. بالإضافة، يواجه تعلم الآلة باستمرار بتحديات و قضايا جديدة. في الآونة الأخيرة مثلا، انتجت تطبيقات مختلفة بيانات متكونة من كمية ضخمة من المشاهدات الغير معلمة بجانب كمية محدودة جدا من المشاهدات المتعددة العلامات مما أدى إلى نشأة اختصاص جديد في مجال تعلم الآلة يطلق عليه تعلم متعدد العلامات. للأسف، مثل هذه البيانات المتعددة العلامات مكلفة و من صعب الحصول عليها، في حين أن البيانات الغير معلمة متوفرة بغزارة و سهلة المنال. في هذا الصدد، يمكن استخدام هذه الاخيرة لتعويض ندرة البيانات المعلمة. على هذا المنوال و على غرار التعلم النصف خاضع للإشراف التقليدي، يميل نظيره المتعدد العلامات إلى الاستفادة من البيانات غير المعلمة لملء الهوة بين عملية اختيار المتغيرات الخاضع للإشراف وغير الخاضع للإشراف.

في هذا الاطار، نتناول في هذه الأطروحة عملية اختيار المتغيرات في نطاق تعلم الآلة المتعدد العلامات و نصف الخاضع للإشراف. حيث نقترح حلول مختلفة بالإعتماد على نظرية الرسم البياني الطيفي.

كلمات مفتاحية :

اختيار المتغيرات، تعلم متعدد العلامات، بيانات شبه معلمة، قيود ثنائية، نظرية الرسم البياني الطيفي .

Contents

Preface **i**

Acknowledgment i

Abstract iii

Résumé v

Contents viii

1 Introduction **1**

1.1 Context and Motivations 1

1.2 Contributions 6

1.3 Organization of the Thesis 7

2 Semi-supervised Feature Selection **9**

2.1 Introduction 11

2.2 Generalities 12

2.2.1 Notations 12

2.2.2 Pairwise constraints 13

2.2.3 Spectral Graph Theory 15

2.3 Graph-theoretic Dimensionality Reduction 18

2.3.1 Constraint Score For Semi-supervised Feature selection (C4) . 18

2.3.2 Semi-supervised Feature Selection via Spectral Analysis (sSelect) 19

2.3.3 Locality sensitive semi-supervised feature selection (LSDF) . . 21

2.3.4 Constrained Laplacian Score (CLS) 22

2.3.5 Constrained Selection-based Feature Selection (CSFS) 24

2.3.6 Semi-supervised Feature Selection for Regression Problems (SSLS) 27

2.3.7 Semi-supervised Dimensionality Reduction (SSDR) 28

2.4 Conclusion 30

3 Multi-label learning 32

3.1 Multi-label classification 34
3.1.1 Formal Definition 36
3.1.2 Applications 36
3.1.2.1 Text Categorization 36
3.1.2.2 Semantic Multimedia Annotation 37
3.1.2.3 Micro-array Gene Expression 38
3.2 Solving the Multi-label Learning Problem 38
3.2.1 Problem Transformation Methods 39
3.2.1.1 Binary Relevance (BR) 39
3.2.1.2 Label Powerset(LP) 40
3.2.1.3 Ranking by Pairwise Comparison (RPC) 41
3.2.1.4 Calibrated Label Ranking (CLR) 42
3.2.1.5 Random k-Labelsets (RakEL) 42
3.2.2 Algorithm Adaptation-based Methods 42
3.2.2.1 ML-kNN 43
3.2.2.2 Multi-label Decision Tree 43
3.2.2.3 Neural Networks 44
3.2.2.4 Ensemble Methods 44
3.2.2.5 Support Vector Machines 45
3.2.3 Challenges in Multi-label Learning 45
3.2.4 Related Tasks 47
3.2.4.1 Multi-instance Learning 48
3.2.4.2 Multiple-label Classification 48
3.2.4.3 Multi-task Learning 49
3.2.4.4 Multi-output Regression 49
3.3 Multi-label Dimensionality Reduction 49
3.3.1 Transformation-based Feature Selection 50
3.3.2 Algorithm Adaptation for Multi-label Feature Selection 53
3.3.3 Feature Extraction Techniques 55
3.4 Multi-label data tools 57
3.5 Conclusion 58

4 Laplacian Scores for semi-supervised multi-label feature selection 59

4.1 Notations 61
4.2 Transformation-based Approach 62
4.2.1 BR-CLS 62
4.2.2 LP-CLS 63
4.3 Soft-Constrained Laplacian Score: S-CLS 63
4.4 Experiments 68
4.4.1 Data sets and Methods 68
4.4.2 Experimental Settings 70
4.4.2.1 Evaluation with *ML-kNN*. 71
4.4.2.2 Evaluation Metrics. 72
4.4.3 Results 74
4.5 Conclusion 82

5 Semi-supervised Multi-label Feature Selection: An Ensemble Approach 84

5.1 Introduction 86
5.2 Ensemble Learning 87
5.3 Ensemble S-CLS 88
5.4 Experiments 90
5.4.1 Experimental Settings 90
5.5 Results and Discussion 90
5.6 Conclusion 94

6 Conclusion and Future Directions 96

Bibliography 115

List of Figures

2.1 A semi-supervised data set 13

3.1 Semantic scene annotation 35
3.2 Original multi-label data set 39
3.3 Binary Relevance 40
3.4 Label Powerset 41

4.1 Data structure for semi-supervised multi-label learning. 61
4.2 General framework of S-CLS. 68
4.2 AUC v.s. number of selected features 77
4.3 Nemenyi test diagram 78

5.0 AUC v.s. number of selected features 92

List of Tables

3.1 Multi-label data set 36

4.1 Description of the data sets used in the experiments 70
4.2 AUC and average rank 77
4.3 Results (mean±std.) on all data sets used, over all measures ("↘ indicates the smaller the better"; "↗ indicates the larger the better"). 79
4.4 Effects of S-CLS on different classifiers. Before and after selection. . 81
4.5 Execution Time in Seconds 82

5.1 Results (mean±std.) on all data sets used, over all measures ("↘ indicates the smaller the better"; "↗ indicates the larger the better"). 93

List of Algorithms

1 SC4 19
2 sSelect 21
3 LSDF 22
4 CLS 23
5 CSFS 26
6 SSDR 30
7 BR-CLS 62
8 LP-CLS 63
9 S-CLS 67
10 3-3FS 89

"There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things."

Niccolo Machiavelli

1 Introduction

1.1 Context and Motivations

NOWADAYS, the rapid surge of high-throughput digital data acquisition technologies has led to an exponential growth in the volume of the harvested data. Aided in this by an ever-increasing capabilities both in storage and computation. More than ever, a massive volume of data from various sources: digital cameras, sensors, patient records, stock market and retail transactions, gene sequencing, etc... is being collected at industrial scale oweing to ubiquitous and pervasive computing. Nevertheless, this deluge of data is not readily useful and needs to be processed and analyzed to extract meaningful knowledge. As John Naisbitt put it, "We are drowning in information and starving for knowledge" [Naisbitt84]. Unfortunately, the processing of such data is beyond human capacities, which calls for the need for effective and efficient automation of the task. In this respect, machine learning offers a plethora of data analysis tools and learning models. But sadly enough, virtually all learning models are challenged by the curse of dimensionality, which occurs when sample to feature ratio is too small [Duda12]. In fact, when

data is high-dimensional, many of the features describing them can be irrelevant and/or redundant. Which may have adverse effect on learning models, and often leads to overfitting, downgrading performance and reduced intelligibility of the learning models [Fukunaga13].

In this regard, dimensionality reduction (DM) has proven to be the ultimate solution used to mitigate the nasty impacts of the curse of dimensionality. Fundamentally, DM is based on the assumption that the intrinsic dimenionality of data often lies in a lower-dimension. This assumption paved the way to the development of a wide pectrum of algorithms and tools, but they basically all fall into two categories: feature extraction and feature selection. In the former, the data is projected into a new space with a lower dimensionality made up of artefact features which are chiefly obtained by a linear mapping of the original features. Herein, the literature abounds with well-known algorithms and widely used tools, among others, PCA [Jolliffe86], LDA [Fisher36] and SVD [Alter00]. The main drawback of this approach is that it reduces the interpretability of the model learnt, as the original features are transformed and artificial features are created in the low-dimensional data. In contrast, feature selection techniques, uses statistical characteristics of the data to select from the original features the most representative and informative features and discard the non relevant ones, thereby confers to feature selection a neat superiority over feature extraction, especially in terms of time complexity and better interpretability; quintessential examples of well-known feature selection techniques may include: Information Gain [Peng05], RelieFf [Robnik-Šikonja03], Fischer Score [Gu11].

More in particular, feature selection, also known as variable selection, consists in selecting from a feature space F, most discriminative features that would describe instances in a given data set at least as well as the whole feature space does; that is without any deterioration in performance or lost information. In other words, as defined in [García15], feature selection is a process that chooses an optimal subset of features according to a certain criterion. In so doing, we do not only reduce time complexity, but also enhance accuracy and interpretability of learning algorithms (in classification or clustering tasks). Feature selection is commonly used on data sets with a huge amount of features (*a.k.a* variables) such as those used in: text processing [Schapire00], gene expression array analysis [Diplaris05], and combinatorial chemistry [Guyon03].

Generally, feature selection is considered as a search problem aiming at rendering the optimal subset of features, in which we have 2^M different subsets of features to consider, where $M = |F|$ is the cardinality of the feature space. To this end, three parameters need to be specified before starting the search: the search direction, the search strategy, and the selection criteria. In fact, different search directions could be adopted to get the optimal subset: We can either begin from the whole feature space, and subsequently remove irrelevant features, or inversely start from an empty subset and add pertinent feature in each iteration, or start from both sides. This gives rise to three search directions: Sequential Forward Generation (SFG), Sequential Backward Generation (SBG), and Bidirectional Generation (BG). Another alternative that could be considered is to choose a random direction [García15].

In addition, different search strategies could be used: exhaustive search, consisting in exploring all possible subsets to find the optimal one; heuristic search that uses heuristics to conduct the search, preventing a brute force search but will certainly provide a non-optimal subsets, and non-deterministic search which is a combination of the previous two. The choice of a particular strategy is determined by finding a trade-off between the optimality of the resulting subset and available resources (in terms of time and space). Finally, the selection criteria also have a great impact on the selection, since they are the real tools to measure the quality of the selected subset, the chosen criteria could emphasize performance either in terms of efficacy or efficiency. For instance in classification we could tend to favor the accuracy over other performance.

Furthermore, selection criteria are often used to produce a ranking in the feature space. This ranking is basically achieved according to the type of the measure in use, this can be: information measures, like the Information Gain (IG); distance measures, like Variance (generally applicable to numeric features); dependence measures, also known as measure of correlation. Other measures can be adopted, like consistency measures which is generally used to detect redundancies in selected subset of features or accuracy measures which relies on the performance of a particular classifier to select a particular set of features. However, these two last measures do not allow to produce a ranking.

From another perspective, feature selection can be categorized in many respects: the cardinality of the evaluated subset; the interaction with the classifica-

tion algorithm, if any; the availability and the amount of supervision information. From each of these aspects stems a different feature selection paradigm.

More precisely, feature selection algorithms evaluate the informational worth of features in two distinct ways: individual evaluation or subset evaluation (univariate/multivariate evaluation). On the one hand, the individual evaluation estimates the worthiness of features independently of each others and assigns them weights (ranks) proportionate to their discriminative power (correlation with the class label). Numerous feature importance measures can be used in this regard. Obviously, this approach incurs less computational expenses. Nevertheless, the individual evaluation falls short of discovering redundant features as they are evaluated in isolation from each others, so it might happen to have features with similar rankings. On the other hand, the subset evaluation approach can cope with both, feature relevance and feature redundancy. In contrast to individual evaluation, the subset evaluation is defined against a subset of features instead of individual features, however it induces higher computational cost.

Besides, considering the interaction or the lack thereof with the learning algorithm, feature selection can be achieved in various ways. Basically, the techniques developed so far come into three categories, namely: filter, wrapper and embedded selection [Guyon03]. The interaction with the learning algorithms is what makes the difference among them. First, in the filter methods, there is no need to the learning algorithm to trigger the selection process. This is basically done by exploiting the general characteristics of the data itself using certain statistics criteria. In the output every single feature will be associated with a score that determines its relevance. Second, the selection in wrapper methods is done by searching the space of the original features using a search strategy to get a subset that best fits the learning task in terms of certain evaluation measures. Finally, embedded methods incorporate the selection into the training stage of the learning process. Moreover, feature selection can be further organized in three more paradigms, according to the prior domain knowledge which could possibly guide the process of selection. In the supervised paradigm, the most discriminant features are those that are highly correlated with the class labels [Dash97]. Unsupervised feature selection is considered as a much more difficult problem. Roughly speaking, this task can be achieved by assessing the variance or the separability power of features [Dy04]. In the semi-supervised context, the task becomes more challenging with

the so-called *small-labeled-sample problem*, in which the amount of data that is unlabeled can be much larger than the amount of labeled data [Zhao07]. Although feature selection is capable of dealing with all types of learning paradigm, the most well-known and frequently used field is classification.

In addition, most of the feature selection techniques developed in the last few years were mainly designed to support single-label learning. Wherein, each data instance is associated with only one class label, and the labels are mutually exclusive. This point of view is very restrictive, simplistic, and does not accommodate a lot of real-world application requirements. Motivated by this fact, the multi-label learning were introduced to allow instances to be associated with more than one class label simultaneously [Tsoumakas07a]. In fact, Multi-label learning is an emerging research field with an increasing number of applications, such as text categorizaiton [Schapire00], bioinformatics [Zhang06], and semantic scene annotation [Boutell04]. In this particular setting, the conventional single-label learning can be viewed as a particular case of multi-label learning. Unquestionably, this generality poses further challenges to feature selection which need to be addressed appropiately so as to take into account and take adavantage of the multi-label information[Zhang07b].

Standard solutions to multi-label classification follow two main strategies: problem transformation and algorithm adaptation. Problem transformation methods suggest to convert the multi-label problem into one or more single-label problem upon which traditional single-label classification could be applied. On the other hand, algorithm adaptation approaches consist in extending traditional single-label algorithms in order to fit the *"multi-labledness"* of the data. To address dimensionality reduction in multi-label framework, feature selection follows suit of multi-label classification, thus the proposed solutions are either transformation-based or algorithm adaptation.

Multi-label feature selection is relatively a new research field and needs more attention from machine learning researchers and practitioners. Actually, in the current literature, little work were interested in multi-label dimensionality reduction and existing research studies are in their vast majority based on feature extraction, and hardly ever focus on feature selection. In addition, the predominant approaches to multi-label feature selection are developed on the basis of problem transformation approach and rarely addressed multi-label selection directly. This

fact is even worse when dealing with semi-supervision and *"multi-labeldness"* conjointly. In this context, we suggest various solutions to achieve feature selection from partially-labeled data.

1.2 Contributions

The research study conducted in this thesis is in the junction of various learning paradigm. Chiefly, we put together semi-supervised, multi-label learning and feature selection. The ultimate goal is to develop new algorithms to address multi-label dimensionality reduction for *small-labeled-samples problems*. In this particular scheme, the thesis makes the following primary contributions:

- A general survey of graph theoretic-based semi-supervised feature selection, besides an overview of the available literature on multi-label classification and dimensionality reduction.
- The development and empirical evaluation of three filter feature selection algorithms for semi-supervised multi-label data sets. Using both prior information from the labeled part and the geometric structure of the unlabeled part of the data.
- An artifact solution to capture label importance by using a weighting mechanism. At the output, each label is endowed with a weight encoding its importance in relation to the target concept underlying the data. Eventually, those label weighs will be integrated into the feature selection procedure.
- An ensemble framework to solve semi-supervised multi-label feature selection in the aim of alleviating the effect of variance in the base algorithms.

More in particular, we introduce and empirically evaluate new feature selection algorithms we dubbed respectively: BR-CLS, LP-CLS, S-CLS [Alalga16] and 3-3FS. Each of which filters out relevant features through a specific objective function that is based on the Laplacian score. This objective function assigns a score to a feature according to its relevance to the target learning concepts, which in turn reflects its correlation to class labels and the ability to preserve the local structure of the data at the same time.

1.3 Organization of the Thesis

Throughout this research study we give emphasis to feature selection from various perspectives. We shall touch on dimensionality reduction for semi-supervised data. The first two chapters are dedicated to a literature survey on single-label semi-supervised feature selection; and multi-label classification and dimensionality reduction. Afterwards the subsequent chapters detail our contributions to multi-label dimensionality reduction for the semi-supervised learning paradigm. For so doing, the remainder of this thesis is structured as follows:

- First in chapter 1.3, we provide an overview on semi-supervised feature selection. More specifically, the chapter gives insights into theoretical facets underlying the most prominent works on semi-supervised feature selection in single-label settings. We focus on methods developed in the light of spectral graph theory, which lay the groundwork for the development of our own algorithms. Before we go into the details of each of the methods, we first briefly introduce principles of spectral graph analysis and underline their relation with semi-supervised learning.

- Then, chapter 2.4 highlights theoretical and practical aspects of multi-label learning. To this end, this chapter is structured into two parts. First, we commence by defining and characterizing multi-label classification, presenting some application domains and reviewing strategies to perform multi-label classification. In the second part of the chapter, we address principles of dimensionality reduction for multi-labeled data and review works on multi-label dimensionality reduction. Finally, the chapter concludes by presenting well-known tools recently coined to support and develop multi-label learning algorithms.

- In chapter 3.5, we present our framework to perform semi-supervised multi-label feature selection and discuss the foundational details of the proposed algorithms. We devise three filter feature selection algorithms based on spectral graph theory and a previous work on single-label feature selection (CLS [Benabdeslem11b]). The first two methods achieve multi-label dimensionality reduction by applying CLS in conjunction with problem transformation approaches. Subsequently a detailed description of our proposed S-CLS

(Soft-Constrained Laplacian Score) is presented. S-CLS achieves feature selection by exploiting prior domain knowledge expressed in terms of class labels from which we derive what we have called *"soft-constraints"* that will be integrated into the final feature importance measure. The experimental results are given solely for S-CLS which is the main contribution of this research. We shall compare our proposed algorithm with other state-of-the-art methods using various evaluation metrics and we shall prove the superiority of our algorithm by using statistical significance tests. Also, for the sake of fair comparison we shall use several benchmark data sets from various application domains.

- Subsequently chapter 4.5 presents an ensemble learning framework to S-CLS. We design an ensemble algorithm based on three-fold resampling. To be specific, we combine three subsampling techniques, each of which is applied on a different level of data. In 3-3FS, a Bagging is applied on the instance level, the Random Subsampling Method is conducted on the dimension level and finally a subsampling without replacement is performed on the label space level. The goal is to mitigate the effect of variance on S-CLS and take more advantage of label correlation when performing dimensionality reduction. Some ensemble methods are then depicted and compared with; empirical results demonstrate that the proposed framework does improve the performance and the stability of S-CLS within reasonable computing costs.

- Finally, chapter 5.6 winds up this thesis by a conclusion and presentation of some perspectives for future work.

"Change your opinions, keep to your principles; change your leaves, keep intact your roots."

Victor Hugo, Intellectual Autobiography

2

Semi-supervised Feature Selection

▷ *This chapter is devoted to feature selection in the context of semi-supervised learning. Issues related to semi-supervised dimensionality reduction are discussed. In fact, like other learning paradigms, high dimensionality is also problematic to semi-supervised algorithms. Which grapple with the curse of dimensionality, tending to produce more complex and less effective models. In this regard, we investigate how to bring the power of the spectral graph theory to design and implement convenient solutions to semi-supervised dimentionality reduction. Throughout the chapter, we outline several algorithmic proposals developed in the light of spectral graph theory.* ◁

Chapter outline

2.1 Introduction **11**

2.2 Generalities **12**

2.2.1 Notations 12

2.2.2 Pairwise constraints 13

2.2.3 Spectral Graph Theory 15

2.3 Graph-theoretic Dimensionality Reduction **18**

2.3.1 Constraint Score For Semi-supervised Feature selection (C4) 18

2.3.2 Semi-supervised Feature Selection via Spectral Analysis (sSelect) 19

2.3.3 Locality sensitive semi-supervised feature selection (LSDF) 21

2.3.4 Constrained Laplacian Score (CLS) 22

2.3.5 Constrained Selection-based Feature Selection (CSFS) . . 24

2.3.6 Semi-supervised Feature Selection for Regression Problems (SSLS) 27

2.3.7 Semi-supervised Dimensionality Reduction (SSDR) . . . 28

2.4 Conclusion **30**

2.1 Introduction

OBTAINING fully labeled data is usually expensive, tedious and time-consuming, for more often than not it requires the endeavor of human experts. This gives rise to the so-called *small-labeled sample problems*, in which a large volume of unlabeled data is provided together with a tiny proportion of labeled ones [Zhu05]. This view seems to be more in line with most real-world applications, where it is prohibitively expensive to produce complete supervision over data. For example, in bioinformatics, protein sequences are churned out at industrial speed, but resolving the functions of a single protein may require years of research efforts. In this regard, the limited amount of labeled data can be augmented by unlabeled data and eventually be utilized together to improve the learning performance. Nevertheless, it might happen that semi-supervised learning will not yield an improvement. It might even be the case where using the unlabeled data leads to performance deterioration by misguiding the prediction. In reality, the feasibility of semi-supervised learning is conditioned by certain assumptions, which need to be satisfied [Chapelle09].

Inspirited by the success of semi-supervised learning, researchers have introduced semi-supervised learning to the field of dimensionality reduction. Likewise, semi-supervised dimensionality reduction is halfway between supervised and unsupervised dimensionality reduction, the goal is to use small amount of labeled data as additional information to improve the performance of unsupervised dimensionality reduction. To confront high-dimensionality in semi-supervised data, solutions come in various flavors reflecting different learning assumptions. More precisely, three basic assumption come into play when dealing with semi-supervision: the smoothness assumption, the manifold assumption and the cluster assumption [Chapelle09]; The smoothness assumption states that if two points x_1, x_2 are close, then so should be the corresponding outputs y_1, y_2; the cluster assumption conjecture that if two points are in the same cluster, they are likely to be annotated by the same class label; more importantly the manifold assumption assumes that the intrinsic structure of the data lies on a low-dimensional manifold embedded in the high-dimensional data space.

There exist some influential guidelines when dealing with semi-supervised data. In particular, it is usually the case to have various forms of partial supervision apart

from label information, *e.g.* pairwise constraints or a distance metric, which give some insights to guide the learning process in the absence of full prior domain knowledge. Another influential direction in the field of semi-supervised dimensionality reduction is the use of the graph theory analysis, whereby the data is represented by a graph encoding the underlying target concept. Basically, the graph is constructed according to a certain leaning assumption. In this perspective, the dimensionality reduction is achieved by choosing the features that best preserve the inherent structure of the graph built. This global framework is generally referred to as graph-theoretic dimensionality reduction. In this chapter, we overview diverse strands of the application of this general framework. But beforehand, it is important to review some theoretical concepts related to the field of semi-supervised dimensionality.

2.2 Generalities

This section lays the foundation to dimensionality reduction based on spectral graph theory. First, we give a formal description of semi-supervised feature selection and introduce some mathematical notations that will be used throughout the thesis. Secondly, we see an example of partial supervision exemplified by the concept of pairwise constraints, for which we give a formal definition and review some theoretical characteristics. Afterwards, we delve into the detail of the Lapacian graph theory. Finally, we investigate the foremost representative methods from the general graph-theoretic framework.

2.2.1 Notations

In semi-supervised learning, a data set of N instances $X = \{x_1, ..., x_N\}$ is divided in two parts according to label availability: a supervised part $X_L = \{x_1, ..., x_L\}$ in which instances are associated with class labels from the set $Y_L = \{y_1, ..., y_L\}$, and an unsupervised part $X_U = \{x_{L+1}, ..., x_{L+U}\}$ comprising solely unlabeled instances. In the small-labeled sample context, the total number of instances $N = L + U$, where $L << U$. Note that, two particular cases occur when $L = 0$ or $U = 0$, which bring us back to unsupervised/supervised learning, respectively.

In this setting, each data instance x_i is a vector with M dimensions (features), and labeled by the integer $y_i \in \{+1, -1\}$, if it belongs to X_L. Let $F_1, F_2, ..., F_M$ denote the M features of X and $f_1, f_2, ..., f_M$ be the corresponding feature vectors that record the feature values in each instance. Figure 2.1 gives an example of semi-supervised data set.

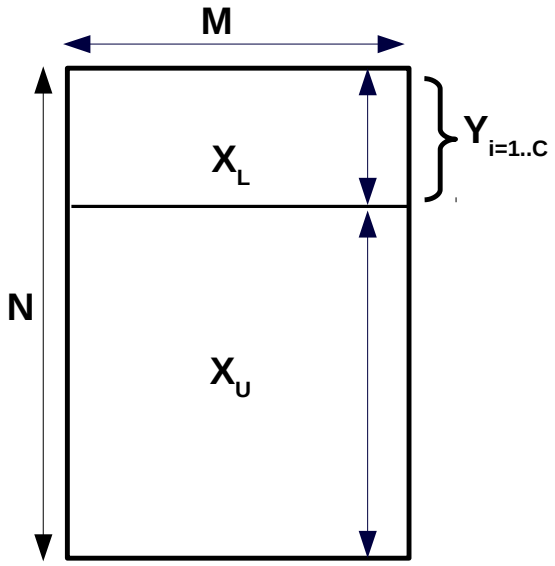


Figure 2.1: A semi-supervised data set

Semi-supervised feature selection exploits X_L together with X_U to single out the subset of most relevant features $F_{j_1}, F_{j_2}, ..., F_{j_h}$ of the target concept, where $h \leq M$ and $j_r \in \{1, 2, ..., M\}$ for $r \in \{1, 2, ..., h\}$. Relevant features here, are those that correlate with the class labels from X_L, while at the same time preserve the intrinsic geometric structure of the data in X_U.

2.2.2 Pairwise constraints

Domain knowledge can be expressed in diverse forms, such as class labels, pairwise constraints or any other kind of prior information. Undoubtedly, pairwise constraints are the most practical; they are much cheaper to obtain, and do not require a deep knowledge about the data. As opposed to label information which need detailed information about the classes and instance affiliations. In other words, pairwise constraints provide weak and general supervision information, *i.e.*

they merely indicate for some pairs of instances whether they are similar and must be grouped together (must-link constraints), or dissimilar and cannot be put together (cannot-link constraints).

Pairwise constraints were first introduced in the field of semi-supervised clustering [Wagstaff01], where they are used to constrain the cluster building by specifying that instances in the must-link relation should be associated with the same cluster, and those in cannot-link relation should be assigned to different clusters. The incorporation of instance-level constraints has shown promising results and has even permitted to outperform conventional method clustering relying solely on the inner structure of the data. In the context of dimensionality reduction these two sets of constraints act as a guide in the attempt of digging out informative features which are those that satisfy the specified must-link and cannot-link constraints.

Constraints preservation, as a relevance criterion, could be used in supervised or semi-supervised frameworks, and has lead to the development of plethora of feature selection algorithms [Tang07, Zhang07a, Zhang08a]. In the presence of pairwise constraints, semi-supervised dimensionality reduction algorithms seek features that best help preserve the structure of data, while, at the same time, reduce the constraint violation.

Typically, the pairwise constraints are expressed in terms of two sets:

- *Must-link*, denoted by $\Omega_{ML} = \{(x_i, x_j), \text{such that } x_i \text{ and } x_j \text{ must be linked}\}$
- *Cannot-link*, denoted by $\Omega_{CL} = \{(x_i, x_j), \text{such that } x_i \text{ and } x_j \text{ cannot be linked}\}$

These instance-level constraints have several interesting properties which are generally used to extend their content. More specifically, Must-link constraints are equivalence relations, that is, symmetrical, reflexive and transitive. The transitivity property allows to infer additional must-link relationships from the initial set [Bilenko04]. Cannot-link constraints, however, do not have such equivalence; $(x_i, x_j) \in \Omega_{CL} \wedge (x_j, x_k) \in \Omega_{CL}$ does not necessarily implies that $(x_i, x_k) \in \Omega_{CL}$., yet we can infer additional cannot-link constraints with the appropriate must-link and cannot-link constraints.

Formally:

- Transitive inference of Must-link constraints: $(x_i, x_j) \in \Omega_{ML} \wedge (x_j, x_k) \in \Omega_{ML} \implies (x_i, x_k) \in \Omega_{ML}$.

- Transitive inference of cannot-link constraints:$(x_i, x_j) \in \Omega_{CL} \wedge (x_j, x_k) \in \Omega_{CL} \implies (x_i, x_k) \in \Omega_{CL}$.

Commonly, a small-sized pairwise constraint subset is manually specified by a domain expert; the cardinality of a constraint subset is generally much lower than the total number of all possible combinations. On the other hand, it is also possible to derive such subsets automatically from the class label information, by stating that pairs of instances associated with the same class will be tied by a must-link constraint, and those pairs with different classes should be assigned to the cannot-link set. However, this cannot apply to multi-label learning, in chapter 3.5 we shall show how to generate appropriate constraints from multi-labeled data.

Finally, when leveraging level-instance constraints to serve feature selection, it would be wise to pay attention to incoherent/inconsistent constraints [Benabdeslem11a], for experiments had shown that they can have adverse effects and harm the performance of feature selection algorithm. Also, all constraints are not equally important, so it would be appropriate to incorporate a constraint weighting algorithm in order to give more importance to critical constraints.

2.2.3 Spectral Graph Theory

To any graph we may associate a matrix which records information about its structure. The goal of spectral graph theory is to study the relationship between eigenvalues and eigenvectors of such matrices and how they relate the structure of the corresponding graphs [Chung97]. Spectral graph theory have found applications in many fields, for instance one of Google's first algorithms was based on spectral analysis by using eigenvectors to rank pages from the Internet [Page99]. Generally, machine learning tries to represent data by graphs on which spectral analysis techniques can be applied to gain useful insights into the data.

Indeed, the target concept underlying the data can be reflected by the structure of a graph. To be specific, a data set X is commonly modeled as a weighted graph $\mathbb{G}(V, E, w)$ characterized by its vertex set V, edge set $E \subseteq VV$, and a weight function $E \rightarrow \mathbb{R}^+$. The vertices of the graph represent the instances, and the weighted edges encode the pairwise relationships. More specifically, the i^{th} vertex v_i of $\mathbb{G}$ corresponds to $x_i \in X$. and there is an edge between each vertex pair (v_i, v_j) endowed with a weight w_{ij}. In the sense that certain edges represent stronger

relations than others, and a missing edge corresponds to a non-relation. Typically, the edges E and weights w in the data graph $\mathbb{G}$ denote similarities between nodes in V. The weight function is often used to represent pairwise instance similarities. These are often symmetric, resulting in an undirected graph.

When building the graph $\mathbb{G}$ we can consider several options:

- The $\epsilon - neighborhood$ graph : Here we connect all points whose pairwise similarities are greater than ϵ
- $k - nearest\,neighbor$ graph: we connect two nodes v_i and v_j if the corresponding instances x_i and x_j are among the k-nearest neighbors of each other.
- The fully connected graph: Here we simply connect all points with positive similarity with each other, and we weight all edges by w_{ij}.

Typically, the graph $\mathbb{G}$ is built by using the nearest neighbors of each point; this ensures that the graph is sparse, which will speed up computation.

Besides, the similarities can be expressed in various ways : geometric structure, class label information or pairwise constraints, this has given rise to various learning algorithms. For instance, using class information, the similarity function could be defined by:

$$w_{i,j} = \begin{cases} \frac{1}{n_l} & y_i = y_j = l \\ 0 & otherwise \end{cases} \tag{2.1}$$

In the absence of prior knowledge, a similarity function could be based on the geometric structure of the data, e.g. a kernel function based on Euclidean distance between data points; the most used function is the Gaussian kernel defined by:

$$w_{ij} = e^{-\frac{\left\|x_i - x_j\right\|^2}{\lambda}} \tag{2.2}$$

Given the similarity function, we define the positive semi-definite adjacency matrix as follows:

$$A_{ij} = \begin{cases} w_{i,j} & if \quad (i,j) \in E \\ 0 & otherwise \end{cases} \tag{2.3}$$

In the context of feature selection, the adjacency matrix is commonly referred to as the similarity matrix and denoted by S

From the adjacency matrix we can compute the degree matrix D of the graph $\mathbb{G}$, which is defined by $D = diag(S\mathbf{1}), \mathbf{1} = [1, \ldots, 1]^T$.

The degree matrix can be interpreted as an estimation of the density around x_i since the more points that are close to x_i the larger D_{ii}

Given the adjacency matrix A and the degree matrix D of $\mathbb{G}$, the Laplacian matrix for the weighted undirected graph $\mathbb{G}$, also known as graph Laplacian is defined by:

$$L = A - D \tag{2.4}$$

The graph Laplacian satisfies the following properties: L is symmetric, positive semi-definite and $x^T Lx = \sum_{i,j}(x_i - x_j)^2$

In practice, it is common to normalize the graph Laplacian to account for the fact that some nodes are more highly connected than others, the normalized Laplacian matrix is defined by:

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{2.5}$$

In view of the graph theory, the learning process can be reduced to performing eigen-decomposition problems. More in particular, the clustering problem can be formulated as follows: We want to partition the graph such that the edges between different groups have low weights and the edges within a group have high weights. In feature selection we seek features that are consistent with the graph structure; features that assign similar values to instances that are near each other (strongly connected in the graph). The next section reviews state-of-the-art methods developed in the highlights of the principle of graph theory.

2.3 Graph-theoretic Dimensionality Reduction

This section exposes methods that rely on spectral analysis to achieve semi-supervised dimensionality reduction. In fact, relevant features are consistent with the target concept (in supervised learning the target concept is related to class affiliation and in unsupervised learning the target concept is encoded in the innate structure of the data) which, as we have seen above, can be captured by the structure of a graph. Therefore, analyzing the structure of similarity graphs to spot relevant features has resulted in a plethora of dimensionality reduction methods, each of which is characterized by its specific definition and building of the underlying similarity graphs. The common denominator of all these methods, referred to as graph-theoretic methods, is the graph based-assumption.

The graph-based assumption states that if $d(x_i, x_j)\, is\, small,\, then\, y_i \approx y_j$, In plain words, this means that the similarity between two points is proportional to their distance. This assumption applies regardless of the availability of the class information. Below, we go into the details of some algorithms that fall in the graph-theoretic framework.

2.3.1 Constraint Score For Semi-supervised Feature selection (C4)

Feature selection in mixed data sets can be achieved by hybridizing supervised and unsupervised algorithms. The hybridization can be as trivial as a mere product between score functions. In this line of thinking, [Kalakech11] proposed a semi-supervised constraint score by combining the Laplacian Score (LS) [He05a] and the Constraint Score (CS) [Zhang08b], two popular spectral graph based feature selection algorithms. Wherein, the Laplacian score is applied to the unlabeled part of the data and the Constraint Score seeks relevant features in the labeled part. The objective score function for a feature f_r, which needs to be minimized, is defined as:

$$C_r^4 = LS_r.CS_r \tag{2.6}$$

where LS_r and CS_r are the Laplacian Score and the Constraint Score for the f_r computed from the labeled and the unlabeled parts of the data, respectively. For the mathematical detail about both scores, see [He05a] and [Zhang08b].

A worthwhile mentioning study in this work in which the authors considered the fact that constraint-based algorithms are highly sensitive to changes in the constraint sets. In terms of feature selection, this means that changing the constraints subsets might lead to change in feature ranks, which eventually lead to different subsets of selected features. To resolve this issue, some work has suggested to use multiple constraint sets within the framework of ensemble learning [Sun10].

The phenomenon has been thoroughly discussed by Kalakch *et al.*, where the authors suggested to use the Kendall's Coefficient [Grzegorzewski06] to measure how the feature ranks vary with respect to different subsets of constraints. The Kendall's Coefficient, which value lies in $[0, 1]$, examine the concordance between feature ranks for different feature selection algorithms; the lower the Coefficient the less sensitive is the algorithm. The authors have empirically demonstrated that their ${C_r}^4$ is less sensitive to constraint changes than other methods they compared with.

Algorithm 1 SC4

Input: Data set X, pairwise constraints sets Ω_{ML}, Ω_{CL} and λ (a tuning parameter in the Laplacian Score, with default value =100)
Output: The ranked features
for $r = 1$ **to** M **do**
1: Calculate LS_r, the Laplacian score of F_r.
2: Calculate CS_r, the Constraint score of F_r.
3: Calculate $SC4_r$, the score of F_r using eq.(2.6)
end for
4: Rank the features according to their scores in ascending order.

2.3.2 Semi-supervised Feature Selection via Spectral Analysis (sSelect)

Based on the clustering assumption, Authors in [Zhao07] resorted to clustering techniques to solve the semi-supervised feature selection. In this spirit, the large amount unlabeled data is used to shape the cluster structure, whereas the role of

the labeled data is to guide the clustering operation. The spectral analysis is used to find the optimal solution to the so-defined clustering problem.

Basically, the proposed approach uses the feature vectors to construct cluster indicators for the clustering algorithm, here the normalized min-cut. the fitness of cluster indicators will determine the relevance of the corresponding features. The passage from a feature vectors to a cluster indicator is assured by a function called the *F-C* Transformation, defined as:

$$g_r = \theta(f_r) = f_r - \frac{f_r^T \boldsymbol{D}\mathbf{1}}{\mathbf{1}^T \boldsymbol{D}\mathbf{1}}.\mathbf{1}; \tag{2.7}$$

Where $f_r \in \mathbb{R}^n$ and $\mathbf{1} = (1, \ldots, 1)^T$.

The fitness of cluster indicator is determined by two factors: separability and consistency. The separability indicates how well separable are the cluster structure, which reflects the unsupervised point of view. On the other hand, the consistency shows the degree of agreement between the cluster structure and label information, reflecting the supervised point of view. In the ideal case, all labeled data of each cluster should come from the same class.

To find the best clustering, the normalized min-cut seeks to find a cut for $\mathbb{G}$ induced from the data X according to spectral graph theory, which minimizes the cost function defined by:

$$\eta \frac{g_r^T \boldsymbol{L} g_r}{g_r^T \boldsymbol{D} g_r} \tag{2.8}$$

Where η is a regularization parameter. Mathematically speaking, the fitness of a cluster indicator is evaluated by the following regularization framework:

$$sSelect_r = \eta \frac{g_r^T \boldsymbol{L} g_r}{g_r^T \boldsymbol{D} g_r} + (1-\eta)(1 - NMI(\hat{g}, Y_L)) \tag{2.9}$$

Where NMI is the normalized mutual information. $\hat{g} = sign(g)^2$ and is used to map the cluster indicators to classes.

The score $sSelect_r$ for a feature f_r reflects its relevance: the smaller the value of the score, the more relevant the feature. In the equation, the first term calculates the cut value of g (best clustering indicators are associated with minimum values), while the second term uses the normalized mutual information to estimate the accuracy of the clustering-induced classification. The regularization parameter η

is set empirically to favor either the impact of the unlabeled data or the labeled data in the fitness score.

Algorithm 2 sSelect

Input: Data set X, η, k
Output: the ranked features list
1: Construct the k-nearest neighbors graph G from X
2: Build the dissimilarity matrix $\boldsymbol{S}$, the degree matrix $\boldsymbol{D}$ and the Laplacian matrix $\boldsymbol{L}$ from G
for $r = 1$ **to** M **do**
  3: Construct the cluster indicators g_r from F_r using eq.(2.8)
  4: Calculate $sSelect_r$, the score of the feature F_r using eq.(2.9)
end for
5: Rank the features according to their scores in descending order.

2.3.3 Locality sensitive semi-supervised feature selection (LSDF)

Authors in [Zhao08] introduced a new algorithm based on manifold learning and spectral graph analysis. In the context of the spectral theory, the proposed approach builds two distinct graphs: a within-class graph G_w and a between-class graph G_b, in order to express the local geometrical and discriminant structure of the data. The importance of features is then calculated according to their power of preserving the structure of the two graphs.

The graph G_w connects instances with the same label or sufficiently close to each other, while G_b connects instances with different labels. The similarity matrices of these graphs are computed as follows:

$$\boldsymbol{S}_{w,ij} = \begin{cases} \gamma & \text{if } x_i \text{ and } x_j \text{ share the same label} \\ 1 & \text{if } x_i \textit{ or } x_j \text{ is unlableled, but } x_i \in KNN(x_j) \text{ or } x_j \in KNN(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

$$\boldsymbol{S}_{b,ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ have different labels} \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Where $KNN(x_i)/KNN(x_j)$ denotes the sets of k nearest neighbors of x_i/x_j respectively, and γ is a suitable constant empirically set to 100.

The graph Laplacians of G_w and G_b are defined by:

$L_w = D_w - S_w, D_w = diag(S_w\mathbf{1})$ $L_b = D_b - S_b, D_b = diag(S_b\mathbf{1})$

Where D_w and D_b are the degree matrices of G_w and G_b respectively.

The pertinence of a feature is then measured by the following objective function, which needs to be maximized:

$$L_r = \frac{f_r^T L_b f_r}{f_r^T L_w f_r} \tag{2.12}$$

for $r = 1, 2, \ldots, M$.

Features with the highest scores are ranked first and will be eventually selected.

The score is self-explanatory, the most informative features are those for which the within-class and between-class graph structure are best preserved. Loosely speaking, a feature is considered relevant if at this dimension nearby points, or points sharing the same label, are close to each other, while points with different labels are far apart. Algorithms 3 outlines all steps of LSDF.

Algorithm 3 LSDF

Input: Data set X, γ, k
Output: List of ranked features
1: Construct the within-class and the between-class graphs (G_w, G_b) from X
2: Compute the weight matrices S_w and S_b, the degree matrices D_w and D_b, and the Laplacian matrices L_w and L_b from G_w and G_b respectively
for $r = 1$ **to** M **do**
  4: Calculate L_r, the score of the feature F_r using eq.(2.12)
end for
5: Rank the features according to their scores in descending order.

2.3.4 Constrained Laplacian Score (CLS)

Another attempt to bridge the chasm between unsupervised and supervised feature selection can be found in [Benabdeslem11b]. The authors devised a more elaborate aggregation of Laplacian score and Constraint score. The proposed score–*CLS*,

benefits from both the data structure and the supervision information to pinpoint discriminative features.

More concretely, *CLS* tends to select features with the best locality and constraints preserving abilities. The justification for this is quite intuitive, on one hand, a relevant feature is expected to have close values for instances linked by a *Must-link* constraint and/or nearby instances, and disparate values for faraway instances or those in the *Cannot-link* subset.

To this end, *CLS* builds two distinct graphs: the dissimilarity graph G_{kn} representing data points in the *Cannot-link* set, and the neighborhood/ similarity graph G_{CL} connecting neighbor data points and/or those belonging to the *Must-link* set. In this particular setting, the importance of a feature is the degree to which it respects the structure of these two graphs.

From the above the objective function of *CLS* is formulated as follows:

$$CLS_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 \boldsymbol{S}_{ij}}{\sum_i \sum_{j|\exists k,(x_k,x_j)\in\Omega_{CL}} (f_{ri} - \alpha^i_{rj})^2 \boldsymbol{D}_{ii}} \tag{2.13}$$

where :

$$\boldsymbol{S}_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are neighbors or } (x_i, x_j) \in \Omega_{ML} \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

and:

$$\alpha^i_{rj} = \begin{cases} f_{rj} & \text{if } (x_i, x_j) \in \Omega_{CL} \\ \mu_r & \text{otherwise} \end{cases} \tag{2.15}$$

Note that if there are no labels ($L = 0$ and $X = X_U$) then $CLS_r = LS_r$ and when ($U = 0$ and $X = X_L$), CLS represents an adjusted CS_r, where the ML and CL information would be weighted by $\boldsymbol{S}_{ij}$ and $\boldsymbol{D}_{ii}$ respectively in the formula.The whole procedure of the proposed CLS is summarized in Algorithm 4.

Algorithm 4 CLS

Input: Data set $X(NM)$, the constant λ, the neighborhood degree k
1: Construct the constraint sets (Ω_{ML} and Ω_{CL}) from Y_L
2: Construct the graphs G_{kn} and G_{CL} from (X, Ω_{ML}) and Ω_{CL} respectively.
3: Calculate the weight matrices $\boldsymbol{S}^{kn}$, $\boldsymbol{S}^{CL}$ and their Laplacians $\boldsymbol{L}^{kn}$, $\boldsymbol{L}^{CL}$ respectively.
for $r = 1$ **to** M **do**
  4: Calculate CLS_r according to eq.(2.13).
end for
5: Rank the features F_r according to their scores CLS_r in ascending order.

It would be important to note that the method also favors features with high representative power (high variance).

Experiments have shown that *CLS* gives good performance in comparison with other state-of-the-art methods, nonetheless exhibits high sensitiveness to noise in the constraint sets (incoherence/inconsistency), which may lead to downgraded performance.

2.3.5 Constrained Selection-based Feature Selection (CSFS)

In the endeavors to leverage the pairwise constraints in feature selection, [Hindawi11] proposed an algorithm called CSFS which tries to alleviate sensitivenss with respect to constraints. CSFS is motivated by the empirical observation that "some" constraints may have adverse effect on performance [Davidson06]. Indeed, incoherent/contradictory pairwise constraints are harmful. It is important to mention that the incoherence is not due to noise or error; the constraints could be directly derived from the label information, and yet be incoherent.

The coherence is a measure proposed to quantify the degree of agreement between constraints. Constraints can be regarded as "attractive/repulsive" forces in the feature space. In this sense, two constraints are incoherent if they exert contradictory forces (have parallel force vectors) in the same vicinity, and coherent if their force vectors are orthogonal. With this in mind, it suffices to compute the projected overlap of each constraint vector on the other to determine their coherence. That is, two constraints are coherent if their corresponding projected overlap is equal to zero.

The above can be stated formally as follows: Let $m \in \Omega_{ML}$, $c \in \Omega_{CL}$ be two constraint vectors connecting two points, the coherence of a constraint set $\Omega = \Omega_{ML} \cup \Omega_{CL}$ is given by:

$$COH(\Omega) = \frac{\sum_{m \in \Omega_{ML}, c \in \Omega_{CL}} \delta(over_m c = 0 \wedge over_c m = 0)}{|\Omega_{ML}|\,|\Omega_{CL}|} \tag{2.16}$$

Where $over_m c$ represents the distance between the two projected points linked by m over c. δ is the number of the overlapped projections.

With this quantification, one effective way to lessen negative effects of incoherence is to perform constraint selection prior to the learning task. More in particular, before submitting the constraint set to the selection algorithm, it is beneficial to apply a kind of coherence-based constraint selection to filter out incoherent constraints. To be selected a constraint needs to be fully coherent with other constraints (zero projected overlap with all available constraints). The complete algorithm is outlined in Algorithm 5

The objective function of the score is formulated as follows:

$$\varphi_r = \frac{\sum_{i,j} (f_{ri} - f_{rj})^2 (\boldsymbol{S}_{ij} + \boldsymbol{N}_{ij})}{\sum_i \left(f_{ri} - \alpha_{rj}^i\right)^2 \boldsymbol{D}_{ii}} \tag{2.17}$$

Where:

$$\boldsymbol{S}_{ij} = \begin{cases} e^{-\frac{\left\|x_i - x_j\right\|^2}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

And:

$$\boldsymbol{N}_{ij} = \begin{cases} -e^{-\frac{\left\|x_i - x_j\right\|^2}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are neighbors and } (x_i, x_j) \in \Omega'_{ML} \\ \left(e^{-\frac{\left\|x_i - x_j\right\|^2}{\lambda}}\right)^2 & \text{if } x_i \text{ and } x_j \text{ are neighbors and } (x_i, x_j) \in \Omega'_{CL} \\ & \text{OR} \\ & \text{if } x_i \text{ and } x_j \text{ are not neighbors and } (x_i, x_j) \in \Omega'_{ML} \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

And:

$$\alpha^i_{rj} = \begin{cases} f_{rj} & \text{if } (x_i,\ x_j) \in \Omega'_{CL} \\ \mu_r & \text{otherwise} \end{cases} \tag{2.20}$$

Where Ω'_{CL} and Ω'_{ML} denote the new constraint subsets (comprising only coherent constraints), λ is a constant to be tuned. x_i, x_j are neighbors in the sense that x_i is among the k-nearest neighbors of x_j and vice-verse. $\mu_r = \frac{1}{n}\sum_i f_{ri}$ is the mean of the column corresponding to the feature f_r.

Note that the term N_{ij} is introduced to penalize bad cases corresponding to distant instances related by *Must-link* constraints, and inversely nearby instances related by *Cannot-link* constraints.

Another major contribution of the method stems from the observation that a fixed number of neighbors k could be misleading when defining the local structure of the data. In fact, A score based on k-nearest neighbors graph could be biased by far neighbors, and thus fail to capture the real locality structure of the data. To fix this issue, the authors proposed to conduct a similarity-based clustering to determine the appropriate k. So that two instances are neighbors if they belong to the same cluster. In this particular setting, k is equal to the number of instances in each cluster and varies in each data set. To obtain an optimal partition, the authors chose to apply AHC(*Ascendant hierarchical clustering*) using *Davies Bouldin* as internal index.

Algorithm 5 CSFS

Input: Data set $X(NM)$, the constant λ
Output: Ranked features

1: Construct the constraint set (Ω_{ML} and Ω_{CL}) from Y_L
2: Select the coherent set (Ω'_{ML} and Ω'_{CL}) from (Ω_{ML} and Ω_{CL}) based on 2.16
3: Construct the graphs G_{kn} and G_{CL} from (X, Ω'_{ML}) and Ω'_{CL} respectively.
4: Calculate the weight matrices $\boldsymbol{S}^{kn}$, $\boldsymbol{N}^{kn}$ and $\boldsymbol{S}^{CL}$ and the Laplacians $\boldsymbol{L}^{kn}$, $\boldsymbol{L}^{CL}$.
5: **for** $r = 1$ **to** M **do**
6: Calculate φ_r according to eq.(2.17)
7: **end for**
8: Rank the features F_r according to their scores φ_r in ascending order.

In the sequel of their research works, the same authors developed two other methods, named CSFSR [Benabdeslem14] and ECLS [Benabdeslem16], which are designed to tackle issues related to feature redundancy and ensemble learning, respectively.

Till now we have seen various scenarios where feature selection benefits classification or clustering tasks. Next, we present a different example that shows that feature selection could also be useful in regression problems.

2.3.6 Semi-supervised Feature Selection for Regression Problems (SSLS)

It is well established that redundant and uninformative features decrease performance and make learning algorithms prone to overfitting. This statement holds true for regression problems. In regression problems, the task is to learn a mapping from the space feature to the output space whose values are defined on $\mathbb{R}$.

In this particular setting, [Doquire13a] devised a new feature selection criterion which is inspired by the *Laplacian Score*. The goal is to choose features according to their ability of preserving the locality structure of the data. With the particularity here that the prior information (class labels) is expressed in terms of continuous values.

To this end, the authors first developed a supervised version of the score upon which they built a semi-supervised variant. The underlying assumption is that

close instances have close output values, and therefore best features are expected to have close values for instances with close outputs.

According to that assumption, authors developed the similarity matrix S^{sup} defined by:

$$S_{ij}^{sup} = \begin{cases} e^{-\frac{(y_i - y_j)^2}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are close} \\ 0 & \text{otherwise} \end{cases} \tag{2.21}$$

The supervised version of the score called *SLS*, which should be minimized, is then defined as follows:

$$SLS_r = \frac{\tilde{f}_r^T \boldsymbol{L}^{sup} \tilde{f}_r}{\tilde{f}_r^T \boldsymbol{D}^{sup} \tilde{f}_r} \tag{2.22}$$

where

$$D^{sup} = diag(S^{sup}\mathbf{1}),\ L^{sup} = D^{sup} - S^{sup},\ \tilde{f}_r = f_r - \frac{{f_r}^T D^{sup}\mathbf{1}}{\mathbf{1}^T D^{sup}\mathbf{1}}\mathbf{1}$$

The explanation of the objective function is straightforward. The first term helps preserve the local geometrical structure of the data by keeping features that are most coherent with the above-defined similarity measure, while the second term is used to discard features associated with low variance since they do not have much descriptive power; the same as in the Laplacian Score.

In the semi-supervised variant of the *SLS Score*, the similarity measure have to be rethought to incorporate information from the unlabeled data. To this end, [Doquire13a] defined a new distance function that computes pairwise distances between instances from the whole data set.

$$d_{ij} = \begin{cases} (y_i - y_j)^2 & \text{if } y_i \text{ and } y_j \text{ are unknown} \\ \frac{1}{n}\sum_{k=1}^{n}(f_{ki} - f_{kj})^2 & \text{otherwise} \end{cases} \tag{2.23}$$

Once the pairwise distances are set, the SSL Score is reformulated into a semi-supervised one (SSLS) as follows:

$$SSLS_r = \frac{\tilde{f}_r^T \boldsymbol{L}^{semi} \tilde{f}_r}{\tilde{f}_r^T \boldsymbol{D}^{semi} \tilde{f}_r} SLS_r \tag{2.24}$$

where the similarity matrix is redefined as:

$$S_{ij}^{semi} = \begin{cases} e^{-\frac{d_{i,j}}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are close and } y_i \text{ and } y_j \text{ are unknown} \\ Ce^{-\frac{d_{i,j}}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are close and } y_i \text{ and } y_j \text{ are known} \\ 0 & otherwise \end{cases} \tag{2.25}$$

Where, the positive constant C is supposed to give more importance to the supervised part of the data. The closeness here is to be understood in the sense of k-nearest neighbors.

The same justification mentioned above applies for the semi-supervised version of the score. The score permits to retain features with high locality structure preserving ability and high predictive power. Finally, it is worthwhile mentioning that both the equations 2.22 and 2.24 are thoughtfully designed to give more importance to the labeled part of data sets, the reason is that we naturally trust more the known than the unknown.

2.3.7 Semi-supervised Dimensionality Reduction (SSDR)

Feature extraction seeks informative and non-redundant features by finding a mapping of a high-dimensional space into a space of fewer dimensions. The new space consists of features that are calculated as a function of the original features. In this context, [Zhang07a] proposed a feature extraction algorithm that uses both prior supervision information in the form of instance-level constraints along with the structure of the unlabeled data. The principle that underpins this approach is to compute from the original feature space a low-dimensional representation that will preserve the intrinsic structure of the data and satisfies the pairwise constraints at least as well as the original feature space does.

More concretely, find the projective vectors W and compute the artificial substitute features $y_i = W^T x_i$ that accommodate the intrinsic structure of the data and comply with the pairwise constraints. For the reason that instances sharing a must-link relation should be close and those in a cannot-link relation should be far away. This leads to the development of the following objective function, which needs to be maximized:

$$\begin{aligned} J(w) = & \frac{1}{2n^2} \sum_{(x_i,x_j)\in C} (w^T x_i - w^T x_j)^2 \\ & + \frac{\alpha}{2n_C} \sum_{(x_i,x_j)\in C} (w^T x_i - w^T x_j)^2 \\ & - \frac{\beta}{2n_M} \sum_{(x_i,x_j)\in M} (w^T x_i - w^T x_j)^2 \end{aligned} \tag{2.26}$$

Where n_C, n_M is the number of instances in the Cannot-link and the must-link sets respectively. α, β are trade-off parameters to balance the impact of the different terms. The formula is self-explanatory, the main goal is to keep distances between instances in the must-link set as small as possible and distances between instances in cannot-link set as large as possible. Similar to PCA [Jolliffe86], the first term enforces the global covariance structure of the data.

From spectral graph point of view, the equation could be reformulated as follows:

$$J(w) = \frac{1}{2} \sum_{i,j} (w^T x_i - w^T x_j)^2 S_{i,j} \tag{2.27}$$

where

$$S_{ij} = \begin{cases} \frac{1}{n^2} + \frac{\alpha}{n_C} & if \quad (x_i, x_j) \in C \\ \frac{1}{n^2} - \frac{\beta}{n_M} & if \quad (x_i, x_j) \in M \\ \frac{1}{n^2} & otherwise \end{cases} \tag{2.28}$$

By further algebraic developments we have $J(w) = w^T X L X^T w$, where L is the *Laplacian matrix*. The last equation is a typical eigen-problem which can be solved by finding the eigenvectors of XLX^T corresponding to the largest eigenvalues. The whole process is outlined in Algorithm 6.

Algorithm 6 SSDR

Input: Data set X, pairwise constraints sets Ω_{ML} and Ω_{CL}, the size of the new dimension d
Output: A d-dimensional data set
1: Build G the similarity graph from data X
2: Compute the similarity matrix $\boldsymbol{S}$ using eq.(2.28)
3: Compute the eigenvectors of XLX^T corresponding to the largest eigenvalues
4: Sort the eigenvalues with the corresponding eigenvectors in descending order
5: Find the projective vectors corresponding the top d sorted eigenvectors
6: Compute the lower-dimensional data matrix $\mathsf{Y} = W^T X$

Note that the approach has the credit of defining the similarity matrix S independently from fetching the k-nearest neighbors; choosing the right k is a prime example of model selection problems which are very tricky to solve. On the opposite side, this leads to neglect the local structure of the data when computing distances between instances not involved in any constraint. What's more, solving an eigen-problem for large-scale data sets is prohibitively expensive and unfeasible.

2.4 Conclusion

This chapter summarizes concepts and existing works related to the domain of semi-supervised feature selection, in which few labeled data are exploited together with huge amount of unlabeled data to filter out irrelevant/redundant. First, we started by a brief presentation of some key concepts such pairwise constraints and spectral graph theory, which are the building blocks for most semi-supervised dimensionality reduction tools. Afterwards, we reviewed the literature of semi-supervised dimensionality reduction, whereby theoretical backgrounds and full descriptions of some representative graph-theoretic algorithms are presented and thoroughly discussed.

"Imagination is the beginning of creation. You imagine what you desire, you will what you imagine and at last you create what you will."

George Bernard Shaw

3

Multi-label learning

▷ *The goal of this chapter is to introduce multi-label learning, as well as to give a broad overview of its main application fields and how it has been tackled by researchers. A general introduction to the matter is provided. First, we begin by defining and motivating the need for multi-label classification. Afterwards, we give a formal definition and illustrate some domain applications. Subsequently, we show how to attain predictions from multi-label data sets. Besides we look into and discuss challenges specific to multi-label classification and what distinguishes it from certain similar learning tasks. Then, we advocate the need for multi-label dimensionality reduction and survey both feature selection and feature extraction algorithms in multi-label data. Finally, we give a bundle of tools developed to support design and implement multi-label solutions.* ◁

Chapter outline

3.1 Multi-label classification **34**
3.1.1 Formal Definition 36
3.1.2 Applications 36
3.2 Solving the Multi-label Learning Problem **38**
3.2.1 Problem Transformation Methods 39
3.2.2 Algorithm Adaptation-based Methods 42
3.2.3 Challenges in Multi-label Learning 45
3.2.4 Related Tasks 47
3.3 Multi-label Dimensionality Reduction **49**
3.3.1 Transformation-based Feature Selection 50
3.3.2 Algorithm Adaptation for Multi-label Feature Selection . 53
3.3.3 Feature Extraction Techniques 55
3.4 Multi-label data tools **57**
3.5 Conclusion **58**

TRADITIONALLY, there have been fundamentally two different types of classification in pattern recognition: binary and multi-class classification. On the one hand, binary classification dichotomizes the data into positive and negative instances with respect to the unique class. On the other hand, multi-class classification generalizes the binary classification by allowing the presence of more than one class. However, the emergence of new domain applications, where the labeling of real world objects such as texts, images, music, and video involves assigning multiple labels at the same time, calls for the need of a novel learning paradigm. In this context, the multi-label classification is introduced to meet this new requirement. Multi-label classification problems abound in practice. As a result, many methods have recently been propose to mine multi-label data. Broadly speaking, this task can be accomplished through two main approaches: data transformation and algorithm adaptation. In this chapter, we shall touch on various aspects of multi-label classification as well as dimensionality reduction for multi-labeled data.

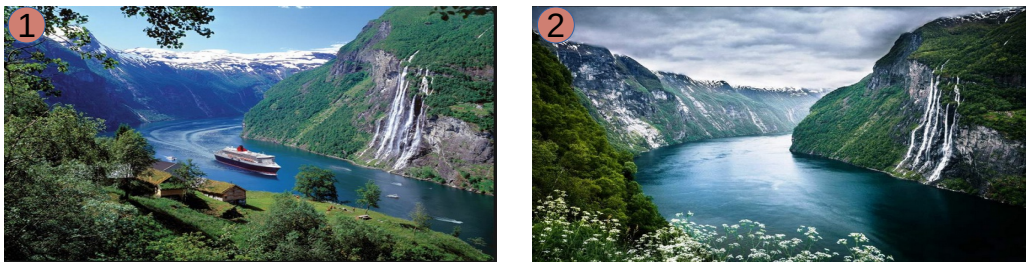
3.1 Multi-label classification

In pattern recognition, conventional single-label classification is concerned with learning from instances belonging only to one category from a set $Y = \{1, 2, ..., y, ..., C\}$ of mutually exclusive categories (C is the number of different labels). In this setting, the cardinality of this set determines the nature of the classification problem. When there is only two disjoint class labels, $C = 2$, we talk about binary classification. On the other hand, if there exist more than two classes ($C > 2$), the classification task is said to be multi-class.

In contrast to single-label learning, multi-label classification transcends the dichotomy of binary approaches and the categorical mutual exclusivity of multi-class classification by allowing instances to be categorized into more than one class simultaneously, for the benefits of a vast variety of real-world applications [Zhang14]. By loosing up the restriction imposed on the number of classes an instance can belong to, each instance will be associated with a subset of class labels.

Multi-label classification takes its origin in works on text categorization, and has ever since rapidly drawn much attention and become a very active research area. In fact, multi-label classification has been applied successfully in many real-

world problems from various expertise domains. For instance, in gene expression array analysis [Diplaris05], a gene can have two or more functions. Similarly in semantic scene annotation [Boutell04, Qi07], an image could refer to different objects it may contain, and in text categorization [Schapire00], a document can be associated with several topics according to the subjects it refers to. Figure 3.1 is a typical example of semantic scene annotation where images are annotated according to the contained objects. Section 3.1.2 goes more in detail through various applications of multi-label classification.



Images	Objects					
	Tree	Fjord	Ferry	Hut	Waterfall	Snow
1	X	X	X	X	X	X
2	X	X			X	X

Figure 3.1: Semantic scene annotation

Commonly, multi-label classification induces a bipartition of the label space into relevant and irrelevant labels in response to a query instance. However, mining multi-label data can be approached from a ranking perspective, in which the goal will be to learn a real-valued function that maps from instances to rankings over labels. Noteworthy that label ranking is not mutli-label learning specific and can be applied in single-label context.

In some cases, it would be beneficial to have a synergy of multi-label classification and label ranking. In this spirit, some methods offer solutions that combine these prediction tasks by learning in one shot a bipartition and an ordering of labels according to their degree of relevance, in such a way that the most relevant labels top the list. The later process is referred to as multi-label ranking and can be thought of as a combination multi-label classification and label ranking. This scenario has numerous practical applications, in a news filtering application for

example, it is helpful to respond to query instance by listing interesting articles only, but it is also important to put the most interesting ones in the top of the list.

3.1.1 Formal Definition

Let X denote the instance space and Y denote the label space, as outlined in Table 3.1, the task of multi-label learning is to learn a hypothesis $h : X \to 2^Y$ from the training set $\{(x_i, Y_i) \mid 1 \leq i \leq N\}$, where $Y_i \subset Y$ is the subset of labels associated with x_i. Note that $Y_i = (y_{i1}, y_{i2}, ..., y_{iC})$ is a binary-valued vector, where $y_{ij} = 1$ if x_i is associated with the label y_j.

Table 3.1: Multi-label data set

	X				Y
E_1	x_{11}	x_{11}	...	x_{1M}	Y_1
E_2	x_{21}	x_{22}	...	x_{2M}	Y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{11}	...	x_{NM}	Y_N

3.1.2 Applications

Successful applications of mutli-label algorithms have been reported for a wide variety of domains, wherein the observed objects are innately related to multiple categories. In this section, we detail the need and usage of multi-label classification in various fields ranging from text categorization to functional genomics [Tsoumakas09a].

3.1.2.1 Text Categorization

In Multi-label text categorization, the aim is to select multiple category labels from a predefined set of not mutually exclusive categories; labels refer to a document's topics. By allowing documents to belong to several categories simultaneously, we accommodate various real-world applications [Schapire00, Ueda03, Veloso07].

Text data are ubiquitous, it can be found in big companies, health care facilities, digital libraries, particular households, where there is all kind of stored reports, medical patient records, e-books, invoices and electronic mail messages. In the World Wild Web, news reports, blog posts, question-answering forum mes-

sages, social networks constitute an overwhelming source of textual corpora. This tremendous volume of textual documents has given birth to a wide range of data sets used as benchmark for classification algorithms motivated by various needs, like automatic cataloging of news reports and web pages, sorting of electronic mail, etc. In many of this applications, documents are generally associated with more than one category.

It is worthwhile mentioning that prior to applying text categorization algorithms, it is important to transform documents into some suitable representation. A straightforward representation would be based on words which are the building blocks of textual documents. To find the best word-based representation, documents are parsed for representative pertinent words and uninformative ones are discarded. In the process, each document will be described by a vector of words and each word is considered as a separate feature. In this setting, there exist two main ways to represent documents: Set-of-words approach, in which a word is associated with a boolean value indicating whether the word occurs or does not occur in the document. The Bag-of-words representation, in which words are associated with a numeric value indicating their frequency in the document. In both representation, it is common to annotate documents with multiple class labels.

3.1.2.2 Semantic Multimedia Annotation

The Advent of multimedia databases and their advanced query interfaces calls for the need of incorporating machine learning techniques for dynamic resource indexing. In fact, knowing the category of a resource a priori helps narrow the search space dramatically. In other respects, semantic multimedia classification finds application in areas such as including content-based image indexing and organization, content-sensitive image enhancement, semantic-level video browsing and tracking objects in video streams. The common denominator of all these tasks is that objects are chiefly associated with multiple tags. Multi-label learning bespoke the needs of semantic multimedia applications. Various research studies have addressed semantic multimedia annotation from the perspective of multi-label classification [Boutell04, Trohidis08, Qi07].

The primary goal in scene classification is to categorize an image automatically based on the content. More often than not, images and videos feature a wide range of objects. Accordingly, multiple concept labels could be simultaneously

affected to it. Multi-label classification is well tailored to semantic multi-media categorization.

3.1.2.3 Micro-array Gene Expression

Functional genomics has witnessed the emergence of innovative sequencing and micro-array technologies allowing high throughput measuring techniques which resulted in overwhelming amount of genomic data. In general, the analysis of this huge quantity of data aims to measure the expression levels of thousands of genes. The sheer volume of these data makes it prohibitively beyond the human capacity of analysis, which has motivated the automation of the task.

Automatic processing of gene expression data falls into three major tasks: gene function prediction is concerned with the classification of gene according to their biological roles [Zhang06]. This prediction relies on the assumption that genes with similar functions have similar expression profiles in cell [Hvidsten01]. By their very nature, genes are related to multiple functions (a single gene can belong to up to 10 different functional classes). In terms of multi-label learning, each function is represented by a different label.

Likewise, protein function prediction goal is to infer the function of proteins, departing from the fact that a protein may perform multiple roles [Yu13]. Finally, protein sub-cellular multi-location has to do with the localization of a protein in a cell; proteins may simultaneously exist at different sub-cellular locations [Wang13]. As such, the aforementioned tasks are readily modeled through multi-label learning.

3.2 Solving the Multi-label Learning Problem

Obviously, solutions to issues addressed in single-label learning cannot be directly imported in the multi-label context, due to its intrinsic complexity. This statement is also true for feature selection problems. Generally, two ways of thinking have emerged to handle this question, namely: problem transformation and algorithm adaptation [Tsoumakas07a]. While the former tries to adapt data to algorithms, the latter seeks to adapt algorithms to the data. More particularly, problem transformation approaches propose converting the original multi-label learning into one

or more single-label learning problems, and then aggregate the individual solutions of different single-label classifiers to obtain a prediction for a multi-label request. On the other hand, problem adaptation methods works by extending existing single-label algorithms to fit multi-label data requirements.

3.2.1 Problem Transformation Methods

In transformation-based methods, the original learning problem is decomposed into serevral single-label subproblems, in the next step all these subproblems are solved separately, and their results are combined to infer a single-label solution. Simple transformation techniques used to convert a multi-label data set into a single-label one consist of selecting among the label subsets of each instance the most frequent label in the data set (select-max), the least frequent label (select-min) or a random label (select-random) [Boutell04, Chen07]. Another type of transformation consists of copying each multi-label instance n times, where n is the number of labels assigned to that instance. Each copied instance is then assigned one distinct single label from the original set. Beside these simple transformations techniques, there is a plethora of more elaborate methods, among which Binary Relevance (BR) and Label Powerset (LP) are the most employed[Tsoumakas09a]. Next, we detail the most prominent methods. For illustrative purposes, consider the multi-label data set sketched in Figure 3.2.

Examples	Classes			
	Health	Technology	Lifestyle	Politics
EX.1	X			X
EX.2			X	X
EX.3	X			
EX.4	X	X	X	

Figure 3.2: Original multi-label data set

3.2.1.1 Binary Relevance (BR)

In Binary relevance, a multilabel problem with C possible classes is decomposed into C binary problems according to the one-against-all (OAA) principle. More specifically, for the prediction of a new instance by binary relevance (BR), one first transforms the multi-label data set X into $X_{|Y|}$ data sets, where Y is the label space. Each resulting data set contains all instances of the original multi-label data set, labeled by y or $\neg y$, depending on whether the label y figured in the label subset of the corresponding instance or not, respectively. Afterwards $|Y|$ binary classifiers are built from previously generated single-label data sets, and the final solution will be the subset of labels positively predicted by every binary classifier [Tsoumakas09a]. Noteworthy that any literature binary classifier can be used in this context. Binary Relevance has a linear complexity with respect to the number of labels which amounts to $\mathcal{O}(C)$. However, the major concern with Binary Relevance is its underlying assumption of label independence, which is generally not true in real world applications. Figure 3.3 depicts the Binary Relevance transformation.

Examples	Class
	Health
EX.1	X
EX.2	
EX.3	X
EX.4	X

Examples	Class
	Technology
EX.1	
EX.2	
EX.3	
EX.4	X

Examples	Class
	Lifestyle
EX.1	
EX.2	X
EX.3	
EX.4	X

Examples	Class
	Politics
EX.1	X
EX.2	X
EX.3	
EX.4	

Figure 3.3: Binary Relevance

3.2.1.2 Label Powerset(LP)

Label Powerset(LP) was introduced to deal with label correlation. As its name indicates, to solve the multi-label problem, LP generates a new class for each subset of labels in the power set of Y. As a result, we get a multi-class data set, on which we can use any traditional multi-class classifier. The final solution to the prediction of an unseen instance will be the subset of labels corresponding to

the predicted class of the multi-class classifier [Tsoumakas09a]. Definitely, this approach does take into consideration correlation between labels, but still has a serious disadvantage; we can end up with a large number of generated classes associated with few number of instances which results to an imbalanced data, and thus makes it prone to overfitting. Besides, its time complexity makes it inefficient in large data sets especially those consisting of a significant number of label combinations.

This later issue was addressed in a variant of Label Powerset referred to as Pruned Problem transformation (PPT) [Read08], which proposes that we discard, from the generated multi-class data set, classes associated with a number of instances lower than a fixed threshold. This pruning has a double effect, it results in a simplified version of the learning problem and ensures that all classes are represented by at least a number of instances equal to a preset threshold, which have been proven to mitigate the class imbalance issue and improve the efficiency and scalability on large data sets. Both PPT and LP has a worse-case time complexity equal to $\mathcal{O}(min(N, 2^C))$ [Read10]. Figure 3.4 illustrates the LP transformation. Note that the generated data set is a multi-class where classes are mutually exclusive.

Examples	Classes			
	Health	health-Politics	Technology-Health-Lifestyle	Lifestyle-Politics
EX.1		X		
EX.2				X
EX.3	X			
EX.4			X	

Figure 3.4: Label Powerset

3.2.1.3 Ranking by Pairwise Comparison (RPC)

Ranking by pairwise comparison [Hüllermeier08] transforms the problem of label ranking to several binary classifications. To this end, RPC generates from the

original multi-label data set a total number of $C(C-1)/2$ binary label data sets, one for each pair of labels (y_i, y_j). Each data set comprises only instances that are associated with at least one of the corresponding labels, but not both. A binary classifier is then trained on each of these data sets. When a query instance is given, it will be submitted to all binary classifiers and their predictions are combined using a voting scheme to produce a ranking.

3.2.1.4 Calibrated Label Ranking (CLR)

Calibrated label ranking was proposed to tackle the multi-label ranking problem [Fürnkranz08]. CLR is an extension of RPC that can learn a bipartite partition of the label space along with the ranking provided by RPC. This is done by introducing an additional label to the original label set, which acts as a split point that distinguishes between relevant and irrelevant labels, so that all relevant labels are ranked above the additional label, which in turn is ranked above all irrelevant labels. By doing so, the virtual label calibrates a ranking by splitting it into positive and negative parts.

3.2.1.5 Random *k*-Labelsets (RakEL)

RakEL is an ensemble method for multi-label classification based on LP transformation [Tsoumakas07b]. The random k-label sets builds an ensemble of LP classifiers, each of which is trained using a small random label subsets of size k drawn without replacement from the original label space. The classification follows a simple and easy way, for a new instance, first each LP classifier provides a binary decision for each label in the corresponding k-label set. Afterwards, the algorithm averages the decision for each label in the original label space, and labels with an average greater than a threshold will be kept for the final classification decision. In this way, *RAkEL* is capable of dealing efficiently with label correlation thanks to constructing different multi-class classifier with smaller subsets of labels. However, this approach suffers from a major drawback in terms of time complexity which amounts to $\mathcal{O}(2^k)$.

3.2.2 Algorithm Adaptation-based Methods

In algorithm adaption paradigm, multi-label methods tries to adapt, extend or customize an existing machine learning algorithm for the task of multi-label learning. The goal is to modify existing algorithms by taking into account the multi-label nature of the samples. For instance, in multi-label decision trees, we have to allow hosting more than one class in the leaves of a tree instead of only one.

Most algorithm adaptation-based methods extend traditional single-label algorithms in order to be applicable to multi-label problems, so as to operate directly on the original data sets without any transformation. Extensions to a wide range of algorithms can be found in literature [Tsoumakas07a]. It has been argued that adaptation-based methods outperform transformation-based ones. In this thesis, two of the algorithms we propose fall in line with the this approach to achieve feature selection in multi-label data sets.

3.2.2.1 ML-kNN

ML-kNN [Zhang05] is a multi-label adaptation of the traditional *kNN* algorithm. In ML-kNN, the classification is straightforward, for an unseen example, the algorithm firstly fetches it's k nearest neighbors, herein the Euclidean metric is used to measure distances between instances. Secondly, for each label in the label space, we compute its frequency within the retrieved neighbors. Finally, the set of labels of the unseen example is determined by maximum a posteriori principle (MAP) which proceeds by computing prior and posterior probabilities. *ML-kNN* belongs to the category of lazy learning methods in which no model of the classifier is built and the generalization is postponed until a request is submitted. Those methods are very sensitive to irrelevant features and widely used to measure performances of feature selection algorithms.

3.2.2.2 Multi-label Decision Tree

With multi-label data sets we need to allow leaves of the tree to potentially have multiple labels. Several variants of traditional decision tree algorithms were extended to be applicable in multi-label data sets. For example in [Clare01], the authors suggested to extend the C4.5 algorithm [Quinlan14] by using a Informa-

tion Gain criterion based on a modified version of Entropy capable of capturing the multi-label nature of instances. The adapted formula of Entropy is defined by:

$$Entropy(X) = -\sum_{i=1}^{|Y|} p(y_i) * log_2 p(y_i) + q(y_i) * log_2 q(y_i)$$

Where, $p(y_i)$ is the probability that an arbitrary instance in the data set D is associated with the class label y_i, and $q(y_i) = 1 - p(y_i)$. $|Y|$ is the number of labels in the data set.

3.2.2.3 Neural Networks

Numerous work proposed to adapt neural networks for multi-label learning by extending different base algorithms. The most succesful one was proposed by[Zhang06], named BP-MLL, is the multi-label version of the popular back-propagation algorithm. The Back-propagation works by iteratively minimizing the sum-of-squares error on output nodes. It runs until the error drops below some threshold or the number of iterations alloted is exhausted. The main aspect in BP-MLL is the introduction of a new error function that takes multi-labeled instances into account. The global error function is reformulated as follows:

$$E = \sum_{i=1}^{N} E_i$$

where $E_i = \sum_{j=1}^{|Y|} (y_j^i - \hat{y}_j^i)^2$

λ_j^i is the predicted class of the network for the instance x_i and $\hat{\lambda}_j^i$ is it's real class. This new error function attempts to consider correlations among labels which plays important role in multi-label classification.

3.2.2.4 Ensemble Methods

The multi-labeled version of the well-known Adaboost algorithm [Freund95] was declined into two variants Adaboost.MH and Adaboost.MR [Schapire00]. Adaboost.MR is designed for ranking purposes whereas Adaboost.MH focuses into minimizing the *Hamming Loss*. As in Adaboost, the key idea is to construct an ac-

curate classifier by a linear combination of simple weak classifiers. The algorithms iteratively increases/decreases weights on wrongly/correctly classified examples. To be specific, Adaboost.MH is presented with instance-label pairs and in each iterations increases the weights of the misclassified ones. While Adaboost.MR works on pairs of labels and adjusts the weights of instances with respect to the corresponding pairwise label order.

3.2.2.5 Support Vector Machines

The first attempt to adapt support vector machines to mulit-label context was introduced in [Elisseeff01]. The proposed algorithm, named Rank-SVM, is a direct approach based on SVM principles, it adapts maximum margin strategy to tackle mutli-label settings by allowing multi-label margin for each instance in the training data set. In the process, a set of linear classifiers, one for each label, are trained and optimized to produce the optimal label ranking through minimizing the empirical ranking loss. As an instantiastion of SVMs, Rank-SVM can be extended to handle nonlinear cases by using a kernel trick [Schölkopf02]. The approach has the merit to take label correlations into account.

Efficiency and performance of Rank-SVM have been improved by two major contributions. The first attempt, called Rank-CVM[Xu13], aims at reducing the computational complexity of Rank-SVM. The resulting classifier gives similar predictive performance to Rank-SVM but is more efficient. The second tweak, named SCRank-SVM [Wang14], focuses on improving the efficiency as well as the accuracy performance of Rank-SVM. To do so, authors proposed to reformulate the calculus of the decision boundary in the underlying SVMs, and to simplify some of the existing constraints to maximize the margin, mainly by discarding some of the SVMs' parameters, thereby reducing the computation complexity.

3.2.3 Challenges in Multi-label Learning

Understandably, the complexity of a classification problem is proportionally related to the number of the classes; the more classes, the more complex the problem. In one sense, this is because it causes the odds of incorrect classifications to increase. This complexity is further compounded by the generality of multi-label learning, which adds another layer of difficulty by dismissing label mutual exclusivity. In

addition to not knowing the number of correct classes, the number of combinations an output can take is now significantly larger, which in turn introduces more room for error.

In the particular context of multi-label learning, since instances are tagged with multiple labels simultaneously, some dependence of labels may occur. This phenomonon has been empirically corroborated by various research studies concluding that often correlations exist between labels [Dembczyński12]. Loosely speaking, this means that the occurence of a certain label could determine whether another one is also more likely to be present or not. In semantic scene classification, for instance, the probability of the label eye would be higher if the label face is also relevant. Usually, researchers distinguish between conditional and unconditional label dependency [Dembszynski10]. Conditional (also local) label dependence is determined by a given instance (the condition), whereas marginal (unconditional) dependence captures the global label dependence in the label space regardless of the instances.

Arguably, leveraging label correlation is of extreme importance to multi-label learning. Indeed, the modeling and exploitation of label correlations would to lead to improving performance and producing more accurate learners. Both empirical and theoretical studies have given sound proof thereof. Unfortunately, this additional information are not explicitly provided in the data. Several attempts have been made to tacit and model such label correlations. On the other side, exploiting label dependence has motivated numerous multi-label learning methods which have produced promising results [Read09, Cesa-Bianchi06, Tsoumakas08]. Nevertheless, effectively using these dependencies to serve multi-label learning is a very challenging task and has not yet attain maturity.

In domains such as semantic scene classification, text categorization and functional genomics classes are generally organized in the form of hierarchical structures such as a tree or a directed acyclic graph (DAG). This gives rise to a specific type of learning commonly referred to as hierarchical multi-label classification. In contrast to flat classification, hierarchical multi-label classification is characterized by an underlying assumption that imposes that an instance belonging to a given class should automatically belong to all its super-classes. Undeniably, harnessing such structure would yet again be very profitable in ameliorating classification performance.

Multi-label data are often highly imbalanced, this means that some classes are heavily populated while others have feeble frequency and are only associated with way too few instances. Usually, the less frequent classes turn out to be the most important. Problems with such strongly skewed class distributions have proven to be very challenging even to the most elaborate state-of-the-art algorithms and this becomes worse with extremely large number of labels (which could amount to millions). Indeed, it is hard to predict sparse classes due to their limited numbers of training instances. However, exploiting label dependencies could help in imbalanced multi-label classification.

As we mentioned above, class labels might be counted in millions. For example, in data sets like Amazon[1](670k labels), WikiLSHTC[2] (320k labels), Delicious-Large [3] (200k labels) the label number amounts to hundreds of thousands. At this scale, multi-label learning would be a daunting task. In such cases, the computational complexity increases sharply with the tremendous (exponential) number of possible label sets. As a sequence, most algorithms will struggle and even become impractical, algorithms like BR and LP, for instance, become completely infeasible due to the prohibitive computational costs. Using label dependencies and structures could help. Some methods suggested to perform dimensionality reduction on the label space in order to lower both time and space complexity [Weston02, Tsoumakas07b, Hsu09]. Dealing efficiently with the scalability problem posed by data sets with large label space is still an open question.

3.2.4 Related Tasks

Supervised learning literature abound about algorithms devised to solve various prediction tasks in different learning contexts. A whole spectrum of interesting learning frameworks are readily available, some of which might naively be mistaken for multi-label learning. Next, we go through algorithms most akin to multi-label learning in a bid to clarify the confusion by explaining their differences and where they may apply.

[1]Developed as benchmark in recommender systems with item-to-item collaborative filtering approach

[2]A data set created from Wikipedia for the Large Scale Hierarchical Text Classification challenge

[3]Collected form the social bookmaring service (https://del.icio.us/)

Indubitably, amongst all learning frameworks the multi-class classification is the most commonly used. As explained in section 3.1, multi-class classification is distinct from multi-label classification in that it assumes mutual exclusivity between class labels. In effect, while mutli-label classification allows instances to be associated with more than one label at a time, multi-class classification enforce mutual exclusivity in the label space. Besides multi-class learning there exist a score of other learning tasks that could misconstrued and confused with multi-label learning, below we review all of them.

3.2.4.1 Multi-instance Learning

Another similar learing pardigm is multiple-instance (also known as multi-instance). Multi-instance classification finds its origins in drug activity prediction works [Gärtner02], since then it has been applied to a wide spectrum of applications ranging from text categorization to stock market prediction [Andrews03, Maron98]. The multi-instance problem arises in domains where the training instances are ambiguous: A single instance in the data set could be described by many alternative feature vectors. In this setting, each example consists of a bag of instances instead of a unique instance. In other words, there is a many-to-one relationship between feature vectors and their class labels. The instances in each bag are at a random order and bags vary in size: different bags may contain different number of instances. A classifier trained on such data, operates at the bag level and takes a bag as its input, and generates a decision for the bag. A bag is labeled positive if at least one of its instances is positive, and it is labeled negative if all its instances are negative. This scenario pertain to the binary case, where there is only two possible outputs. Certain tasks, however, align better with the multi-label case, given rise to mutli-instance mulit-label learning where the goal is to classify unseen bags by assigning them subsets of class labels [Zhou07].

3.2.4.2 Multiple-label Classification

There exist another learning framework that may be confused with multi-label learing and contrasts with multi-instance learning, in which the ambiguity lies on the side of the classes instead of the instances as we have seen in multi-instance classfication. This learning framework, referred to as multiple-label classification, is a variant of semi-supervised learning where each instance in the training data

set is associated with a set of candidate class labels and only one is presumed to be the correct label [Jin02]. What's more, there is no prior information to tell for certain which one it is. Such problems might arise, during the labeling of data if different expert domains give different labels to the same instance, in such cases, we would not be able to know which, if any, is sure enough correct.

3.2.4.3 Multi-task Learning

While in multi-label learning we learn multiple labels simultaneously, there exist a learning paradigm called multi-task learning [Caruana98] that seeks to learn multiple tasks at the same time. In essence, multi-task learning is based on learning classifiers for many similar prediction tasks in parallel, using a shared representation. The goal is to improve generalization accuracy by leveraging common domain knowledge. The main assumption that underlies multi-task learning is that combined learning of multiple related tasks can outperform learning each task in isolation. In fact, related tasks can benefit each other if trained together so that what is learned for one particular task can help other tasks to be learned better. Multi-task learning is applicable to a wide variety of domains, where there exist multiple sources of information but not sufficient when considered separately, in speech recognition, for instance, data from different speakers can be combined to yield better results.

3.2.4.4 Multi-output Regression

Multi-label learning can be considered as a particular case of the more general learning framework, called multi-target learning in which each instance is associated with a multi-dimensional output [Tsoumakas09b]. Accordingly, different types of response variables would give rise to different instantiations of multi-target learning. In the same vein, multi-target, multi-output or multi-variate regression is another imporant learning tasks in this general framework, which occurs when the observed response variables are quantitative real-valued. As opposed to the closely related task of multi-label clasification where the response variables are qualitative. Multi-ouput regression aims to predict a vector of real values for an unseen query. Numerous applications for multi-output regression have been studied, such as stock market prediction, power generation forecasting and ecological modeling [Kocev09, Aho12].

3.3 Multi-label Dimensionality Reduction

Inevitably, all learning algorithms are challenged by the curse of dimensionality posed by high-dimensional data, and multi-label learning is no exception. In fact, Many high-performing multi-label learning algorithms completely fail when the dimensionality is high since data points become sparse and far apart from each other due to noisy/irrelevant and redundant features. Sadly enough, most real-world multi-label applications generates massive data sets with an overwhelming number of potential features. In text categorization and functional genomics, for instance, feature space may be in an order of magnitude of thousands to millions features.

Dimensionality reduction techniques are prerequisites for combating the curse of dimensionality, and dealing effectively and efficiently with large data. Dimensionality reduction is fundamentally based on projecting the data to a lower dimensional feature space which is considered to be a compact representation of the essence of data. This representation can be artificially fabricated by combining original features (feature extraction) or be merely a result of selecting most relevant features from the original data without transformation (feature selection).

Multi-label dimensionality reduction can be approached in several ways. An intuitive direction is to follow suit of multi-label classification solutions by adapting either the multi-label data to traditional single-label (binary/mulit-class) algorithms or by extending single-label algorithms to handle multi-label problems. While feature selection algorithms can be employed either ways, feature extraction cannot be applicable with binary transformation techniques (like Binary Relevance). Obviously, Binary data transformation does not make sense here, since there is no way to aggregate artificial features of the different single-label subproblems to get the final multi-label solution. Hence, the only way to apply feature extraction in multi-label label data is to devise new algorithms from scratch or work the other way around through adapting single-label algorithms to fit multi-label data. Next, we shall touch on different techniques according to the approaches aforementioned.

3.3.1 Transformation-based Feature Selection

The most common transformation-based feature selection methods fall in the category of filter approaches. Basically, they estimate the importance of a feature based on some statistical properties computed from the training data set independently of any particular classifier.

In the literature, various transformation strategies are used together with filter approaches to achieve mutli-label feature selection. Filter approaches need to pass by a transformation phase to be applicable in the multi-label context [Tsoumakas09a]. A straightforward solution is to combine single-label feature selection algorithms with data transformation methods. The first successful multi-label feature selection techniques were obtained using data transformation along with single-label importance measures to evaluate the usefulness of features, such as the Fisher score, Chi-square, ReliefF, Information Gain, to name a few [Zhao10]. The most used transformation techniques are BR and LP with its variant PPT.

Binary relevance (BR) is the most used binarization technique in dimensionality reduction. As explained earlier, BR converts the multi-label problem to a number of binary sub-problems. After which, the prediction decision for the multi-label problem is obtained by joining the individual predictions made by the binary classifiers. In the same spirit, multi-label filter feature selection by binary relevance evaluates the discriminative power of features with respect to each of the labels in isolation from the rest of the labels. Afterwards, feature scores are aggregated to obtain a global ranking. The most common aggregation strategies consist in considering the maximum or a weighted average of the obtained scores. Various evaluation measures have been combined with BR to achieve multi-label features selection. In [Spolaôr13a], authors applied BR in conjunctin with ReliefF and Information Gain, the resulting methods named, RF-BR and IG-BR showed good results in comparison with other methods. In [Spolaôr13b], χ^2 statistic was combined with BR, the authors used and compared various aggregation techniques: average, maximum, round-robin and rand-robin [Forman04]. χ^2 statistic was also used in combination with BR in various works related to text categorization [Tsoumakas07b, Chen07]. In all these methods labels are treated separately, thus they fall short in capturing label correlation which is evidently a determining factor for measuring the importance of features.

Label power set transformation, converts the multi-label problem into a multi-class problem, by mapping each distinct label combination to new meta-class in the corresponding multi-class problem. Subsequently, conventional multi-class classifiers are used to make predictions in the form of a meta-class, which will eventually be translated back to the original label subset. Likewise, LP multi-label feature selection transforms the multi-label problem into a multi-class problem. Afterwards feature selection is performed by classical single-label algorithms. Several research studies align with this framework. In [Spolaôr13a], authors used LP along with ReliefF and Information Gain to evaluate informative features in multi-label data, and the developed methods give good performance. [Trohidis08] combined LP with χ^2 statistic, the combination was successfully used to reduce data size without compromising performance. Feature selection is then ensured by selecting the top-ranked features according to a chosen evaluation measure. Unlike Binary Relevance, the LP framework has the merit to effectively take into account label correlation. Nonetheless, overfitting and class imbalance from which suffer LP methods may harm the quality of feature selection. For that reason, many feature selection works have resorted to another variant of LP, called Pruned Problem Transformation.

With the Pruned Problem Transformation the data instances whose class labels apprear less than a fixed threshold in the training set are discarded. PPT-based multi-label feature selection is achieved by using PPT as a preprocessing step to classical feature selection algorithms. In this setting, several evaluation measures have been used. In [Doquire11] used Mutual Information as evaluation measure. PPT is first applied to transform the multi-label problem into a muti-class one. Thereupon, a greedy forward feature selection strategy based on multivariate mutual information is conducted. It begins with an empty set of features and first selects the feature with the highest Mutual information with the class vector. Then, progressively, the algorithm selects the feature not yet selected whose addition to the current subset of selected features leads to the set having the highest MI with the classes. The selection continues until a stop criterion is met. The proposed algorithm has the merit to capture redundant features owing to the use of a multivariate evaluation measure. Authors in [Reyes15] presented PPT-ReliefF, which uses Pruned Problem Transformation to convert the multi-label problem into a multi-class problem. Once the problem is converted, ReliefF is used to measure and rank the features according to their usefulness in distinguishing instances. In

their experiments, the authors applied PPT-ReliefF prior to learning several lazy classifiers, which has proven to give better performance than the baseline classifiers (without the selection step).

Another work on data transformation which is worthwhile mentionning is [Spolaôr14b]. In contrast to other transformation-based methods, the proposed algorithm, called the Label Construction for Feature Selection (LCFS), performs data transformation on the label space level. LCFS constructs new labels based on pairwise relations between the original labels. In so doing, the original data set is augmented with second-order information before being submitted to the feature selection algorithm. More specifically, in a two-step process, LCFS iteratively employs a selection strategy (random selection/heuristic strategy) to choose, from the original label space, a pair of correlated labels, then according to a certain combination strategy, generates a new single label encoding the pairwise relationship. Afterwards, the new generated labels are incorporated as new labels in the original data set. Eventually, feature selection is performed by applying BR in conjunction with Information Gain as a feature importance metrics on the augmented data set. Empirical results obtained on 10 benchmark multi-label data sets, showed that LCFS gives superior performance in comparison to the standard feature selection (without the label construction phase).

The major drawback of transformation-based algorithms is that the performance are closely biased by the transformation step. What's more, problem transformation is computationally expensive for domains with a large number of labels. This calls for the need for direct methods to achieve multi-label feature selection.

3.3.2 Algorithm Adaptation for Multi-label Feature Selection

Adaptation based methods squarely conduct feature selection on the original multi-labeled data without the need to any prior transformation. This leads to better performance, computational scalability and understanding of the learning problem. In this context, a large score of feature selection techniques from different paradigm were extend to deal with multi-label data. They consist mostly of algorithm adaptations of well-known single-label feature selection techniques. They could be categorized in the three well-known approaches: filter, wrapper and embedded. Wrapper methods may be directly portable to the multi-label data; it

suffices to use a multi-label classifier to evaluate feature subsets that were found in the previous search step. On the other hand, traditional filter methods are not readily usable and must be adapted to be applicable on multi-label data set. Next, we review most prominent filter-based feature selection works.

The Mutual Information measure was adapted by [Lee13]. To select features with the most discriminating power, the authors devised a score function that rewards features having the largest dependencies, in terms of Mutual Information, with respect to the set of labels. To this end, the score function is obtained by decomposing MI between the feature and the label sets into a series of multivariate mutual information. Subsequently, an incremental selection is performed by using a forward search strategy which starts with an empty subset of features and iteratively adds features which maximize the score of the subset. The developed method is capable of taking into account label interactions.

[Pereira15] has adapted another information-theoretic measure, namely the Information Gain measure to handle multi-label data directly. The authors used the multi-label version of Entropy, defined in [Clare01], to contrive a new multi-label Information Gain measure, which was used to evaluate the best features individually. The developed algorithm, MLInfoGain, was compared against other transformation-based Information Gain metrics on a large set of databases. In most cases, MLInfoGain achieved best performance in terms of accuracy, also for larger data sets, the algorithm exhibits better scalability than the other feature selection methods.

The ReliefF measure was adapted by [Pupo13]. The proposed method called ReliefF-ML extends single-label ReliefF to assign weights to features according to their discriminative power. More in particular, the feature weight reflects the ability of the feature to distinguish class labels, a high weight indicates that for instances with very different sets of labels the feature values are far apart and are very close for instances with similar label subsets. Initially, ReliefF-ML is designed to be a feature weighting, not as a multi-label feature selection method. Nonetheless, feature selection could be achieved through ranking the features according to their weights and then selecting the top ranked ones. The same approch was used in another work presented in [Reyes15]. Based on the principles of ReliefF, the authors proposed two extensions named ReliefF-ML and RReliefF-ML. On the other hand, RReliefF-ML is based on RReliefF, a variant of the classical ReliefF which

was designed for regression problems. [Spolaôr14a] also employed ReliefF to resolve feature selection in mutli-label data. This work proposes an algorithm called RF-ML. All proposed algorithms take into account interaction among features and deal effectively with label dependencies.

[Lastra11] extended the single-label Fast Correlation-Based Filter (FCBF) algorithm which was first introduced in[Yu04]. In their algorithm named MLfR (multi-label feature ranker), The authors employed a graphical model to represent the correlation and interdependence among labels and features. These relationships are encoded by a non-linear correlation measure called Symmetrical Uncertainty SU (A normalized variant of the Mutual Information). More in particular, given a mutli-label data set, MLfR first builds a matrix of SU scores including all the features and the labels. Subsequently, it computes the spanning tree of the complete undirected graph. The vertices of the graph include the features and the labels, and the edges are weighted by SU scores. The selection is performed by choosing the vertices corresponding to features whose distance from the whole set of labels is lower or equal to a given threshold. In addition to the relevance analysis, MLfR is also capable of detecting redundancies in both the feature and label spaces.

The correlation-based feature selection was also used in the multi-label setting by [Jungjit12]. The proposed technique proceeds by evaluating subsets of features, instead of individual features. In particular, the algorithm called ML-CFS uses simple hill-climbing algorithm to perform a heuristic search in the space of the candidate features looking for the feature subset that maximizes the objective function. Like in [Hall00], the objective function is defined so as to favor features with high predictive accuracy and minimize the correlations between pairs of features in the selected feature subset (To discard redundant features). By evaluating feature subsets, at each iteration, the method effectively deals with interaction among features. The same authors proposed in [Jungjit13], two extensions to their work by incorporating the absolute value of the correlation coefficient in the equations of the objective function and using Mutual Information for weighting the class labels.

3.3.3 Feature Extraction Techniques

Feature extraction is a task different from feature selection. In the sense that, it projects the original feature space into another low-dimensional space made up of new artifact features. Basically, the projection is obtained through a linear transformation. When the projection is guided by prior information about class labels, the ultimate goal is to maximize the correlation between the feature space and the label space. In multi-label framework, one possible solution to achieve feature extraction is to use problem transformation algorithms as a preprocessing step to applying traditional single-label dimensionality reduction algorithms. For instance, in [Park08], the standard single-label Linear discriminant analysis LDA [Fisher36] is used together with the copy transformation to achieve multi-label LDA. However, similar to feature selection, the major drawback is that the label correlations are often ignored in the transformation step. On the other hand, the computational costs increase exponentially with size of the label space. For these reasons, a large body of research has extended single-label feature extraction algorithms to directly handle multi-label data sets. In the following, we give an overview of most important works.

[Yu05] proposed the Multi-label Informed Latent Semantic Indexing (MLSI), as an extension of the traditional LSI (Latent Semantic Indexing [Deerwester90]) algorithm. MLSI was used in text categorization where the label information is leveraged to improve document indexing. While LSI is a completely unsupervised algorithm, MLSI makes use of the additional knowledge encoded in the labels to build the lower dimensional projection. In order to incorporate the labeling information, the mapping is computed for both the feature space and the corresponding labels. The final solution is obtained by solving an eigen problem where eigenvectors with largest eigenvalues are directly integrated into the mapping function. By so doing, MLSI preserves the information of the input space and capture the dependencies among the labels. As a result, the new feature space will represent both the information of the original features and the label space.

In [Wang10], the standard LDA is modified to handle multi-label data. The proposed technique, the Class Balanced Linear Discriminant Analysis (BLDA), is based on redefining the single-label LDA within-class and between-class scatter matrices for the multi-label setting, in order to allow instances to belong to several

classes at the same time. From the multi-label between-class scatter matrix and the multi-label within-class scatter matrix, the transformation matrix is computed using the standard LDA algorithm. In fact, the problem is reduced to solving an eigen problem where the new dimensions of the data set correspond to the top eigenvectors. The effectiveness of the algorithm was experimented on various data sets from different domains.

[Zhang10], proposed a dimensionality reduction algorithm called Multi-label Dimensionality reduction via Dependence Maximization (MDDM). The underlying key idea is to find a projection such that the dependencies between the features (data) and the corresponding labels is maximized in the new projection. To this end, it creates a ranking of features by maximizing the dependence between the features and the associated class labels using the well-known dependence measure, Hilbert-Schmidt independence criterion[Gretton05]. In the process, the dimensionality reduction problem is formulated by an eigen-decomposition problem, whose solutions constitutes the reduced feature space. The algorithm showed competitive results when compared with other methods like PCA and MLSI in conjunction with the Multi-Label ML-kNN classifier.

Hypergraph spectral learning was proposed in [Sun08], to exploit correlation information among labels. A hypergraph is a generalization of the traditional graph in which the edges, called hyperedges, are arbitrary non-empty subsets of the vertex set [Agarwal06]. In this context, the proposed algorithm called, Hypergraph Spectral Learning (HSL), constructs a hypergraph structure by assigning a hyperedge for each label in the label space, and including all instances related to this label into the hyperedge. Afterwards, on the basis of the traditional spectral graph theory, HSL computes the Laplacian of the obtained hypergraph, solves the corresponding eigen problem to retrieve the transformation matrix. Finally, the low-dimensional embedding is obtained by a linear transformation, in which the instance-label relations encoded in the hypergraph are preserved. The authors also proposed to use a least squares formulation for the hypergraph for a better scalability of HSL, especially with large data sets where eigen solutions may be computationally expensive.

Before we round up this chapter, we go through some available tools developed to support mulit-label learning.

3.4 Multi-label data tools

Recently, many tools supporting multi-label learning have been developed and made available to researchers and data mining practitioners. The most used are the following:

- **MULAN**[4]: was developed by [Tsoumakas11] based on the well-known software WEKA [Holmes94]. It is a full-fledged multi-label software which contains a wide spectrum of multi-label tools ranging from prepossessing data sets to the implementation of classification evaluation measures. Several of the most common transformations and mutli-label classification algorithms are already included, ready to use. Unlike Weka, MULAN doesn't have a graphical user interface, but instead offers an open source library written in Java which provides a rich programming interface (API) to support the development of multi-label learning applications.

- **MEKA**[5]: like MULAN, MEKA is also based on WEKA with a very similar graphical user interface [Read15]. MEKA permits through a user-friendly graphical interface to load and save multi-label data sets, as well as to perform various tasks and experiments related to multi-label learning with simple mouse clicks.

- **mldr**[6]: is an R package [Charte15], implementing a large score of functions to load, edit, and save mutli-label data sets, along with general exploratory analysis methods. The package also provides a Web-based graphical user interface from which most tasks can be easily performed.

- **Scikit-multilearn**[7] [Szymański17]: was developed based on the well-known scikit-learn Python module. It offers a native Python implementation of a variety of multi-label classification algorithms. The current version is in 0.0.4 and it provides BR, LP, RAkEL and HOMER; many other methods are still under development.

[4]http://mulan.sourceforge.net/
[5]http://meka.sourceforge.net/
[6]https://github.com/fcharte/mldr
[7]https://pypi.python.org/pypi/scikit-multilearn

3.5 Conclusion

In this chapter, definitions, main application fields and challenges in multi-label learning were presented. More precisely, we have seen how multi-label classification has been introduced to meet the needs of new application domains such as semantic scene annotation, where real-world objects are often classified into more than one category at the same time. What's more, we have investigated various multi-label classification techniques which are roughly categorized into problem transformation or problem adaptation. On the other hand, to alleviate the curse of dimensionality in multi-label settings, numerous solutions were also explored from both feature selection and feature extraction perspectives.

"You hear and you forget; you see and you remember; you do and you understand."

Confucius

4

Laplacian Scores for semi-supervised multi-label feature selection

▷*In this chapter, we propose a new filter framework for solving semi-supervised multi-label feature selection, based on Laplacian Score. To this end, we embraced both problem transformation and algorithm adaptation frameworks. On the one hand, we use Binary Relevance and Label Powerset in conjunction with CLS, to come up with two algorithms named BR-CLS and LP-CLS. On the other hand, we introduce S-CLS by extending CLS to deal with multi-label data directly. In so doing, we show how to constrain the objective function of this score, when data are partially labeled and instances are associated with subsets of labels. In this regard, we transform the labeled part of data into soft constraints and show how to integrate them in a new measure of feature relevance, according to the available labels. Experiments on benchmark data sets are provided for validating the proposed approach and comparing it with some other state-of-the-art feature selection methods in a multi-label context.* ◁

Chapter outline

4.1 Notations **61**

4.2 Transformation-based Approach **62**

4.2.1 BR-CLS 62

4.2.2 LP-CLS 63

4.3 Soft-Constrained Laplacian Score: S-CLS **63**

4.4 Experiments **68**

4.4.1 Data sets and Methods 68

4.4.2 Experimental Settings 70

4.4.3 Results 74

4.5 Conclusion **82**

4.1 Notations

This section presents a mathematical formulation of the semi-supervised multi-label problems that will be used in the rest of this thesis.

In semi-supervised multi-label learning, a data set of N instances $X = \{x_1, ..., x_N\}$ consists of two parts depending on label availability: a supervised part $X_L = \{x_1, ..., x_L\}$ in which instances are associated with subsets of labels from the set $Y_L = \{y_1, ..., y_L\}$, and an unsupervised part $X_U = \{x_{L+1}, ..., x_{L+U}\}$ consisting of completely unlabeled instances, i.e in which all instances are unlabeled (any labeled, even partially, instance would be part of X_L). Worthwhile mentioning, that each $y_i \subset Y_L$ associated with an instance in X_L, is in fact a subset of atomic labels (Figure 4.1).

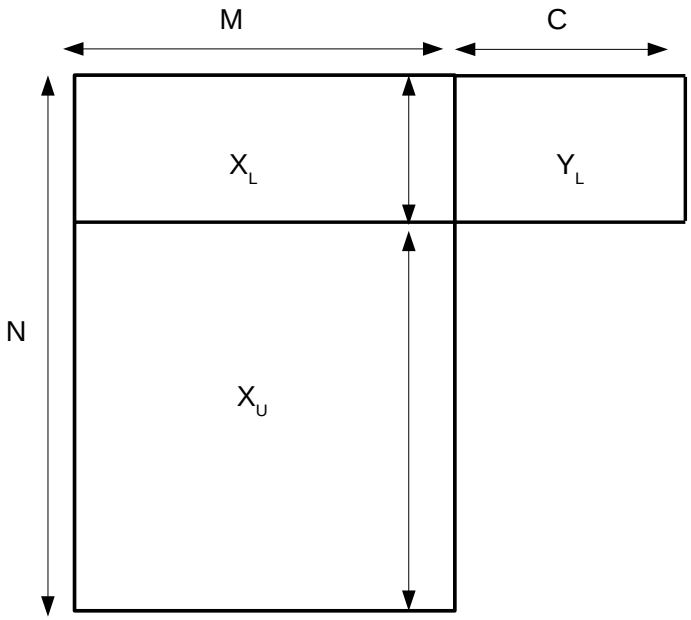


Figure 4.1: Data structure for semi-supervised multi-label learning.

Besides, each data instance x_i is a vector with M dimensions (features), and associated with a vector of labels $y_i = (y_{i1}, y_{i2}, ..., y_{iC})$, if it is in X_L, where $y_{ij} \in \{0, 1\}$, and $L + U = N$ (N is the total number of instances).

Let $F_1, F_2, ..., F_M$ denote the M features of X and $f_1, f_2, ..., f_M$ be the corresponding feature vectors that record the feature value in each instance.

Semi-supervised multi-label feature selection uses simultaneously, X_L, X_U and Y_L to identify the set of most relevant features $F_{j_1}, F_{j_2}, ..., F_{j_h}$ of the target concept, where $h \leq M$ and $j_r \in \{1, 2, ..., M\}$ for $r \in \{1, 2, ..., h\}$.

4.2 Transformation-based Approach

In this section, we propose two algorithms developed on the basis of the Constrained Laplacian Score [Benabdeslem11b] in conjunction with two well-known problem transformation methods. More in particular, by combining CLS with Binary relevance we come up with BR-CLS , and the combination of CLS with Label Powerset gives rise to LP-CLS. Next we give the details of each method.

4.2.1 BR-CLS

BR-CLS combines Binary Relevance and Constrained Laplacian Score to achieve feature selection in multi-labeled data. Initially, we split the labeled part of the data set X_L into C data sets, one for each label y, each of which contains all instances in X_L. Once this has been done, the Constrained Laplacian Score is applied on each data set together with the unlabeled part, yielding C different groups of feature scores. After an ascending sorting of the feature scores in each group, the appropriate ranking of a feature is obtained by averaging the corresponding group rank. Finally, The top-ranked features will be selected, if a desired number of relevant features is fixed. Alogrithm 7 outlines all steps of BR-CLS.

Algorithm 7 BR-CLS

Input: Data set X, the parameters λ and k
1: Transform the Labeled part X_L into C ${X^l}_L$ single-label data sets.
for each $l \in Y$ **do**
  2: Apply CLS on ${X^l}_L$ and X_U
  3: Sort the resulting feature scores in an ascendant order
end for
4: Average the feature ranks over all the data sets ${X^l}_L$.
5 : Sort the resulting average ranks in an ascendant order

Obviously, BR-CLS fails to deal with correlations between labels, since each label is treated separately with each data set. Besides, the scaling-up of BR-CLS is not effective, because the time and space costs will be tremendous.

4.2.2 LP-CLS

The use of Label Powerset as transformation method will allow to consider label correlations, and reduce overheads of the whole selection process. LP-CLS operates in two stages: in the first step, we transform the labeled part X_L of the training data set into a multi-class data set, in which each subset of labels is replaced by a new class. After the transformation step, we call Constrained Laplacian Score with both the resulting multi-class data set and the unlabeled part of the data. The last step will result in a ranked list of feature scores from the most relevant to less relevant. we summarize LP-CLS in Algorithm 8.

Algorithm 8 LP-CLS

Input: Data set X, the parameters λ and k
1: Apply LP on X_L
2: Call CLS on the resulting multi-class data set (obtained from the labeled part X_L) together with the unlabeled part X_U.
3: Rank the features F_r according to their CLS_r in an ascending order.

Obviously, LP-CLS takes into account the dependence between labels. However, challenging issues arise when the number of classes and training examples grows larger. In that case, the computational cost will rise sharply, and most importantly, we may end up with classes associated a tiny number of examples, making the multi-class data imbalanced. Furthermore, LP can only treat label subsets observed in the training set. This is an important limitation, because new label subsets may be found in test data sets. To overcome the limitations of transformation-based approaches, in the next section we introduce a more elaborate solution based on algorithm adaptation.

4.3 Soft-Constrained Laplacian Score: S-CLS

This section describes in detail the theoretical aspects of S-CLS [Alalga16]. We present our approach for semi-supervised feature selection from multi-labeled data.

We propose to extend our prior work, CLS [Benabdeslem11b] to deal with such data. To this end, we provide a generalization to the concept of pairwise constraints, used in semi-supervised learning, to take into account multi-labeled data.

S-CLS represents an adapted version of both scores: Laplacian [He05b] and Constraint based [Zhang08a]. In fact, Laplacian score can be seen as a special version of S-CLS when there are no labels ($X = X_U$), and when ($X = X_L$) S-CLS can be considered as a generalization of constraint score for multi-labeled data. Loosely speaking, S-CLS proposed an efficient combination of both scores in a new score function, exploiting the geometrical structure of unlabeled data and the constraint-preserving ability of labeled data at the same time.

With S-CLS, on the one hand, a relevant feature should be the one on which a pair of instances (neighbors or related by a *must-link: ML* constraint) are close to each other. On the other hand, the relevant feature should be the one with a large variance or on which a pair of instances (related by a *cannot-link: CL* constraint) are far away.

In the single-label context, constraints can be driven from the label information, that is, two instances are in the must-link set if they have the same label and in the cannot-link set otherwise. However, when dealing with multi-label data sets, we cannot simply state that instances belong either to must-link or cannot-link constraint sets. Indeed, we may have partially similar instances, i.e instances having only some labels in common. To capture this notion, we introduce a new parameter that we call *soft constraint* and denote by P.

Before introducing our Soft-Constrained Laplacian Score, some notations are given as follows:

Each instance x_i in the labeled part of data X_L, is associated with a target vector $y_i = (y_{i1}, ..., y_{il}, ..., y_{iC})$, where the l^{th} element is equal to 1 if x_i is labeled by the label l, and 0 otherwise. Given two instances x_i and x_j labeled by y_i and y_j, respectively, the soft constraint is computed as follows:

$$P_{ij} = \frac{1}{C} \sum_{l=1}^{C} \delta(y_{il}, y_{jl}) \tag{4.1}$$

where:

$$\delta(y_{il}, y_{jl}) = \begin{cases} 1 & \text{if } (y_{il} = 1) \wedge (y_{jl} = 1) \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Literally, P_{ij} of two instances is the number of shared labels out of the total number of labels in the label space Y_L. The greater P_{ij}, the more powerful the soft constraint is. $P_{ij} = 1$ means that x_i and x_j must be linked (i.e. are in the must-link set, to give the equivalence with traditional constraints), inversely if $P_{ij} = 0$ x_i and x_j must not be linked (i.e, are in the cannot-link set). Thus, P_{ij} can be considered as a *linkage strength* between x_i and x_j.

The selection is carried out on the whole data set and for each feature F_r (M-dimensional vector $(f_{r1}, .., f_{ri}, .., f_{rM})$). The score function S-CLS (to be minimized) is then defined by:

$$S\text{-}CLS_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_{i,j}(\alpha^i_{rj} - \beta^i_{rj})^2 D_{ii}} \tag{4.3}$$

The matrix S, whose elements are defined by equation(4.4), expresses the pairwise similarity between instances. If two instances x_i and x_j are both labeled (i.e. belong to X_L), S_{ij} is evaluated to $e^{-\frac{\|x_i - x_j\|^2}{\lambda}} P_{ij}$. If one or both are unlabeled, and if x_i is among the k-nearest neighbors of x_j (or vice-versa), $S_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\lambda}}$. Else $S_{ij} = 0$.

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\lambda}} & \text{if } x_i \text{ and } x_j \text{ are neighbors} \\ & \text{and } (x_i \in X_U) \wedge (x_j \in X_U) \\ e^{-\frac{\|x_i - x_j\|^2}{\lambda}} P_{ij} & \text{if } (x_i \in X_L) \wedge (x_j \in X_L) \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

D is a diagonal matrix, where:

$$D_{ii} = \sum_{j=1}^{N} S_{ij} \tag{4.5}$$

λ is a tuning parameter, and the terms α_{ri} and β_{rj} are defined by equations (4.6) and (4.7), respectively.

$$\alpha_{rj}^{i} = \begin{cases} f_{ri}(1 - P_{ij}) & \text{if } (x_i \in X_L) \wedge (x_j \in X_L) \\ f_{ri} & \text{otherwise} \end{cases} \tag{4.6}$$

$$\beta_{rj}^{i} = \begin{cases} f_{rj}(1 - P_{ij}) & \text{if } (x_i \in X_L) \wedge (x_j \in X_L) \\ \mu_r & \text{otherwise} \end{cases} \tag{4.7}$$

Where μ_r is the weighted data mean of the feature F_r:

$$\mu_r = \sum_{i=1}^{N} (f_{ri} \frac{D_{ii}}{\sum_i D_{ii}}) \tag{4.8}$$

The interpretation of S-CLS is straightforward. Actually, features for which similar instances have close values will receives lower scores, and thus will be selected as the most relevant. More specifically, for the supervised subset: X_L, the best features are those with close values for instances sharing large strength linkage (i.e tend to be in the must-link set), and inversely large values for instances with low strength linkage (i.e. tend to be in the cannot-link set). As for the unsupervised subset: X_U, relevant features will have large variance in instances fart apart in terms of Euclidean Distance.

Note that if two instances x_i and x_j are labeled by all labels, then their label vectors will be $y_i = (1, 1, ..., 1)$ and $y_j = (1, 1, ..., 1)$, respectively. Thus, $P_{ij} = 1$ (which is equivalent to a must-link constraint). In this case, in the numerator, S_{ij} is maximized and the denominator is not increased. For a good feature F_r, $(f_{ri} - f_{rj})$ must be smaller, and thus S-CLS tends to be small.

In contrast, if $x_i \in X_L$ and $x_j \in X_L$, and there is no common label between them, then $P_{ij} = 0$ (which is equivalent to a cannot-link constraint). Thus, the numerator is not increased and in the denominator, $(f_{ri} - f_{rj})$ must be bigger for a good feature F_r.

Subsequently, in these two cases, S-CLS would be equivalent to CLS [Benabdeslem11b]. Another remark is that, if there is no label at all ($L = 0, X = X_U$), S-CLS would be equivalent to the Laplacian score [He05b].

Algorithm 9 S-CLS

Input: Data set X, the parameters λ and k
1: Construct the matrix of soft constraints P_{ij} from the label space Y_L, according to eq (4.1)
2: Calculate the similarity and diagonal matrices S_{ij}, D_{ii} according to eq(4.4) and eq(4.5), respectively.
for $r = 1$ **to** M **do**
  3: Calculate S-CLS$_r$ according to eq (4.3)
end for
4: Rank the features F_r according to their S-CLS$_r$ in ascending order.

The whole procedure is summarized in Algorithm 9. Note that this algorithm is computed in time $\mathcal{O}(Mmax(N^2, log\, M))$. Indeed, the first step of the algorithm requires L^2 operations. Step 2 builds the matrices requiring N^2 operations. Step 3 evaluates the M features requiring MN^2 operations, while the last step ranks features according to their scores with $M\ Log(M)$ operations.

To reduce this complexity, we propose that we apply a clustering on X_U, in a similar way to [Benabdeslem11b]. The idea is to substitute this huge volume of data by a smaller one $X'_U = (u_1, ..., u_K)$, while at the same time preserving the geometric structure of X_U, where K is the number of clusters. We propose using the Self-Organizing Map (SOM) based clustering [Kohonen01], for its ability to preserve the topological relationship of data and thus the geometric structure of their distribution. With this strategy, we reduce the complexity to $O(Mmax(U, log M))$, where U is the size of X_U. Figure 4.2 outlines the framework of feature selection used in S-CLS.

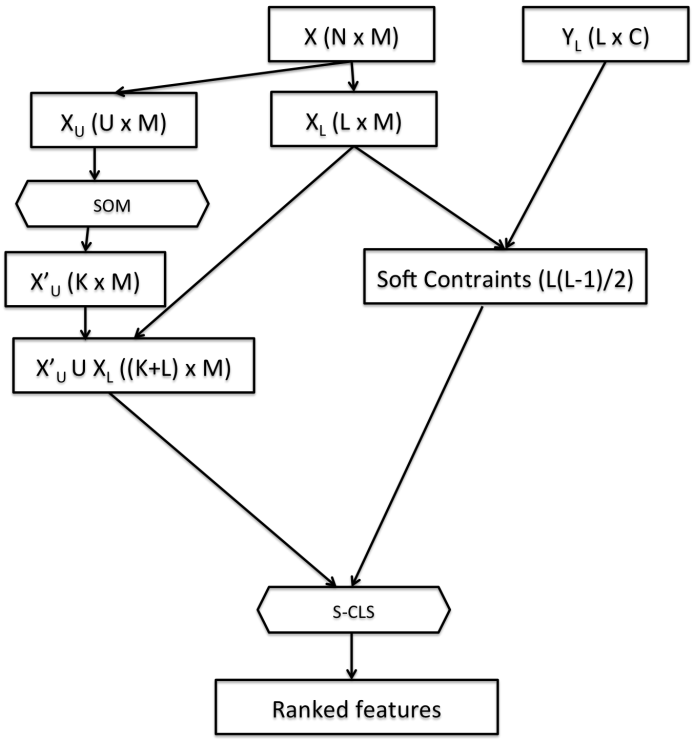


Figure 4.2: General framework of S-CLS.

4.4 Experiments

In this section, we present the data sets and algorithms, as well as the evaluation measures used to conduct the empirical studies. The results of the different experiments are illustrated by figures and tables. At the end of this section, we give a discussion to help understand the results obtained.

4.4.1 Data sets and Methods

Before presenting the data sets used in this work, two important parameters need to be discussed. The multi-label nature of a data set is commonly quantified by its cardinality defined by equation (4.9) and the density defined by equation (4.10). The first parameter represents the average number of labels in each instance, while the second parameter is simply a normalized cardinality. In fact, it has been widely argued that two data sets with roughly the same cardinality but with a

great difference in density could lead to different behaviors in multi-label learning methods.

Let $\mathcal{D}$ be the whole data set formed by the data space X and the label space Y. Label cardinality and the density can be calculated as follows:

$$LC(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \|y_i\|_1 \tag{4.9}$$

$$LD(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \frac{\|y_i\|_1}{C} \tag{4.10}$$

where $\|.\|_1$ is the l_1-norm.

To assess the efficiency of our method, we conducted the experiments on eight benchmark data sets from various domains ranging from text categorization to protein function analysis. **"Birds"** is a collection of bird sounds from different species containing 645 instances used in sound recognition systems [Briggs12]. **"Emotions"** consists of 593 songs, each of which is associated with different emotional reactions [Trohidis08]. **"Scene"** is widely used in literature to illustrate semantic scene annotation tasks [Boutell04]. **"Yeast"** has been created in biology to capture different functional classes of genes [Elisseeff01]. **"Slashdot"** is a collection of article blurbs with subject categories collected from http://slashdot.org. **"LangLog"** was generated from the Language Log Forum, which discussed various topics relating to language: the label space consists of 75 different topics. The last two data sets are widely used in text categorization tasks [Read12]. **"Plant"**, This data set is composed of sequences of plant protein, with 440 features for each protein sequence: There are 12 location sites representing the different class labels. **"Human"** is composed of sequences of human protein, with 440 features for each protein sequence, there are 14 location sites. The main task, in the last two data sets, is to predict the subcellular locations of proteins according to their sequences [Xu13]. Detailed statistics about these data sets[1] are shown in Table4.1.

To evaluate performance of S-CLS, we compare it with three recent multi-label feature selection methods, namely: PPT+MI [Doquire13b], ML-MI [Lee13] and Memetic [Lee15]. The two first ones are filter methods based on mutual information, whereas Memetic is a wrapper method based on a new evolutionary algo-

[1]All data sets are readily available at: http://mulan.sourceforge.net/datasets-mlc.html

Table 4.1: Description of the data sets used in the experiments

Data sets	domain	instances	features	labels	cardinality	density
Birds	audio	645	260	19	2.402	0.015
Emotions	music	593	72	6	1.869	0.311
Scene	image	2407	294	6	1.074	0.179
Yeast	biology	2417	103	14	1.237	0.303
Slashdot	text	3782	1079	22	1.18	0.053
Langlog	text	1460	1004	75	1.18	0.0015
Plant	biology	948	440	12	1.08	0.089
Human	biology	3108	440	14	1.19	0.084

rithm. More specifically, PPT+MI is a transformation-based method, which combines Pruned Powerset Transformation with MI (Mutual Information) to perform feature selection. ML-MI is an adaptation-based method that uses a multi-variate mutual information in order to achieve feature selection in multi-labeled data sets without resorting to any prior transformation. On the other hand, Memetic feature selection makes use of a new generation of genetic algorithms called Memetic along with a performance measure, as a fitness function, to select the best subset of features. As mentioned in the introduction, being a wrapper method, this approach, apart from being time-consuming, is tightly dependent on the classifier used in assessing the selected feature subsets, hence the results are often biased by the choice of the classifier.

Due to the lack of readily available semi-supervised multi-label feature selection algorithms to compare with, we included in the experiments a feature extraction algorithm which uses semi-supervised multi-label data to achieve dimensionality reduction, namely SSDR-MC [Qian10]. This algorithm is intended to be used as a dimension reduction tool and a multi-label classifier. To this end, SSDR-MC first finds the proper projection of the original data set into a lower dimension, afterwards it gradually fills in the missing labels of the unlabeled part by propagating the class labels from the supervised.

4.4.2 Experimental Settings

S-CLS exploits small-labeled-sample data sets to achieve feature selection, to simulate this context, we randomly select 10% of instances from original data sets as labeled data, while the remaining instances are used as unlabeled data. On the

other hand, the parameter λ in the objective function of S-CLS should be carefully tuned according to the problem at hand. If overestimated, the function in eq (4.4) will behave almost linearly, and lose its non-linear power. If underestimated, it will lack regularization, and the decision boundary will be highly sensitive to noise. In the experiments, we set:

$$\lambda = percentile(\{\|x_i - x_j\|^2; i, j = 1..N\}; 20). \tag{4.11}$$

The k-neighborhood parameter for eq (4.4) is set at 10 for all experiments.

Each stage of the experiments is divided in two steps: feature selection and classification. In the feature selection step, S-CLS is provided with 10% of labeled data along with 90% of unlabeled data, whereas the other feature selection algorithms use a wholly supervised data sets (100% of supervision, which is an advantage for them). Afterwards in the classification step, a base classifier is used with completely labeled data sets (100% supervised) and uses all features as a baseline in one hand, and only features selected by different feature selection algorithms on the other hand. The classier is tuned via 3-fold cross-validation on each data set. Classifiers and different evaluation metrics are discussed below.

4.4.2.1 Evaluation with *ML-kNN.*

This classifier is a multi-label version of the traditional *kNN* algorithm. The prediction is straightforward: for an unseen instance, the algorithm firstly fetches its k nearest neighbors. Secondly, for each label in the label space, we compute its frequency in those neighbors. Finally, the set of labels of the unseen instance is determined by the maximum a posteriori principle (MAP)[Zhang07b].

ML-kNN belongs to the category of lazy learning methods in which no model of the classifier is built, and in which generalization is postponed until a request is submitted. These methods are very sensitive to irrelevant features and are widely used to measure performances of feature selection algorithms. In this research, we have chosen to use *ML-kNN* to carry out our experiments.

Along with *ML-kNN* we used other state-of-the art multi-label classifiers to show benefits of feature selection in multi-label classification. In what follows we give a brief description of them:

RaKEL: Is an ensemble method based on Label Powerset transformation. *RaKEL* builds an ensemble of LP classifiers each of which is trained on different small random subset of the label space [Tsoumakas07b].

ECC (Ensemble of Classifier Chains): is based on Binary Relevance transformation, and uses a chain of binary classifiers each of which is trained upon the prediction of previous ones [Read11].

HOMER (Hierarchy of Multi-label ClassifiERs): Constructs a hierarchy of Multi-label classifiers each of them deals with a smaller subset of labels [Tsoumakas08].

IBRL (Instance Based Logistic Regression): Combines instance-based learning and logistic regression. It considers the labels of neighboring instances as features of an unseen instance and reduces instance-based learning to logistic regression [Cheng09].

4.4.2.2 Evaluation Metrics.

Evaluation measures in a multi-label context are different from those used with single-label learning. The former has to manage the new concept of partially correct labels, as we deal with subsets of labels rather than single labels. Usually, multi-label evaluation measures are either instance-based or label-based. Instance-based measures are calculated separately for each test instance and averaged across the test set, whereas label-based measures are calculated separately for each label and then averaged across all labels.

Let $h : X \to Y$ be a classification hypothesis, which for each instance $x_i \in X$, predicts its vector of labels $h(x_i) \in Y$. The most commonly used evaluation measures are defined as follows:

1. Hamming loss: Indicates how many times an instance-label pair is misclassified [Schapire00].

$$Hamming\,Loss(h, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\|h(x_i) \oplus y_i\|_1}{C} \tag{4.12}$$

where $h(x_i) \oplus y_i$ stands for the XOR operation between the predicted label-vector $h(x_i)$ and the true label-vector y_i.

2. Subset 0/1 loss: A strict measure that requires complete equality between the predicted and true sets of labels. A function I is defined on $\{true, false\}$ and takes values on $\{0, 1\}$.

$$0/1\ loss(h, \mathcal{D}) = 1 - \frac{1}{N}\sum_{i=1}^{N} I(h(x_i) = y_i) \tag{4.13}$$

3. Macro-F: Averages the F-measure on the predictions of different labels.

$$Macro - F(h, \mathcal{D}) = \frac{1}{C}\sum_{l=1}^{C} \frac{2\sum_{i=1}^{N} h^l(x_i) y_i^l}{\sum_{i=1}^{N} y_i^l + \sum_{i=1}^{N} h^l(x_i)} \tag{4.14}$$

4. Micro-F: Calculates the F-measure on the predictions of different labels as a whole.

$$Micro - F(h, \mathcal{D}) = \frac{2\sum_{i=1}^{N} \|h(x_i) \wedge y_i\|_1}{\sum_{i=1}^{N} \|y_i\|_1 + \sum_{i=1}^{N} \|h(x_i)\|_1} \tag{4.15}$$

Other types of measures are employed to evaluate the performance of label ranking. These measures require a real-valued output function $f : XY \rightarrow R$. $f(.,.)$ can be transformed to a ranking function $rank_f(.,.)$, which maps the outputs of $f(x_i, y_s)$ for any $y_s \in Y$ to $\{1, 2, ..., C\}$ such that if $f(x_i, y_s) > f(x_i, y_t)$ then $rank_f(x_i, y_s) < rank_f(x_i, y_t)$. We note Y_i to indicate the subset of Y corresponding to the label vector y_i.

5. Average Precision: Computes the average fraction of labels ranked above a particular label $y \in Y_i$ [Salton91].

$$Avgprec(f, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|\{y' \in Y_i | rank_f(x_i, y') \leq rank_f(x_i, y)\}|}{rank_f(x_i, y)} \quad (4.16)$$

$|.|$ denotes the cardinality of a set.

6. One error: Evaluates how many times the top-ranked label is not in the set of ground-truth labels of the instances.

$$One - Error(f, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} I([arg\ \underset{y\in Y}{max} f(x_i, y)] \notin Y_i) \quad (4.17)$$

where $I(x)$ is the indicator function which equals 1 if x holds and 0 otherwise.

7. Coverage: Evaluates how far, on average, we need to go down the label ranking list to cover all the ground-truth labels of the instance.

$$Coverage(f, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \underset{y\in Y_i}{max}\ rank_f(x_i, y) - 1 \quad (4.18)$$

8. Ranking loss: Evaluates the average fraction of label pairs that are not correctly ordered.

$$Ranking\ Loss(f, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\overline{Y_i}|}|\{(y_s, y_t) \in Y_i\overline{Y_i}; f(x_i, y_s) \leq f(x_i, y_t)\}| \quad (4.19)$$

where $\overline{Y_i}$ denotes the complementary set of Y_i in Y.

9. AUC: Area under ROC (stands for Receiver Operating Characteristic) curve is a criterion for measuring the quality of ranking, and more specifically the probability that a random positive instance is ranked before a random negative one.

$$AUC_{total} = \frac{1}{C}\sum_{j=1}^{C} AUC_j \quad (4.20)$$

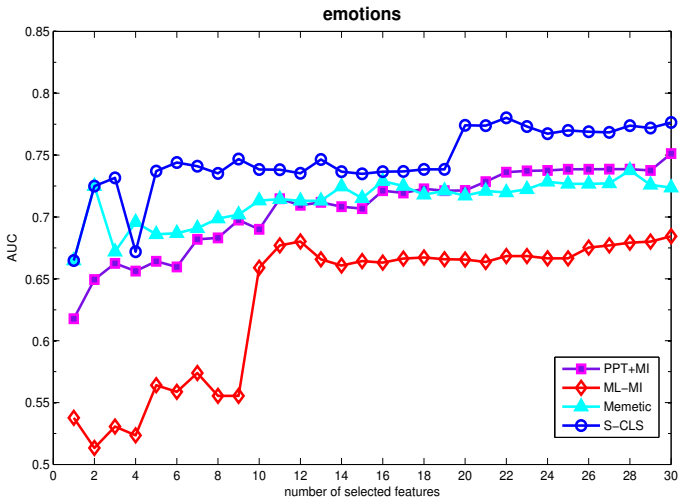
4.4.3 Results

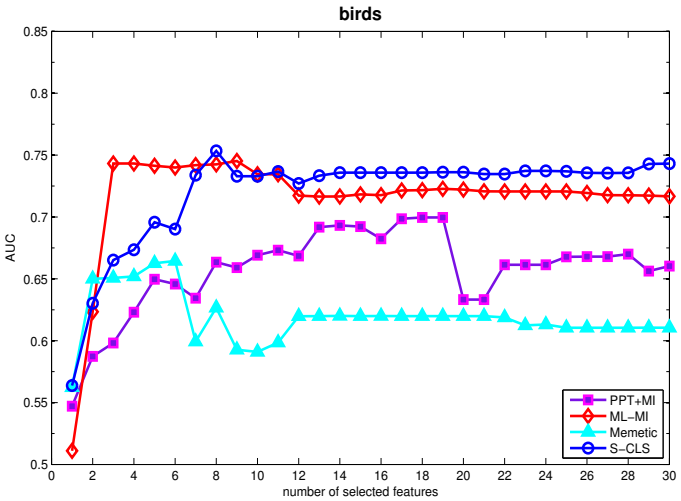
First and foremost, to verify the potential of our method, experiments were carried out in three different scenarios: in the first one, we used *ML-kNN* as a base classifier using all features in the baseline case, and on the other hand using only features selected by different feature selection algorithms beforehand. In the second scenario we compared between different classifiers using all features on one hand and the best subset of top ranked features selected by S-CLS on the other hand. Finally we wrap up the experiments by using statistical tests to help have a better view on the comparisons between different feature selection approaches. We must underline that, in the selection step, apart from S-CLS all competing methods use 100% labeled data sets(which favors them), whereas S-CLS achieves feature selection with only 10% of supervision(90% of each data set is unlabeled).

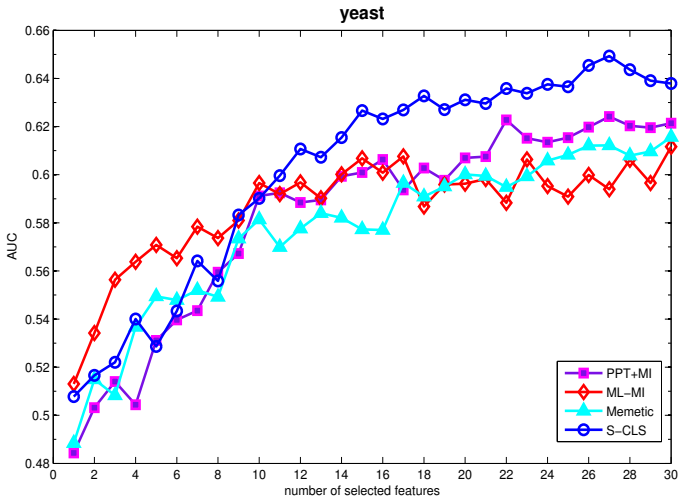
It is also important to emphasize that our primary concern is not to prove the classification power neither of *ML-kNN* nor any other classifier we used, but to show benefits of S-CLS in boosting classification tasks. Therefore, the choice of a particular classifier is not important per se, quite the contrary it is generally advised to use lazy classifiers like *kNN* in feature selection experiments. Besides, it is worthwhile mentioning that since S-CLS is a semi-supervised feature selection algorithm, that is, it uses both supervised and unsupervised data, selected features can serve both in classification and clustering tasks.

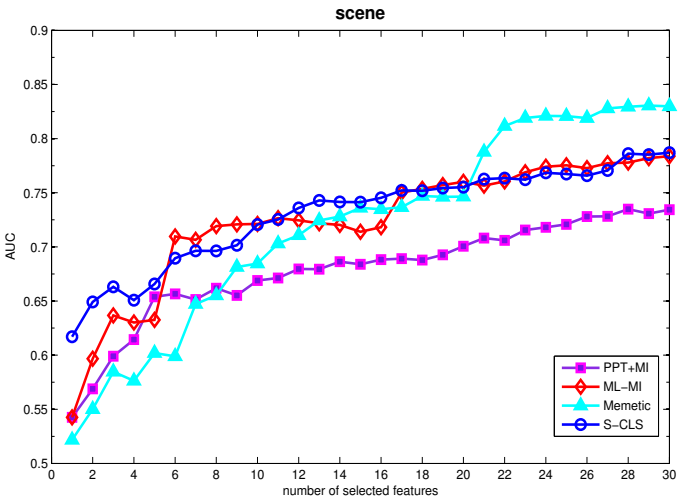
As mentioned above, in the first scenario of experiments, we used *ML-kNN* as a multi-label base classifier along with eight benchmark data sets. Results are reported on Figure 4.2 that represents values of AUC for an increasing subsets built from 30-top-ranked features (selected features) in each data set.

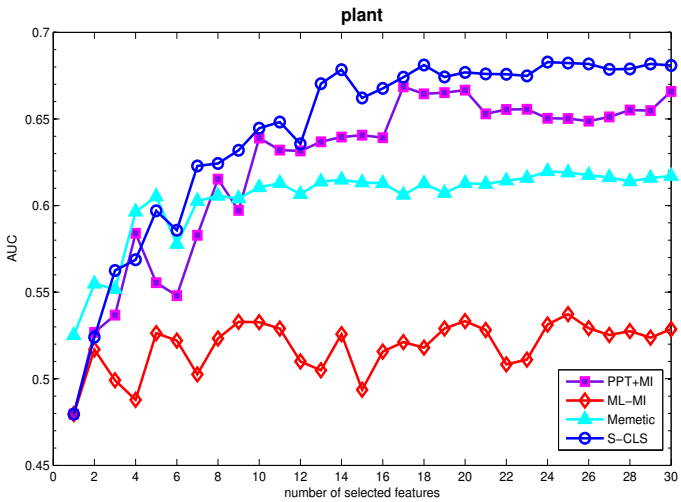
Figure 4.2 shows classification performances of the *ML-kNN* algorithm in terms of Area under ROC (AUC), using an increasing number of features, which are selected by our proposed method S-CLS (curve in blue), PPT+MI, ML-MI, and Memetic. As can be observed, S-CLS generally outperforms its competitors in all data sets, apart from in “Scene”, where Memetic performs slightly better, and Langlog where S-CLS is comparable to ML-MI. In the other data sets S-CLS is either followed by PPT+MI, ML-MI or Memetic.

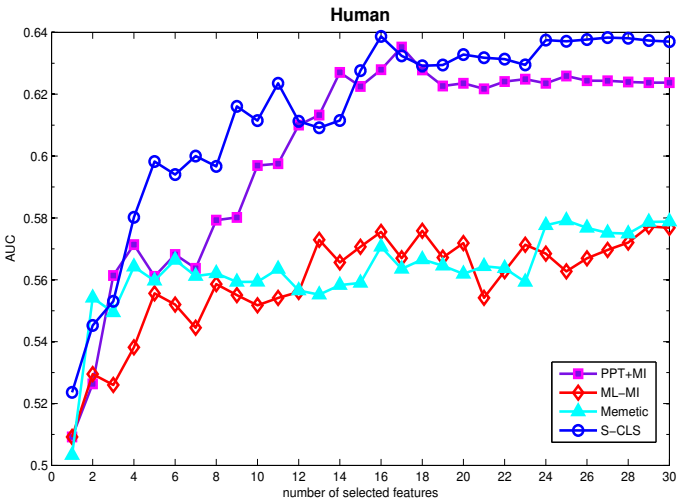

emotions
AUC
number of selected features
PPT+MI
ML−MI
Memetic
S−CLS


birds
AUC
number of selected features
PPT+MI
ML−MI
Memetic
S−CLS


yeast
AUC
number of selected features
PPT+MI
ML−MI
Memetic
S−CLS


scene
AUC
number of selected features
PPT+MI
ML−MI
Memetic
S−CLS


plant
AUC
number of selected features
PPT+MI
ML−MI
Memetic
S−CLS


Human
AUC
number of selected features
PPT+MI
ML−MI
Memetic
S−CLS

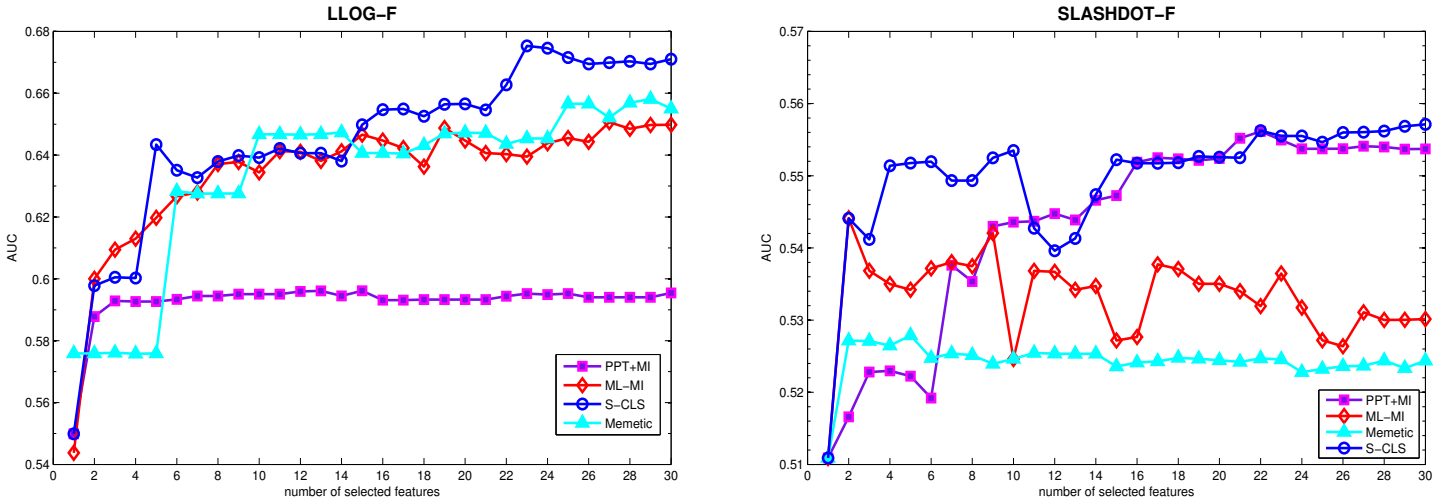


Figure 4.2: AUC v.s. number of selected features

Table 4.2 lists mean values of AUC, and thus summarises Figure 4.2. In the second step of this scenario, we use Table 4.2 to conduct statistical tests according to the methodology proposed in [Demšar06, García10]. The intent is to determine whether there is enough evidence to "reject" the null hypothesis suggesting that all feature selection methods perform equally. To this end, we first use the non-parametric Friedman test. The analysis is based on ranking multiple algorithms on multiple data sets, whereby each feature selection algorithm is ranked for each data set separately. According to a given evaluation measure, this ranking is done in an ascending order, from the best-performing to the worst-performing algorithm. In case of ties, we assign average ranks. After calculating the Friedman statistic, we compare it to a critical value for a given level of significance, in order to accept or reject the initial null hypothesis.

Table 4.2: AUC and average rank

Data set	PPT+MI	ML-MI	Memetic	S-CLS
Birds	0.66±0.03	0.71±0.04	0.62±0.02	**0.72±0.04**
Emotions	0.71±0.03	0.63±0.06	0.71±0.03	**0.75±0.03**
Scene	0.68±0.05	0.72±0.06	0.72±0.09	**0.73±0.04**
Yeast	0.58±0.04	0.58±0.03	0.57±0.03	**0.60±0.03**
Slashdot	0.54±0.01	0.53±0.01	0.52±0.00	**0.55±0.01**
Langlog	0.54±0.02	**0.63±0.02**	0.61±0.04	**0.63±0.03**
Plant	0.62±0.05	0.52±0.01	0.58±0.01	**0.64±0.05**
Human	0.60±0.03	0.56±0.02	0.56±0.01	**0.61±0.03**
Av Rank	3,1250	2.8750	3.3125	**1.0625**

If the differences between algorithms are statistically significant, we secondly proceed with post hoc tests to have a zoomed-in view on the nature of these differences. In our case, we use Nemenyi test to perform a pairwise comparison between feature selection algorithms. The Nemenyi test is based on computing a q statistic over the difference in average ranks of classifiers. Performance of two compared classifiers are considerably different if the difference between their corresponding average ranks is larger than or equal a Critical Distance (CD).

The critical value and the Critical Distance in the previous tests depend on the number of algorithms in the comparison, the number of data sets, as well as the chosen significance level (set to 0.05 in our experiments). Note that SSDR-MC has been excluded from both Figure 4.2 and Table 4.2, because it is a feature extraction algorithm that doesn't return a subset of original features. Instead, it artificially create a new feature dimension, and hence is not suitable for this particular scenario of experiment.

Results showed that the Friedman test rejected the null hypothesis and revealed a considerable differences among feature selection algorithms when applied as preprocessing step to *ML-kNN* (with p-value $= 0.0023 < 0.05$).

Results for the Nemenyi test are shown on Figure 4.3. When the distance between ranks of two algorithms is smaller than the computed critical distance (Here CD $= 1.0625$), there will be a line connecting them on the resulting diagram. As can be seen, S-CLS tops the ranking, with no connections to other algorithms, meaning that it does statistically differ from others. PPT+MI ranks second, and Memetic and ML-MI rank third and forth respectively, with no significant statistical difference between them ($Distance < CD$).

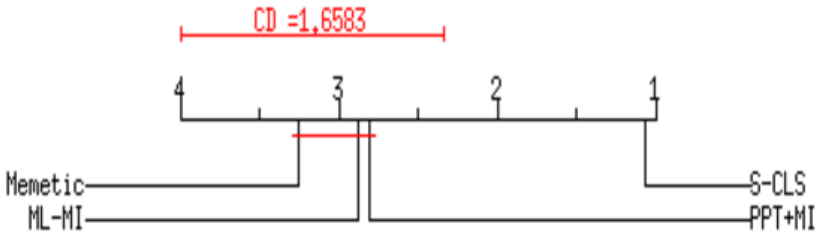


Figure 4.3: Nemenyi test diagram

Worthwhile mentioning that Memetic performs very well on certain data sets, and even outperforms S-CLS on "Scene" but fails dramatically on others, which

explains its rank.

Considering the fact that classifiers react differently to evaluation metrics, that is, a classifier can show good results with respect to Hamming loss for instance, but does not perform well vis-a-vis Micro-F. In the second scenario, we decided to use all known measures, to ensure a better view in terms of comparison. The comparison is conducted by measuring both label-classification and label-ranking performances. Table 4.3 details results on eight data sets using all evaluation measures (384 values), obtained by using exactly the 30-top-ranked feature all at once, except for Memetic where we use the whole subset of features that it has selected, which can exceed 30 features. Besides, SSDR-MC has been forced to yield data sets with exactly 30 features in order to be suitable for the comparisons. Measures are listed with their standard deviation.

Table 4.3: Results (mean±std.) on all data sets used, over all measures ("↘ indicates the smaller the better"; "↗ indicates the larger the better").

Data sets	Feature Selection Algorithm					
	Baseline	PPT+MI	ML-MI	SSDR-MC	Memetic	S-CLS
	Hamming Loss↘					
Birds	0.05±0.006	0.05±0.005	0.05±0.006	0.06±0.000	0.05±0.004	0.05±0.005
Emotions	0.27±0.018	0.27±0.013	0.28±0.021	0.35±0.001	0.27±0.010	0.25±0.019
Scene	0.11±0.008	0.19±0.017	0.18±0.023	0.23±0.003	0.12±0.016	0.19±0.009
Yeast	0.20±0.003	0.21±0.001	0.21±0.002	0.25±0.001	0.21±0.005	0.20±0.004
Slashdot	0.02±0.002	0.02±0.001	0.02±0.001	0.03±0.001	0.02±0.001	0.02±0.002
Langlog	0.16±0.009	0.19±0.009	0.17±0.009	0.15±0.001	0.17±0.011	0.17±0.009
Plant	0.09±0.002	0.09±0.003	0.09±0.002	0.11±0.003	0.09±0.002	0.09±0.002
Human	0.08±0.000	0.08±0.001	0.08±0.000	0.09±0.000	0.08±0.000	0.08±0.001
	0/1 Loss↘					
Birds	0.54±0.039	0.53±0.032	0.53±0.033	0.55±0.001	0.54±0.036	0.51±0.039
Emotions	0.85±0.007	0.75±0.072	0.87±0.022	0.96±0.002	0.80±0.039	0.72±0.055
Scene	0.38±0.044	0.95±0.035	0.83±0.054	0.93±0.004	0.50±0.079	0.49±0.090
Yeast	0.78±0.041	0.83±0.023	0.85±0.025	0.90±0.000	0.84±0.020	0.78±0.011
Slashdot	0.29±0.025	0.26±0.018	0.24±0.014	0.27±0.002	0.23±0.007	0.25±0.012
Langlog	0.84±0.022	0.83±0.022	0.83±0.020	0.95±0.000	0.83±0.018	0.84±0.023
Plant	0.94±0.011	0.91±0.043	0.99±0.007	0.99±0.000	0.97±0.012	0.91±0.021
Human	0.92±0.022	0.90±0.009	0.94±0.017	0.99±0.001	0.98±0.017	0.93±0.016
	Macro-F↗					
Birds	0.04±0.011	0.11±0.052	0.11±0.048	0.06±0.002	0.08±0.011	0.23±0.012
Emotions	0.43±0.020	0.51±0.032	0.42±0.015	0.30±0.001	0.47±0.007	0.54±0.042
Scene	0.62±0.031	0.09±0.063	0.25±0.079	0.10±0.000	0.59±0.063	0.62±0.111
Yeast	0.63±0.016	0.60±0.003	0.58±0.014	0.51±0.000	0.59±0.009	0.63±0.008
Slashdot	0.70±0.017	0.76±0.009	0.76±0.008	0.68±0.001	0.75±0.003	0.75±0.009
Langlog	0.53±0.005	0.41±0.039	0.47±0.003	0.20±0.001	0.49±0.010	0.53±0.019
Plant	0.11±0.029	0.15±0.057	0.01±0.016	0.10±0.002	0.06±0.024	0.15±0.033
Human	0.14±0.030	0.18±0.011	0.12±0.033	0.02±0.003	0.05±0.035	0.14±0.017

Continued on next page

Table 4.3 – *Continued from previous page*

Data sets	Feature Selection Algorithm					
	Baseline	PPT+MI	ML-MI	SSDR-MC	Memetic	S-CLS
			Micro-F↗			
Birds	0.02±0.003	0.05±0.023	0.04±0.026	0.03±0.001	0.03±0.003	0.07±0.013
Emotions	0.33±0.018	0.44±0.008	0.33±0.021	0.25±0.001	0.39±0.007	0.49±0.041
Scene	0.50±0.044	0.08±0.049	0.19±0.061	0.13±0.004	0.47±0.067	0.50±0.085
Yeast	0.35±0.023	0.29±0.008	0.28±0.015	0.23±0.000	0.29±0.010	0.35±0.011
Slashdot	0.04±0.001	0.04±0.001	0.05±0.008	0.04±0.001	0.04±0.006	0.04±0.000
Langlog	0.26±0.013	0.15±0.028	0.19±0.002	0.07±0.000	0.20±0.008	0.27±0.010
Plant	0.04±0.018	0.06±0.016	0.01±0.005	0.01±0.001	0.03±0.007	0.06±0.014
Human	0.06±0.008	0.06±0.006	0.03±0.005	0.04±0.002	0.02±0.009	0.06±0.012
			Average Precision↗			
Birds	0.38±0.018	0.44±0.011	0.46±0.015	0.32±0.002	0.43±0.027	0.57±0.020
Emotions	0.69±0.010	0.72±0.020	0.69±0.021	0.55±0.001	0.71±0.044	0.73±0.041
Scene	0.82±0.017	0.54±0.004	0.60±0.029	0.44±0.006	0.79±0.025	0.78±0.014
Yeast	0.76±0.009	0.73±0.003	0.73±0.004	0.68±0.000	0.73±0.004	0.76±0.008
Slashdot	0.88±0.004	0.88±0.006	0.88±0.003	0.84±0.000	0.88±0.005	0.88±0.005
Langlog	0.61±0.011	0.63±0.018	0.62±0.011	0.41±0.004	0.62±0.005	0.62±0.011
Plant	0.56±0.009	0.56±0.019	0.49±0.020	0.50±0.001	0.54±0.010	0.56±0.026
Human	0.56±0.007	0.59±0.006	0.54±0.012	0.50±0.001	0.55±0.008	0.58±0.006
			One Error↘			
Birds	0.73±0.036	0.68±0.034	0.66±0.035	0.75±0.002	0.68±0.015	0.51±0.014
Emotions	0.42±0.013	0.38±0.044	0.41±0.033	0.63±0.001	0.41±0.055	0.37±0.038
Scene	0.30±0.030	0.69±0.021	0.61±0.056	0.80±0.008	0.34±0.047	0.35±0.031
Yeast	0.23±0.008	0.25±0.001	0.25±0.004	0.31±0.000	0.26±0.006	0.23±0.013
Slashdot	0.09±0.004	0.09±0.004	0.10±0.003	0.08±0.003	0.09±0.005	0.09±0.005
Langlog	0.19±0.020	0.19±0.054	0.18±0.012	0.41±0.013	0.17±0.017	0.16±0.022
Plant	0.64±0.018	0.62±0.023	0.71±0.031	0.71±0.002	0.67±0.008	0.62±0.032
Human	0.61±0.009	0.59±0.012	0.64±0.014	0.62±0.011	0.64±0.003	0.60±0.015
			Coverage↘			
Birds	3.80±0.365	3.19±0.391	2.92±0.300	3.96±0.018	3.13±0.214	2.44±0.205
Emotions	2.37±0.076	2.16±0.064	2.34±0.145	3.29±0.001	2.25±0.082	2.07±0.117
Scene	0.62±0.053	1.82±0.041	1.50±0.047	2.45±0.025	0.76±0.041	0.78±0.042
Yeast	6.35±0.113	6.66±0.053	6.66±0.082	7.42±0.000	6.63±0.094	6.35±0.102
Slashdot	1.02±0.021	1.02±0.036	0.99±0.026	0.97±0.001	0.98±0.035	1.02±0.034
Langlog	48.75±1.82	49.23±1.71	49.67±1.80	50.31±0.00	49.68±2.29	48.67±2.24
Plant	2.36±0.058	2.27±0.114	2.75±0.121	2.75±0.005	2.45±0.078	2.29±0.150
Human	2.41±0.047	2.34±0.045	2.63±0.071	2.97±0.003	2.60±0.100	2.40±0.022
			Ranking Loss↘			
Birds	0.31±0.001	0.26±0.004	0.24±0.009	0.34±0.002	0.25±0.011	0.19±0.010
Emotions	0.28±0.010	0.24±0.014	0.27±0.017	0.46±0.001	0.25±0.037	0.22±0.038
Scene	0.11±0.010	0.34±0.004	0.28±0.012	0.47±0.005	0.13±0.013	0.13±0.007
Yeast	0.17±0.006	0.19±0.002	0.19±0.004	0.23±0.000	0.19±0.004	0.17±0.005
Slashdot	0.05±0.002	0.05±0.004	0.05±0.002	0.05±0.001	0.05±0.004	0.05±0.003
Langlog	0.20±0.009	0.19±0.010	0.20±0.008	0.22±0.005	0.20±0.010	0.20±0.010
Plant	0.20±0.007	0.19±0.012	0.24±0.012	0.24±0.003	0.21±0.009	0.19±0.017
Human	0.16±0.003	0.16±0.003	0.18±0.005	0.20±0.002	0.18±0.007	0.16±0.001

Table 4.4: Effects of S-CLS on different classifiers. Before and after selection.

Data set	RakEL	IBLR	Homer	ECC	ML-kNN
	Before | After	Before | After	Before | After	Before | After	Before | After
Birds	0.69±0.01|0.70±0.01	0.72±0.01|0.73±0.03	0.61±0.04|0.61±0.04	0.78±0.03|0.79±0.02	0.75±0.02|0.78±0.04
Emotions	0.80±0.01|0.81±0.01	0.83±0.02|0.83±0.01	0.67±0.02|0.67±0.02	0.80±0.01|0.81±0.01	0.81±0.02|0.81±0.01
Scene	0.82±0.01|0.77±0.00	0.90±0.00|0.89±0.00	0.75±0.01|0.76±0.05	0.92±0.00|0.92±0.00	0.93±0.00|0.88±0.00
Yeast	0.63±0.00|0.64±0.02	0.69±0.00|0.70±0.01	0.56±0.01|0.57±0.00	0.67±0.02|0.68±0.01	0.68±0.01|0.69±0.01
Slashdot	0.54±0.00|0.55±0.00	0.54±0.02|0.59±0.01	0.55±0.01|0.55±0.01	0.58±0.00|0.59±0.00	0.53±0.01|0.55±0.01
Langlog	0.70±0.01|0.70±0.00	0.72±0.00|0.72±0.01	0.58±0.01|0.58±0.01	0.76±0.01|0.76±0.01	0.70±0.03|0.70±0.02
Plant	0.59±00.2|0.62±0.00	0.67±0.00|0.69±0.02	0.49±0.02|0.53±0.02	0.62±0.01|0.67±0.01	0.61±0.00|0.65±0.02
Human	0.58±0.00|0.58±0.00	0.62±0.01|0.62±0.01	0.52±0.01|0.53±0.02	0.62±0.02|0.63±0.01	0.57±0.01|0.62±0.01

As can be seen from Table 4.3, by taking results of the baseline as reference, it is obvious that S-CLS gives the best results in the majority of data sets, and even outperform the baseline in certain cases. However, some degradation in performance are reported with "Scene" where S-CLS is somewhat weaker. Indeed, from all performance measure values listed in this table, only 6 of them that correspond to S-CLS (in the last column of the table), mostly in "Scene", show a degradation in comparison with other algorithms and the baseline.

Note that label correlation has been seriously discussed in [Dembczyński12]. Indeed, this notion only provides evidence for the existence of marginal dependence between labels, even though conditional dependence might be a cause of this dependence. In this context, it is known that in "Emotions" and "Yeast" data sets, there is a strong conditional dependence between labels. Thus, it is important to see that their values on 0/1 loss are low for S-CLS compared to the other methods, meaning that S-CLS performs better in data sets where there is a high correlation level between labels.

In the last scenario, we compare performance of various state-of-the-art multi-label classifiers trained on different data sets using: firstly all features, and secondly the best subset among the 30-top-ranked features selected by S-CLS that gives best performance. Table 4.4 reports performance of various classifiers in terms of AUC.

As shown in Table 4.4, with subsets of at most 30 features, all classifiers manage to perform as good as with the total feature space, and in some cases outperform. Some degradation in performance can be observed but, it is negligible, considering the gain in execution time and interpretability availed by using S-CLS.

Before wrapping up the experiments, we found it interesting to compare between different feature selection algorithms in terms of execution time. As can be seen from Table 4.5, S-CLS consumed by far less time than any other algorithm,

Table 4.5: Execution Time in Seconds

Data set	Feature Selection Algorithm			
	PPT+MI	ML-MI	Memetic	S-CLS
Birds	14.192	164.18	434.05	**04.572**
Emotions	12.627	08.678	135.62	**02.024**
Scene	404.09	108.19	701.25	**02.575**
Yeast	509.45	76.600	808.76	**01.337**
Slashdot	71.000	2221.8	5038.2	**21.430**
langlog	04.500	4319.9	3025.0	**03.338**
Plant	49.890	182.94	238.27	**19.287**
Human	440.50	663.30	1137.7	**06.081**

and this was on all data sets. On the other side, surprisingly PPT+MI runs faster than ML-MI on 5 data sets out of 8, where the latter execution time increases exponentially. The worst time is given by the wrapper algorithm, Memetic, which shows once again the superiority of filter selection methods over other approaches.

In the light of the results exposed in the different stages of our experiments, we can confidently conclude that S-CLS exhibits competitive performance against other high-performing supervised multi-label feature selection methods. S-CLS, with only a small part of labeled data, has the merit of treating multi-label feature selection without any deterioration in performance. Actually, experiments on different benchmark data sets from various domains have shown that the best performances are obtained when S-CLS is used as a preprocessing step for *ML-kNN*, meaning that the most relevant features are rather chosen by S-CLS. It is worthwhile mentioning that all algorithms we used do in fact take into account label correlation. Last but not least, we point out again that S-CLS performs with only 10% of instances as labeled data, whilst other algorithms use completely labeled data sets to achieve feature selection.

4.5 Conclusion

In this chapter, we proposed various filter-based solutions to conduct dimensionality reduction. More specifically, we achieved feature selection by considering simultaneously two important aspects of modern real-world applications: *multi-labeledness* and *small-labeled-samples*. For this setting, we introduced S-CLS as well as two transformation-based algorithms, all of which are based on Laplacian Score. Chiefly, with S-CLS we showed how the label information could be converted into *soft constraints* and used together with the geometric structure available from

the unlabeled data to be incorporated seamlessly in a new feature importance measure. For the sake of experiments, we introduced many available multi-label evaluation metrics, specifically designed to support assessing performance of multi-label algorithms, besides numerous multi-label data sets from various application domains were presented. Results were yielded by measuring both prediction and ranking accuracies. We showed that with only a small number of constraints, the proposed framework can efficiently outperform other state-of-the-art multi-label feature selection methods operating with fully labeled data.

"It is the harmony of the diverse parts, their symmetry, their happy balance; in a word it is all that introduces order, all that gives unity, that permits us to see clearly and to comprehend at once both the ensemble and the details."

Henri Poincare

5

Semi-supervised Multi-label Feature Selection: An Ensemble Approach

▷*In this chapter, we are particularly interested in ensemble methods. Mainly, to mitigate the effect of variance on S-CLS and take better advantage of label dependency, we propose an ensemble methodology by a 3-way approach which is based on a resampling of data (Bagging), a random subspace method (RSM) and an additional random sub-labeling strategy (RSL). The diversity created thanks to the proposed ensemble framework will help in boosting performance of the base algorithm S-CLS. Experiments were carried out on some benchmark data sets, and results were promising, showing that our method either outperform state-of-the-art algorithms or in the worst cases give comparable results.* ◁

Chapter outline

5.1 Introduction **86**
5.2 Ensemble Learning **87**
5.3 Ensemble S-CLS **88**
5.4 Experiments **90**
5.4.1 Experimental Settings 90
5.5 Results and Discussion **90**
5.6 Conclusion **94**

5.1 Introduction

Like any learning algorithm, feature selection struggles in dealing with high-dimensionality data. In fact, when the number of features grows significantly most of feature selection algorithms lose in accuracy. On the other hand, filter feature selection algorithms are peculiarly dependent on data and have great variance, which means that a slight change in data lead to skewed feature scores which will eventually bias the whole selection procedure. In this regard, ensemble methods are good artifacts that offer satisfying solutions to overfitting and variance. In fact, authors in [Bauer99], showed empirically that bagging reduces the variance of the underlying classifiers. Besides, it is claimed that overfitting can be avoided by projecting the high dimensional data on several lower dimensional spaces like with random subspace method.

Much research has proposed to use ensemble methods in multi-label learning. In [Tsoumakas07b], the random k-label sets (RAkEL) constructs an ensemble of classifiers, each of which is trained using a different small random subset of the set of labels. RAkEL has the merit to deal efficiently with label correlation. [Tsoumakas08] proposed a method that uses a hierarchy of multi-label classifiers (HOMER), where each classifier operates on a subset of labels. [Read11] introduces an ensemble method, called ensemble of classifier chains (ECC), which uses a chain of binary classifiers each of which is trained upon the prediction of previous ones. In multi-label text categorization, a robust ensemble learning method called BoosTexter uses Adaboost.MH and Adaboost.MR algorithms [Schapire00] by applying Adaboost [Freund97] on weak classifiers. Surprisingly, little focus has been given to the application of ensemble methods to feature selection despite of their potential.

In this chapter, we propose an ensemble feature selection method by combining three random sampling techniques, each of which is iteratively applied on a different level of a data set: instance, dimensional and label space levels. More specifically, a Bagging is used to draw bootstrap samples from the original data, whilst Random Subspace Method (RSM) samples through the feature space, and a random sampling without replacement is performed in the label space. The latter, we will call RSL subsequently, is similar to RSM but performs on labels. The purpose is, in fact three-fold, while bagging and RSM help reduce variance and

increase learning accuracy, RSL addresses, in particular, label correlation which is a major concern when dealing with multi-label data.

5.2 Ensemble Learning

Ensemble Methods are learning algorithms that combine multiple learners to classify new examples. It has been experimentally established that ensembles are mostly much more accurate than their individual constituent, provided that the individual members are as diverse as possible. Ultimately, the real challenge will be to find appropriate techniques for creating diversity inside ensembles.

Accordingly, the combining schemes developed so far are either based on manipulating data, or manipulating learning algorithms. The manipulation of data gives rise to well-known methods like: boosting [Freund96], bagging [Breiman96], Random Subspace Method [Ho98], those methods commonly employ the same base model on generated replicates of the original data. On the other hand, with the second approach, methods like stacking and error-correcting output codes combine models of different types, which are then applied on a single data set[Wolpert92, Dietterich95]. This research proposes an ensemble method by data manipulation, more specifically by using bagging and Random Subspace Method in a unified framework. Bagging(bootstrap aggregating) is a subsampling technique intended to reduce the variance in unstable classifiers. The idea behind is very simple, in fact bagging draws bootstrap samples (random sampling with replacement) from the original training data, then applies the base classifier to each replicate of the data, finally the prediction decision is obtained either by a majority vote or simple averaging.

Random Subspace Method aims at alleviating overfitting in classification problems where the dimensionality is way too larger than the number of instances. To this end, RSM trains the base classifier on randomly chosen subspaces (without replacement) from the original feature space. Similar to bagging, the decision boundaries are computed by a majority vote or averaging.

In addition to bagging and RSM, we present another combining strategy that randomly samples without replacement in the label space (RSL). Hence, the resulting data replicates are modified along the sample, the feature and the label

spaces. In so doing, we do not only enrich diversity, but it will allow us to deal efficiently with label correlation as well.

5.3 Ensemble S-CLS

The success of ensemble methods is highly dependent on the degree of diversity among individual members. This claim is supported by various research studies, which show that performance is definitely commensurate with diversity within the ensemble.

In this regard, we propose to apply simultaneously three ensemble techniques on different aspects of a data set. First, RSM performs a subspace selection to feature space, whereas RSL applies a random subset selection to label space. Afterwards, bagging applies a random sub-sampling with replacement to instance space. In the end, each data replicate will be a subsample obtained by bagging, projected on a feature subset selected by RSM, and associated with a label subspace chosen by RSL.

By such a way, we ensure a large scale of diversity among components, help alleviate the curse of dimensionality and treat label correlation, all in one shot. More specifically, the diversity is definitely ensured by the application of the three ensemble methods. On the other hand, the projection from the high dimensional feature space into the low dimensional space can avoid the problems caused by high dimensionality, while the selection of smaller subset of labels for each subsample permits to deal efficiently with label correlation. Algorithm 10 outlines the proposed approach we call 3-3FS, which stands for 3-way approach (bagging,+RSM+RSL) dealing with 3 paradigms (feature selection + semi-supervised learning + multi-label context).

Algorithm 10 3-3FS

Input:
Set of labeled instances (X_L); set of unlabeled instances (X_U); input space ($F = \{F_1, \ldots, F_m\}$); label space ($Y = \{Y_1, \ldots, Y_c\}$); ensemble size (N)
Initialize the scores $\mathbf{I}(f_r)$ to zero for each feature F_r
Initialize the occurrences $\mathbf{O}(f_r)$ to zero for each feature F_r
for $i = 1 : N$ **do**
  RSM^i = randomly draw M features from F without replacement
  RSL^i = randomly draw L labels from Y without replacement
  ${X_{L,b}}^i$ = bootstrap sample from X_L projected onto RSM^i and assigned to labels in RSL^i
  ${X_U}^i$ = the unlabeled sample X_U projected onto RSM^i
  tmp^i = $S\text{-}CLS({X_{L,b}}^i, {X_U}^i)$ compute the soft-constrained Laplacian score of each feature in RSM^i using Algorithm 9
  for each feature $F_r \in RSM^i$ **do**
    $\mathbf{I}(f_r) = \mathbf{I}(f_r) + tmp^i(f_r)$
    $\mathbf{O}(f_r) = \mathbf{O}(f_r) + 1$
  end for
end for
for each feature $F_r \in F$ **do**
  $\mathbf{I}(f_r) = \frac{\mathbf{I}(f_r)}{\mathbf{O}(f_r)}$
end for
rank features in F according to their scores $\mathbf{I}$ in ascending order.
return F

In each iteration, we randomly select from F, the original input space, M features to build the new feature space denoted by RSM^i. Meanwhile, we randomly select L labels from the label space Y, RSL^i will be the new label space for this iteration. Note that, the former selections are done without replacement. Afterwards, bootstrap sampling is performed in the supervised part of the data set X_L, the new sample X_L^i is then projected on the corresponding feature subspace RSM^i and assigned to RSL^i as well. On the other hand, the unsupervised part X_U of the original data set, is also projected on the same feature subspace RSM^i, the result of this projection is denoted by X_U^i. By the end of the iteration, the resulting parts X_L^i and X_U^i are submitted to S-CLS, which will compute the scores of each feature in the related feature subspace RSM^i. Finally, according to their occurrences in the different subspace through the iterations, each feature will be assigned a final score corresponding to the averaging of its scores in each iteration.

5.4 Experiments

This section describes the general layout of the experiments. We will reuse the same experimental protocol outlined in Chapter 3.5, in particular we shall run experiments on all the data sets by using *ML-kNN* as the base line classifier. Besides, tuning parameters for the various ensemble learning approaches will be discussed and set.

5.4.1 Experimental Settings

Data sets used in this work are originally supervised. However to be usable in semi-supervised context, we operate a random selection with a rate of 10% of supervision, that is to say that 90% of data is selected as unlabeled, and the rest is taken as multi-labeled. S-CLS and 3-3FS perform on semi-supervised data, while other methods are totally supervised.

In the experiments, parameters of our ensemble method are set as follows: The ensemble size is determined by equation 5.1

$$N = 10ceil\left(\frac{\log(0.01)}{\log(1-1/\sqrt{m})}\right). \tag{5.1}$$

The number of features in each member is $M = \sqrt{m}$, where m is the size of the feature space, and the size L of each label subset is $L = \sqrt{C}$, where C is the cardinality of the original label space.

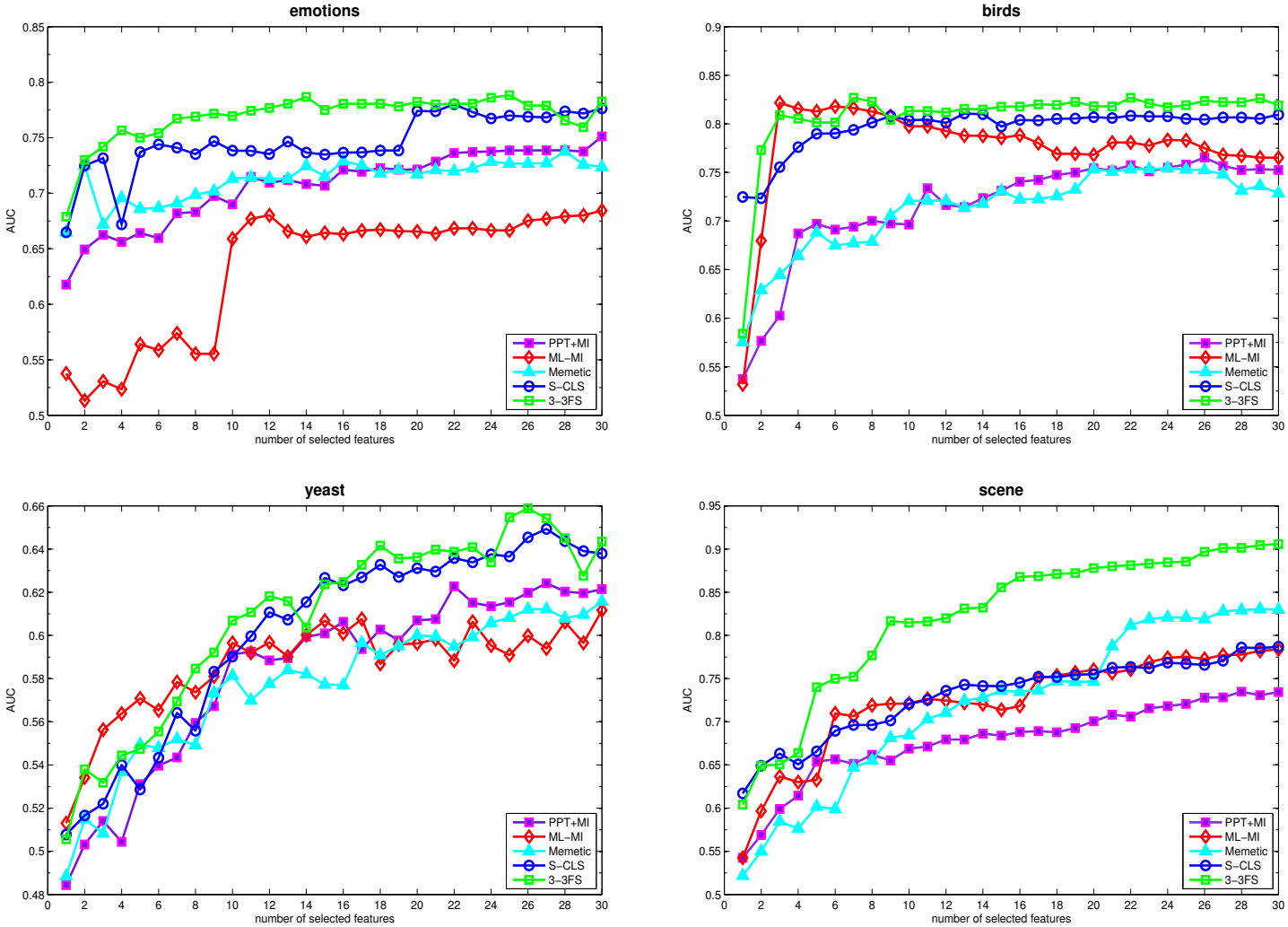
As we did in the previous chapter, classification performance of the different feature selection methods are evaluated as follows: After a feature selection step, selected features are submitted to *ML-kNN*, and the performance of *ML-kNN* in terms of certain evaluation measures are recorded. *ML-kNN*classifier is tuned via 3-fold cross-validation.

5.5 Results and Discussion

Results reported in this section interpret the impact of feature selection on the performance of the base classifier *ML-kNN*. First, 3-3FS along with other feature selection algorithms are used to pick out most relevant features in relation to each

data set. Subsequently, using the selected features, *ML-kNN* is tested by means of different evaluation measures and results are reported in Figure 5.0 and Table 5.1.

Figure 5.0 shows the classification performance of *ML-kNN* using increasingly the 30 top-ranked features according to different feature selection methods. Performance is illustrated in terms of Area Under ROC (AUC). It is obviously clear that our ensemble method outperforms other algorithms in all data sets, and S-CLS comes second. 3-3FS outstandingly do well in the "scene" data set, recall that "scene" have a strong correlation between labels. Meaning that 3-3FS, thanks to the use of RSL, succeed to capture with efficacy the correlation between labels. On the other data sets, 3-3FS achieves also good results, which shows that features selected by our method are indeed relevant and helped in improving the ranking performance of *ML-kNN*.



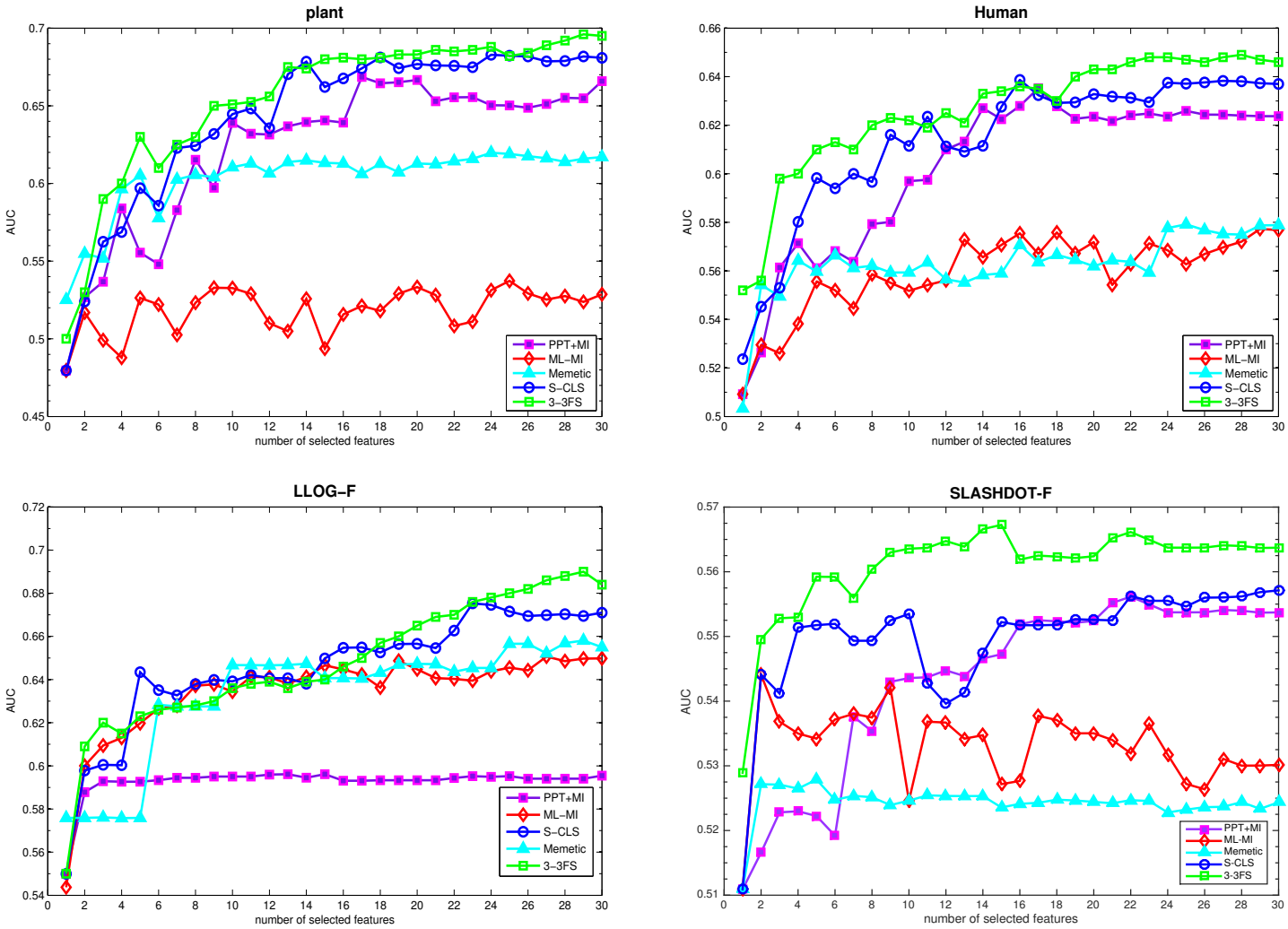


Figure 5.0: AUC v.s. number of selected features

In Table 5.1, we report the performance of the baseline classifier *ML-kNN* using the entire feature space, that is without selection. The baseline will serve as a ground truth to show the gain/loss in performance when applying feature selection. Table 4.3 summarizes performance in terms of the most used and powerful evaluation measures. Values are obtained by using exactly 30 top-ranked features. For all evaluation measures, 3-3FS has the best values and shows the best performance in all data sets. More specifically, in terms of classification performance, represented by hamming loss and micro-F we see a neat improvement of S-CLS when using the proposed ensemble technique. On the other hand, ranking performance quantified by average precision and ranking loss, shows also a good gain when employing 3-3FS as a combining strategy to S-CLS.

Table 5.1: Results (mean±std.) on all data sets used, over all measures ("↘ indicates the smaller the better"; "↗ indicates the larger the better").

Data sets	Feature Selection Algorithm					
	Baseline	PPT+MI	ML-MI	Memetic	S-CLS	3-3FS
	Hamming Loss↘					
Birds	0.05±0.006	0.05±0.005	0.05±0.006	0.05±0.004	0.05±0.005	0.04±0.002
Emotions	0.27±0.018	0.27±0.013	0.28±0.021	0.27±0.010	0.25±0.019	0.23±0.010
Scene	0.11±0.008	0.19±0.017	0.18±0.023	0.12±0.016	0.19±0.009	0.11±0.002
Yeast	0.20±0.003	0.21±0.001	0.21±0.002	0.21±0.005	0.20±0.004	0.20±0.001
Slashdot	0.02±0.002	0.02±0.001	0.02±0.001	0.02±0.001	0.02±0.002	0.01±0.001
Langlog	0.16±0.009	0.19±0.009	0.17±0.009	0.17±0.011	0.17±0.009	0.16±0.005
Plant	0.09±0.002	0.09±0.003	0.09±0.002	0.09±0.002	0.09±0.002	0.08±0.001
Human	0.08±0.000	0.08±0.001	0.08±0.000	0.08±0.000	0.08±0.001	0.07±0.004
	0/1 Loss↘					
Birds	0.54±0.039	0.53±0.032	0.53±0.033	0.54±0.036	0.51±0.039	0.51±0.001
Emotions	0.85±0.007	0.75±0.072	0.87±0.022	0.80±0.039	0.72±0.055	0.69±0.002
Scene	0.38±0.044	0.95±0.035	0.83±0.054	0.50±0.079	0.49±0.090	0.37±0.004
Yeast	0.78±0.041	0.83±0.023	0.85±0.025	0.84±0.020	0.78±0.011	0.76±0.027
Slashdot	0.29±0.025	0.26±0.018	0.24±0.014	0.23±0.007	0.25±0.012	0.22±0.034
Langlog	0.84±0.022	0.83±0.022	0.83±0.020	0.83±0.018	0.84±0.023	0.81±0.016
Plant	0.94±0.011	0.91±0.043	0.99±0.007	0.97±0.012	0.91±0.021	0.90±0.007
Human	0.92±0.022	0.90±0.009	0.94±0.017	0.98±0.017	0.93±0.016	0.90±0.009
	Macro-F↗					
Birds	0.04±0.011	0.11±0.052	0.11±0.048	0.08±0.011	0.23±0.012	0.25±0.002
Emotions	0.43±0.020	0.51±0.032	0.42±0.015	0.47±0.007	0.54±0.042	0.58±0.001
Scene	0.62±0.031	0.09±0.063	0.25±0.079	0.59±0.063	0.62±0.111	0.63±0.029
Yeast	0.63±0.016	0.60±0.003	0.58±0.014	0.59±0.009	0.63±0.008	0.66±0.011
Slashdot	0.70±0.017	0.76±0.009	0.76±0.008	0.75±0.003	0.75±0.009	0.77±0.005
Langlog	0.53±0.005	0.41±0.039	0.47±0.003	0.49±0.010	0.53±0.019	0.54±0.001
Plant	0.11±0.029	0.15±0.057	0.01±0.016	0.06±0.024	0.15±0.033	0.17±0.014
Human	0.14±0.030	0.18±0.011	0.12±0.033	0.05±0.035	0.14±0.017	0.19±0.003
	Micro-F↗					
Birds	0.02±0.003	0.05±0.023	0.04±0.026	0.03±0.003	0.07±0.013	0.08±0.022
Emotions	0.33±0.018	0.44±0.008	0.33±0.021	0.39±0.007	0.49±0.041	0.51±0.005
Scene	0.50±0.044	0.08±0.049	0.19±0.061	0.47±0.067	0.50±0.085	0.53±0.012
Yeast	0.35±0.023	0.29±0.008	0.28±0.015	0.29±0.010	0.35±0.011	0.38±0.001
Slashdot	0.04±0.001	0.04±0.001	0.05±0.008	0.04±0.006	0.04±0.000	0.06±0.032
Langlog	0.26±0.013	0.15±0.028	0.19±0.002	0.20±0.008	0.27±0.010	0.29±0.005
Plant	0.04±0.018	0.06±0.016	0.01±0.005	0.03±0.007	0.06±0.014	0.09±0.008
Human	0.06±0.008	0.06±0.006	0.03±0.005	0.02±0.009	0.06±0.012	0.07±0.011
	Average precision↗					
Birds	0.38±0.018	0.44±0.011	0.46±0.015	0.43±0.027	0.57±0.020	0.59±0.001
Emotions	0.69±0.010	0.72±0.020	0.69±0.021	0.71±0.044	0.73±0.041	0.74±0.012
Scene	0.82±0.017	0.54±0.004	0.60±0.029	0.79±0.025	0.78±0.014	0.81±0.004
Yeast	0.76±0.009	0.73±0.003	0.73±0.004	0.73±0.004	0.76±0.008	0.77±0.013
Slashdot	0.88±0.004	0.88±0.006	0.88±0.003	0.88±0.005	0.88±0.005	0.89±0.033
Langlog	0.61±0.011	0.63±0.018	0.62±0.011	0.62±0.005	0.62±0.011	0.64±0.001
Plant	0.56±0.009	0.56±0.019	0.49±0.020	0.54±0.010	0.56±0.026	0.59±0.003
Human	0.56±0.007	0.59±0.006	0.54±0.012	0.55±0.008	0.58±0.006	0.60±0.001
	One Error↘					
Birds	0.73±0.036	0.68±0.034	0.66±0.035	0.68±0.015	0.51±0.014	0.50±0.012

Continued on next page

Table 5.1 – *Continued from previous page*

Data sets	Feature Selection Algorithm					
	Baseline	PPT+MI	ML-MI	Memetic	S-CLS	3-3FS
Emotions	0.42±0.013	0.38±0.044	0.41±0.033	0.41±0.055	0.37±0.038	0.35±0.011
Scene	0.30±0.030	0.69±0.021	0.61±0.056	0.34±0.047	0.35±0.031	0.28±0.008
Yeast	0.23±0.008	0.25±0.001	0.25±0.004	0.26±0.006	0.23±0.013	0.21±0.004
Slashdot	0.09±0.004	0.09±0.004	0.10±0.003	0.09±0.005	0.09±0.005	0.07±0.003
Langlog	0.19±0.020	0.19±0.054	0.18±0.012	0.17±0.017	0.16±0.022	0.15±0.016
Plant	0.64±0.018	0.62±0.023	0.71±0.031	0.67±0.008	0.62±0.032	0.59±0.053
Human	0.61±0.009	0.59±0.012	0.64±0.014	0.64±0.003	0.60±0.015	0.57±0.061
	Coverage↘					
Birds	3.80±0.365	3.19±0.391	2.92±0.300	3.13±0.214	2.44±0.205	2.14±0.018
Emotions	2.37±0.076	2.16±0.064	2.34±0.145	2.25±0.082	2.07±0.117	1.99±0.013
Scene	0.62±0.053	1.82±0.041	1.50±0.047	0.76±0.041	0.78±0.042	0.60±0.020
Yeast	6.35±0.113	6.66±0.053	6.66±0.082	6.63±0.094	6.35±0.102	6.24±0.021
Slashdot	1.02±0.021	1.02±0.036	0.99±0.026	0.98±0.035	1.02±0.034	0.95±0.003
Langlog	48.75±1.82	49.23±1.71	49.67±1.80	49.68±2.29	48.67±2.24	46.41±0.013
Plant	2.36±0.058	2.27±0.114	2.75±0.121	2.45±0.078	2.29±0.150	2.19±0.043
Human	2.41±0.047	2.34±0.045	2.63±0.071	2.60±0.100	2.40±0.022	2.31±0.001
	Ranking Loss↘					
Birds	0.31±0.001	0.26±0.004	0.24±0.009	0.25±0.011	0.19±0.010	0.17±0.001
Emotions	0.28±0.010	0.24±0.014	0.27±0.017	0.25±0.037	0.22±0.038	0.21±0.002
Scene	0.11±0.010	0.34±0.004	0.28±0.012	0.13±0.013	0.13±0.007	0.11±0.015
Yeast	0.17±0.006	0.19±0.002	0.19±0.004	0.19±0.004	0.17±0.005	0.16±0.002
Slashdot	0.05±0.002	0.05±0.004	0.05±0.002	0.05±0.004	0.05±0.003	0.03±0.005
Langlog	0.20±0.009	0.19±0.010	0.20±0.008	0.20±0.010	0.20±0.010	0.20±0.006
Plant	0.20±0.007	0.19±0.012	0.24±0.012	0.21±0.009	0.19±0.017	0.18±0.005
Human	0.16±0.003	0.16±0.003	0.18±0.005	0.18±0.007	0.16±0.001	0.15±0.004

The usage of S-CLS on diverse subsamples obtained by applying a 3-pass combining technique has led to very interesting results when compared to the basic S-CLS and other state-of-the-art algorithms. Features selected by our ensemble method helped gain in both classification and ranking accuracy of the chosen base classifier (*ML-kNN*).

5.6 Conclusion

This chapter is concerned with ensemble methods for feature selection. In particular, to combat the curse of dimensionality we have designed and implemented an ensemble framework using S-CLS as the base algorithm. To this end, we incorporated three ensemble methods simultaneously, each of which addresses a different

aspect of the data. More specifically, we used the Bagging for the samples, the Random Subspace Method for the input space and Random Subspace Labeling for the label space. On another side, issues related to label correlation and label importance, were addressed. We showed how to compute label importance from the training data set and integrate those weights into a variant of S-CLS to measure feature importance. To put the framework to the test, we conducted experiments on several real-world data sets from various application domains, the results gave credit and validated the merit of the proposed framework in comparison with different state-of-the-art algorithms both in classification and ranking.

"Logic founded on passions reverses the traditional sequence of reasoning and places the conclusions before the premises"

Albert Camus, The Rebel

6

Conclusion and Future Directions

Feature selection, semi-supervised learning and multi-label classification are different challenges for machine learning and data mining communities. While other works have addressed each of these problems separately, in this research we showed how they can be addressed together. Essentially, we investigated different ways of tackling the problem of semi-supervised mutli-label feature selection. In so doing, we embraced spectral graph theory as our groundwork to achieve dimensionality reduction. All of the proposed algorithms make use of few amount of multi-labeled data together with a large volume of unlabeled ones to select the most informative and discriminative features. In this setting, we proposed several filter feature selection algorithms each of which involves a disparate approach, including data transformation and adaptation of the Laplacian Score. In particular, we used two well-known algorithms, Binary Relevance and Label Powerset, to transform the multi-label data into single-label data upon which we applied a traditional single label feature selection algorithm. On the other hand, we coined an adaptation-based algorithm to handle multi-label feature selection directly without the need for the transformation pre-processing step. Lastly, we devised an ensemble method

to help reduce the variance in the latter method and make more effective use of label dependency.

To sum up, we started by reviewing the literature of dimensionality reduction and multi-label classification. First, we gave an overview on some of key ideas in semi-supervised learning. In particular, we discussed the "small-labeled-sample" problem, in which the volume of the unlabeled data is much larger than the amount of the labeled ones. We explained how semi-supervision is used to fill the gap between supervised and unsupervised learning and overcome their major drawbacks. Which on the one hand, is that supervised algorithms require a large amount of labeled training data to give good results. And on the other hand, unsupervised algorithms ignore the label information, which may lead to performance degradation. In this context, we introduced the notions of partial background information usually expressed in terms of pairwise constraints, which is used to specify certain similarity/dissimilarity between instances and tell if two instances are from the same class (must-link constraint) or from disparate classes (cannot-link constraint). We explained how those constraints could be used to guide the learning process and yield good results. We also showed how to tap into the powerful and sound theory of spectral graph analysis to solve semi-supervised dimensionality reduction. What's more, we illustrated the semi-supervised dimensionality reduction by some representative methods while giving emphasis to methods related to the graph-theoritic framework. In addition, we took a deep delve into the field of multi-label learning, starting by a formal definition and passing by the most prominent applications and real-world problems, to wrap up by investigating numerous ways to perform multi-label prediction and to solve dimensionality reduction in multi-label data. This literature tour gave us precious and useful understandings and allowed us to gain important insights on how to deal with semi-supervision in multi-label data.

In order to tackle the problem of dimensionality reduction, we proposed several approaches. More precisely, we embraced a filter paradigm to resolve semi-supervised multi-label feature selection. First, we used a transformational approach, which combines two powerful multi-label tools, namely Binary Relevance and Label Powerset, which transform the multi-label data into single-label data, and combine them with a traditional feature importance measure. In particular, to measure the goodness of the features we used CLS, This gives rise to two feature

selection methods: BR-CLS and LP-CLS. More importantly, in a further step we introduced S-CLS, a new multi-label feature selection method which extends CLS to handle multi-label data directly. We demonstrated how the background knowledge offered by the labeled part of the data could be transformed into constraints and integrated into the feature selection process, along with the geometrical structure provided by the unlabeled part of data. Specifically, we used the multi-label information from the labeled part of the data to build the *soft-constaints* and integrated them in the objective function. Experimental results on various benchmark data sets were performed by measuring both prediction and ranking accuracy. The extensive empirical study validated our proposal and demonstrated that with only a small number of *soft-constraints*, the proposed algorithm outperformed other multi-label feature selection methods and the baseline algorithm without feature selection.

Finally, to combat the variance in the previous algorithms and take more advantage of label correlation we introduced an ensemble framework, in which we combined three ensemble algorithms, each of which is applied on a different level of the data. To be specific, we used bagging for samples, Random Subspace Method (RSM) for the input space and a random subspace labeling for the label space, also we experimented various scenarios including a sub-sampling with and without replacement. Besides, issues related to label dependency and label importance were addressed. In addition, we presented how to compute the weighs of each label from the data sets and showed how they could be incorporated in the final feature importance measure. To evaluate the proposed framework we conducted experiments on numerous high-dimensional domains. Results were reported for various types of data sets from disparate domains, and showed the merit of the proposed ensemble framework in comparison with different state-of-the-art algorithms.

Last but not least, there are some avenues that could be investigated and planned for future research. First, it would be interesting to broaden the experiments to put our algorithms to the test on a large panel of data sets spanning various application domains. Besides, we plan to give more elaborate ideas to capture two key concepts in multi-label classification, namely label importance and label dependency. Furthermore, a special interest should be drawn to the scaling-up of the proposed methods. In this regard, we intend to go beyond the graph representation of the data so as to deal more effectively and efficiently with large-

scale data sets. Indeed, more efforts are needed to tailor the methods to make sense of the huge volume of high-dimensional data (involving millions of instances, features and labels), which is continuously being generated on a massive scale. To this end, it would be useful to consider a big data solution in a distributed computing environment. Finally, it is worthwhile to fine-tune S-CLS in order to accommodate another emerging issue, which is multi-output regression problems.

Bibliography

[Agarwal06] Sameer Agarwal, Kristin Branson & Serge Belongie. *Higher order learning with graphs*. In Proceedings of the 23rd international conference on Machine learning, pages 17–24. ACM, 2006. 57

[Aho12] Timo Aho, Bernard Ženko, Sašo Džeroski & Tapio Elomaa. *Multi-target regression with rule ensembles*. Journal of Machine Learning Research, vol. 13, no. Aug, pages 2367–2407, 2012. 49

[Alalga16] Abdelouahid Alalga, Khalid Benabdeslem & Nora Taleb. *Soft-constrained Laplacian score for semi-supervised multi-label feature selection*. Knowledge and Information Systems, vol. 47, no. 1, pages 75–98, Springer, 2016. 6, 63

[Alter00] Orly Alter, Patrick O Brown & David Botstein. *Singular value decomposition for genome-wide expression data processing and modeling*. Proceedings of the National Academy of Sciences, vol. 97, no. 18, pages 10101–10106, National Acad Sciences, 2000. 2

[Andrews03] Stuart Andrews, Ioannis Tsochantaridis & Thomas Hofmann. *Support vector machines for multiple-instance learning*. Advances in neural information processing systems, pages 577–584, MIT; 1998, 2003. 48

[Bauer99] Eric Bauer & Ron Kohavi. *An empirical comparison of voting classification algorithms: Bagging, boosting, and variants*.

Machine learning, vol. 36, no. 1-2, pages 105–139, Springer, 1999. 86

[Benabdeslem11a] K. Benabdeslem & M. Hindawi. *Constrained Laplacian score for semi-supervised feature selection.* In Proceedings of ECML-PKDD conference, pages 204–218, 2011. 15

[Benabdeslem11b] Khalid Benabdeslem & Mohammed Hindawi. *Constrained Laplacian score for semi-supervised feature selection.* In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 204–218. Springer, 2011. 7, 22, 62, 64, 66, 67

[Benabdeslem14] Khalid Benabdeslem & Mohammed Hindawi. *Efficient semi-supervised feature selection: constraint, relevance, and redundancy.* IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 5, pages 1131–1143, IEEE, 2014. 26

[Benabdeslem16] Khalid Benabdeslem, Haytham Elghazel & Mohammed Hindawi. *Ensemble constrained Laplacian score for efficient and robust semi-supervised feature selection.* Knowledge and Information Systems, vol. 49, no. 3, pages 1161–1185, Springer, 2016. 26

[Bilenko04] Mikhail Bilenko, Sugato Basu & Raymond J Mooney. *Integrating constraints and metric learning in semi-supervised clustering.* In Proceedings of the twenty-first international conference on Machine learning, page 11. ACM, 2004. 14

[Boutell04] Matthew R Boutell, Jiebo Luo, Xipeng Shen & Christopher M Brown. *Learning multi-label scene classification.* Pattern recognition, vol. 37, no. 9, pages 1757–1771, Elsevier, 2004. 5, 35, 37, 39, 69

[Breiman96] Leo Breiman. *Bagging predictors.* Machine learning, vol. 24, no. 2, pages 123–140, Springer, 1996. 87

[Briggs12] Forrest Briggs, Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Z Fern, Raviv Raich, Sarah JK Hadley, Adam S Hadley &

Matthew G Betts. *Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach.* The Journal of the Acoustical Society of America, vol. 131, no. 6, pages 4640–4650, Acoustical Society of America, 2012. 69

[Caruana98] Rich Caruana. *Multitask learning.* In Learning to learn, pages 95–133. Springer, 1998. 49

[Cesa-Bianchi06] Nicolò Cesa-Bianchi, Claudio Gentile & Luca Zaniboni. *Incremental algorithms for hierarchical classification.* Journal of Machine Learning Research, vol. 7, no. Jan, pages 31–54, 2006. 46

[Chapelle09] Olivier Chapelle, Bernhard Scholkopf & Alexander Zien. *Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews].* IEEE Transactions on Neural Networks, vol. 20, no. 3, pages 542–542, IEEE, 2009. 11

[Charte15] Francisco Charte & David Charte. *Working with multilabel datasets in R: the mldr package.* R J, vol. 7, no. 2, pages 149–162, 2015. 58

[Chen07] Weizhu Chen, Jun Yan, Benyu Zhang, Zheng Chen & Qiang Yang. *Document transformation for multi-label feature selection in text categorization.* In Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, pages 451–456. IEEE, 2007. 39, 51

[Cheng09] Weiwei Cheng & Eyke Hüllermeier. *Combining instance-based learning and logistic regression for multilabel classification.* Machine Learning, vol. 76, no. 2-3, pages 211–225, Springer, 2009. 72

[Chung97] Fan RK Chung. Spectral graph theory, volume 92. American Mathematical Soc., 1997. 15

[Clare01] Amanda Clare & Ross D King. *Knowledge discovery in multi-label phenotype data.* In European Conference on Principles of

Data Mining and Knowledge Discovery, pages 42–53. Springer, 2001. 43, 54

[Dash97] Manoranjan Dash & Huan Liu. *Feature selection for classification.* Intelligent data analysis, vol. 1, no. 3, pages 131–156, IOS Press, 1997. 4

[Davidson06] Ian Davidson, Kiri L Wagstaff & Sugato Basu. *Measuring constraint-set utility for partitional clustering algorithms.* In European Conference on Principles of Data Mining and Knowledge Discovery, pages 115–126. Springer, 2006. 24

[Deerwester90] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer & Richard Harshman. *Indexing by latent semantic analysis.* Journal of the American society for information science, vol. 41, no. 6, page 391, American Documentation Institute, 1990. 56

[Dembczyński12] Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng & Eyke Hüllermeier. *On label dependence and loss minimization in multi-label classification.* Machine Learning, vol. 88, no. 1-2, pages 5–45, Springer US, 2012. 45, 81

[Dembszynski10] Krzysztof Dembszynski, Willem Waegeman, Weiwei Cheng & Eyke Hüllermeier. *On label dependence in multilabel classification.* In LastCFP: ICML Workshop on Learning from Multi-label data. Ghent University, KERMIT, Department of Applied Mathematics, Biometrics and Process Control, 2010. 46

[Demšar06] Janez Demšar. *Statistical comparisons of classifiers over multiple data sets.* Journal of Machine learning research, vol. 7, no. Jan, pages 1–30, 2006. 77

[Dietterich95] Thomas G. Dietterich & Ghulum Bakiri. *Solving multiclass learning problems via error-correcting output codes.* arXiv preprint cs/9501101, 1995. 87

[Diplaris05] Sotiris Diplaris, Grigorios Tsoumakas, Pericles A Mitkas & Ioannis Vlahavas. *Protein classification with multiple algo-*

rithms. In Panhellenic Conference on Informatics, pages 448–456. Springer, 2005. 2, 35

[Doquire11] Gauthier Doquire & Michel Verleysen. *Feature selection for multi-label classification problems.* In Advances in Computational Intelligence, pages 9–16. Springer, 2011. 52

[Doquire13a] Gauthier Doquire & Michel Verleysen. *A graph Laplacian based approach to semi-supervised feature selection for regression problems.* Neurocomputing, vol. 121, pages 5–13, Elsevier, 2013. 27, 28

[Doquire13b] Gauthier Doquire & Michel Verleysen. *Mutual information-based feature selection for multilabel classification.* Neurocomputing, vol. 122, pages 148–155, 2013. 69

[Duda12] Richard O Duda, Peter E Hart & David G Stork. Pattern classification. John Wiley & Sons, 2012. 1

[Dy04] Jennifer G Dy & Carla E Brodley. *Feature selection for unsupervised learning.* The Journal of Machine Learning Research, vol. 5, pages 845–889, JMLR. org, 2004. 4

[Elisseeff01] André Elisseeff & Jason Weston. *A kernel method for multi-labelled classification.* In NIPS, volume 14, pages 681–687, 2001. 45, 69

[Fisher36] Ronald A Fisher. *The use of multiple measurements in taxonomic problems.* Annals of eugenics, vol. 7, no. 2, pages 179–188, Wiley Online Library, 1936. 2, 55

[Forman04] George Forman. *A pitfall and solution in multi-class feature selection for text classification.* In Proceedings of the twenty-first international conference on Machine learning, page 38. ACM, 2004. 51

[Freund95] Yoav Freund & Robert E Schapire. *A desicion-theoretic generalization of on-line learning and an application to boosting.* In European conference on computational learning theory, pages 23–37. Springer, 1995. 44

[Freund96] Yoav Freund, Robert E Schapire*et al. Experiments with a new boosting algorithm.* In ICML, volume 96, pages 148–156, 1996. 87

[Freund97] Yoav Freund & Robert E Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of computer and system sciences, vol. 55, no. 1, pages 119–139, Elsevier, 1997. 86

[Fukunaga13] Keinosuke Fukunaga. Introduction to statistical pattern recognition. Academic press, 2013. 2

[Fürnkranz08] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía & Klaus Brinker. *Multilabel classification via calibrated label ranking.* Machine learning, vol. 73, no. 2, pages 133–153, Springer, 2008. 42

[García10] Salvador García, Alberto Fernández, Julián Luengo & Francisco Herrera. *Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power.* Information Sciences, vol. 180, no. 10, pages 2044–2064, Elsevier, 2010. 77

[García15] Salvador García, Julián Luengo & Francisco Herrera. Data preprocessing in data mining. Springer, 2015. 2, 3

[Gärtner02] Thomas Gärtner, Peter A Flach, Adam Kowalczyk & Alexander J Smola. *Multi-instance kernels.* In ICML, volume 2, pages 179–186, 2002. 48

[Gretton05] Arthur Gretton, Olivier Bousquet, Alex Smola & Bernhard Schölkopf. *Measuring statistical dependence with Hilbert-Schmidt norms.* In International conference on algorithmic learning theory, pages 63–77. Springer, 2005. 56

[Grzegorzewski06] Przemysław Grzegorzewski. *The coefficient of concordance for vague data.* Computational Statistics & Data Analysis, vol. 51, no. 1, pages 314–322, Elsevier, 2006. 19

[Gu11] Q. Gu, Z. Li & J. Han. *Generalized fisher score for feature selection.* In Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence UAI '11, pages 266–273, 2011. 2

[Guyon03] Isabelle Guyon & André Elisseeff. *An introduction to variable and feature selection.* Journal of machine learning research, vol. 3, no. Mar, pages 1157–1182, JMLR. org, 2003. 2, 4

[Hall00] Mark A Hall. *Correlation-based feature selection of discrete and numeric class machine learning.* University of Waikato, Department of Computer Science, 2000. 55

[He05a] Xiaofei He, Deng Cai & Partha Niyogi. *Laplacian score for feature selection.* In NIPS, volume 186, page 189, 2005. 18, 19

[He05b] Xiaofei He, Deng Cai & Partha Niyogi. *Laplacian score for feature selection.* In Advances in neural information processing systems, pages 507–514, 2005. 64, 66

[Hindawi11] Mohammed Hindawi, Kais Allab & Khalid Benabdeslem. *Constraint Selection-Based Semi-supervised Feature Selection.* In ICDM, pages 1080–1085. IEEE, 2011. 24

[Ho98] Tin Kam Ho. *The random subspace method for constructing decision forests.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 20, no. 8, pages 832–844, IEEE, 1998. 87

[Holmes94] Geoffrey Holmes, Andrew Donkin & Ian H Witten. *Weka: A machine learning workbench.* In Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on, pages 357–361. IEEE, 1994. 57

[Hsu09] Daniel J Hsu, Sham Kakade, John Langford & Tong Zhang. *Multi-label prediction via compressed sensing.* In NIPS, volume 22, pages 772–780, 2009. 47

[Hüllermeier08] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng & Klaus Brinker. *Label ranking by learning pairwise preferences.* Artificial Intelligence, vol. 172, no. 16, pages 1897–1916, Elsevier, 2008. 41

[Hvidsten01] Torgeir R Hvidsten, Henryk Jan Komorowski, Arne K Sandvik, Astrid Lægreid*et al. Predicting gene function from gene expressions and ontologies.* In Pacific Symposium on Biocomputing, volume 6, pages 299–310, 2001. 38

[Jin02] Rong Jin & Zoubin Ghahramani. *Learning with multiple labels.* In NIPS, volume 2, pages 897–904, 2002. 48

[Jolliffe86] IT Jolliffe. *Principal component analysis. 1986.* Spring-verlag, New York, 1986. 2, 29

[Jungjit12] Suwimol Jungjit, Alex A Freitas, M Michaelis & J Cinatl. *A multi-label correlation-based feature selection method for the classification of neuroblastoma microarray data.* In Advances in Data Mining: 12th Industrial Conference (ICDM 2012) Workshop Proceedings Â Workshop on Data Mining in Life Sciences (DMLS 2012)., pages 149–157. IBAI Publishing, 2012. 55

[Jungjit13] Suwimol Jungjit, M Michaelis, Alex A Freitas & Jindrich Cinatl. *Two extensions to multi-label correlation-based feature selection: A case study in bioinformatics.* In 2013 IEEE International Conference on Systems, Man, and Cybernetics, pages 1519–1524. IEEE, 2013. 55

[Kalakech11] Mariam Kalakech, Philippe Biela, Ludovic Macaire & Denis Hamad. *Constraint scores for semi-supervised feature selection: A comparative study.* Pattern Recognition Letters, vol. 32, no. 5, pages 656–665, Elsevier, 2011. 18

[Kocev09] Dragi Kocev, Sašo Džeroski, Matt D White, Graeme R Newell & Peter Griffioen. *Using single-and multi-target regression trees and ensembles to model a compound index of vegetation*

condition. Ecological Modelling, vol. 220, no. 8, pages 1159–1168, Elsevier, 2009. 49

[Kohonen01] T. Kohonen. Self organizing map. Springer Verlag, Berlin, 2001. 67

[Lastra11] Gerardo Lastra, Oscar Luaces, Jose R Quevedo & Antonio Bahamonde. *Graphical feature selection for multilabel classification tasks*. In International Symposium on Intelligent Data Analysis, pages 246–257. Springer, 2011. 54

[Lee13] Jaesung Lee & Dae-Won Kim. *Feature selection for multi-label classification using multivariate mutual information*. Pattern Recognition Letters, vol. 34, no. 3, pages 349–357, Elsevier, 2013. 53, 69

[Lee15] Jaesung Lee & Dae-Won Kim. *Memetic feature selection algorithm for multi-label classification*. Information Sciences, vol. 293, pages 80–96, Elsevier, 2015. 69

[Maron98] Oded Maron. *Learning from ambiguity*. Doctorat, Massachusetts Institute of Technology, 1998. 48

[Naisbitt84] John Naisbitt & J Cracknell. *Megatrends: Ten new directions transforming our lives*. Rapport technique, Warner Books New York, 1984. 1

[Page99] Lawrence Page, Sergey Brin, Rajeev Motwani & Terry Winograd. *The PageRank citation ranking: Bringing order to the web.* Rapport technique, Stanford InfoLab, 1999. 15

[Park08] Cheong Hee Park & Moonhwi Lee. *On applying linear discriminant analysis for multi-labeled problems*. Pattern Recognition Letters, vol. 29, no. 7, pages 878–887, Elsevier, 2008. 55

[Peng05] H. Peng, F. Long & C. Ding. *Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-relevance, and Min-Redundancy*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pages 1226–1238, 2005. 2

[Pereira15] Rafael B Pereira, Alexandre Plastino, Bianca Zadrozny & Luiz HC Merschmann. *Information Gain Feature Selection for Multi-Label Classification.* Journal of Information and Data Management, vol. 6, no. 1, page 48, 2015. 54

[Pupo13] Oscar Gabriel Reyes Pupo, Carlos Morell & Sebastián Ventura Soto. *ReliefF-ML: an extension of ReliefF algorithm to multi-label learning.* In Iberoamerican Congress on Pattern Recognition, pages 528–535. Springer, 2013. 54

[Qi07] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei & Hong-Jiang Zhang. *Correlative multi-label video annotation.* In Proceedings of the 15th international conference on Multimedia, pages 17–26. ACM, 2007. 35, 37

[Qian10] Buyue Qian & Ian Davidson. *Semi-Supervised Dimension Reduction for Multi-Label Classification.* In AAAI, 2010. 70

[Quinlan14] J Ross Quinlan. C4. 5: programs for machine learning. Elsevier, 2014. 43

[Read08] Jesse Read. *A pruned problem transformation method for multi-label classification.* In Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008), volume 143150, 2008. 41

[Read09] Jesse Read, Bernhard Pfahringer, Geoff Holmes & Eibe Frank. *Classifier chains for multi-label classification.* In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 254–269. Springer, 2009. 46

[Read10] Jesse Read. *Scalable multi-label classification.* Doctorat, University of Waikato, 2010. 41

[Read11] Jesse Read, Bernhard Pfahringer, Geoff Holmes & Eibe Frank. *Classifier chains for multi-label classification.* Machine learning, vol. 85, no. 3, page 333, Springer, 2011. 72, 86

[Read12] Jesse Read, Albert Bifet, Geoff Holmes & Bernhard Pfahringer. *Scalable and efficient multi-label classification for evolving data streams.* Machine Learning, vol. 88, no. 1-2, pages 243–272, Springer, 2012. 69

[Read15] Jesse Read & Peter Reutemann. *Meka multi-label dataset repository.* `http://meka.sourceforge.net/`, 2015. 58

[Reyes15] Oscar Reyes, Carlos Morell & Sebastián Ventura. *Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context.* Neurocomputing, vol. 161, pages 168–182, Elsevier, 2015. 52, 54

[Robnik-Šikonja03] Marko Robnik-Šikonja & Igor Kononenko. *Theoretical and empirical analysis of ReliefF and RReliefF.* Machine learning, vol. 53, no. 1-2, pages 23–69, Springer, 2003. 2

[Salton91] Gerard Salton. *Developments in automatic text retrieval.* Science, vol. 253, no. 5023, pages 974–980, American Association for the Advancement of Science, 1991. 73

[Schapire00] Robert E Schapire & Yoram Singer. *BoosTexter: A boosting-based system for text categorization.* Machine learning, vol. 39, no. 2-3, pages 135–168, Springer, 2000. 2, 5, 35, 36, 44, 72, 86

[Schölkopf02] Bernhard Schölkopf & Alexander J Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002. 45

[Spolaôr13a] Newton Spolaôr, Everton Alvares Cherman, Maria Carolina Monard & Huei Diana Lee. *A comparison of multi-label feature selection methods using the problem transformation approach.* Electronic Notes in Theoretical Computer Science, vol. 292, pages 135–151, Elsevier, 2013. 51, 52

[Spolaôr13b] Newton Spolaôr & Grigorios Tsoumakas. *Evaluating feature selection methods for multi-label text classification.* BioASQ workhsop, Citeseer, 2013. 51

[Spolaôr14a] Newton Spolaôr & Maria Carolina Monard. *Evaluating ReliefF-based multi-label feature selection algorithm.* In Ibero-American Conference on Artificial Intelligence, pages 194–205. Springer, 2014. 54

[Spolaôr14b] Newton Spolaôr, Maria Carolina Monard, Grigorios Tsoumakas & Huei Lee. *Label construction for multi-label feature selection.* In Intelligent Systems (BRACIS), 2014 Brazilian Conference on, pages 247–252. IEEE, 2014. 52

[Sun08] Liang Sun, Shuiwang Ji & Jieping Ye. *Hypergraph spectral learning for multi-label classification.* In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 668–676. ACM, 2008. 57

[Sun10] Dan Sun & Daoqiang Zhang. *Bagging constraint score for feature selection with pairwise constraints.* Pattern Recognition, vol. 43, no. 6, pages 2106–2118, Elsevier, 2010. 19

[Szymański17] Piotr Szymański & Tomasz Kajdanowicz. *A scikit-based Python environment for performing multi-label classification.* `http://scikit.ml/`, 2017. 58

[Tang07] Wei Tang & Shi Zhong. *Pairwise constraints-guided dimensionality reduction.* Computational Methods of Feature Selection, Chapman and Hall, CRC, pages 295–312, 2007. 14

[Trohidis08] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris & Ioannis P Vlahavas. *Multi-Label Classification of Music into Emotions.* In ISMIR, volume 8, pages 325–330, 2008. 37, 52, 69

[Tsoumakas07a] Grigorios Tsoumakas & Ioannis Katakis. *Multi-label classification: An overview.* International Journal of Data Warehousing and Mining (IJDWM), vol. 3, no. 3, pages 1–13, IGI Global, 2007. 5, 38, 43

[Tsoumakas07b] Grigorios Tsoumakas & Ioannis Vlahavas. *Random k-labelsets: An ensemble method for multilabel classification.* In European

Conference on Machine Learning, pages 406–417. Springer, 2007. 42, 47, 51, 72, 86

[Tsoumakas08] Grigorios Tsoumakas, Ioannis Katakis & Ioannis Vlahavas. *Effective and efficient multilabel classification in domains with large number of labels.* In Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), pages 30–44, 2008. 46, 72, 86

[Tsoumakas09a] Grigorios Tsoumakas, Ioannis Katakis & Ioannis Vlahavas. *Mining multi-label data.* In Data mining and knowledge discovery handbook, pages 667–685. Springer, 2009. 36, 39, 40, 51

[Tsoumakas09b] Grigorios Tsoumakas, Min-Ling Zhang & Zhi-Hua Zhou. *Tutorial on learning from multi-label data.* In ECML PKDD, Bled, Slovenia, 2009. 49

[Tsoumakas11] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek & Ioannis Vlahavas. *Mulan: A java library for multi-label learning.* Journal of Machine Learning Research, vol. 12, no. Jul, pages 2411–2414, 2011. 57

[Ueda03] Naonori Ueda & Kazumi Saito. *Parametric mixture models for multi-labeled text.* Advances in neural information processing systems, pages 737–744, MIT; 1998, 2003. 36

[Veloso07] Adriano Veloso, Wagner Meira Jr, Marcos Gonçalves & Mohammed Zaki. *Multi-label lazy associative classification.* In European Conference on Principles of Data Mining and Knowledge Discovery, pages 605–612. Springer, 2007. 36

[Wagstaff01] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl*et al. Constrained k-means clustering with background knowledge.* In ICML, volume 1, pages 577–584, 2001. 14

[Wang10] Hua Wang, Chris HQ Ding & Heng Huang. *Multi-Label Classification: Inconsistency and Class Balanced K-Nearest Neighbor.* In AAAI, 2010. 56

[Wang13] Xiao Wang & Guo-Zheng Li. *Multilabel learning via random label selection for protein subcellular multilocations prediction.* IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 10, no. 2, pages 436–446, IEEE Computer Society Press, 2013. 38

[Wang14] Jiarong Wang, Jun Feng, Xia Sun, Su-Shing Chen & Bo Chen. *Simplified Constraints Rank-SVM for Multi-label Classification.* In Chinese Conference on Pattern Recognition, pages 229–236. Springer, 2014. 45

[Weston02] Jason Weston, Olivier Chapelle, Andre Elisseeff, Bernhard Schölkopf & Vladimir Vapnik. *Kernel dependency estimation.* In NIPS, volume 3, page 4, 2002. 47

[Wolpert92] David H Wolpert. *Stacked generalization.* Neural networks, vol. 5, no. 2, pages 241–259, Elsevier, 1992. 87

[Xu13] Jianhua Xu. *Fast multi-label core vector machine.* Pattern Recognition, vol. 46, no. 3, pages 885–898, Elsevier, 2013. 45, 69

[Yu04] Lei Yu & Huan Liu. *Efficient feature selection via analysis of relevance and redundancy.* Journal of machine learning research, vol. 5, no. Oct, pages 1205–1224, 2004. 54

[Yu05] Kai Yu, Shipeng Yu & Volker Tresp. *Multi-label informed latent semantic indexing.* In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 258–265. ACM, 2005. 56

[Yu13] Guoxian Yu, Huzefa Rangwala, Carlotta Domeniconi, Guoji Zhang & Zhiwen Yu. *Protein function prediction using multi-label ensemble classification.* IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 10, no. 4, pages 1–1, IEEE Computer Society Press, 2013. 38

[Zhang05] Min-Ling Zhang & Zhi-Hua Zhou. *A k-nearest neighbor based algorithm for multi-label classification.* In Granular Computing, 2005 IEEE International Conference on, volume 2, pages 718–721. IEEE, 2005. 43

[Zhang06] Min-Ling Zhang & Zhi-Hua Zhou. *Multilabel neural networks with applications to functional genomics and text categorization.* IEEE transactions on Knowledge and Data Engineering, vol. 18, no. 10, pages 1338–1351, IEEE, 2006. 5, 38, 44

[Zhang07a] D. Zhang, Z.H. Zhou & S. Chen. *Semi-supervised dimensionality reduction.* In Proceedings of SIAM International Conference on Data Mining, 2007. 14, 29

[Zhang07b] Min-Ling Zhang & Zhi-Hua Zhou. *ML-KNN: A lazy learning approach to multi-label learning.* Pattern recognition, vol. 40, no. 7, pages 2038–2048, Elsevier, 2007. 5, 71

[Zhang08a] Daoqiang Zhang, Songcan Chen & Zhi-Hua Zhou. *Constraint Score: A new filter method for feature selection with pairwise constraints.* Pattern Recognition, vol. 41, no. 5, pages 1440–1451, Elsevier, 2008. 14, 64

[Zhang08b] Daoqiang Zhang, Songcan Chen & Zhi-Hua Zhou. *Constraint Score: A new filter method for feature selection with pairwise constraints.* Pattern Recognition, vol. 41, no. 5, pages 1440–1451, Elsevier, 2008. 18, 19

[Zhang10] Yin Zhang & Zhi-Hua Zhou. *Multilabel dimensionality reduction via dependence maximization.* ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 4, no. 3, page 14, ACM, 2010. 56

[Zhang14] Min-Ling Zhang & Zhi-Hua Zhou. *A review on multi-label learning algorithms.* IEEE transactions on knowledge and data engineering, vol. 26, no. 8, pages 1819–1837, IEEE, 2014. 34

[Zhao07] Zheng Zhao & Huan Liu. *Semi-supervised Feature Selection via Spectral Analysis.* In SDM. SIAM, 2007. 5, 19

[Zhao08] Jidong Zhao, Ke Lu & Xiaofei He. *Locality sensitive semi-supervised feature selection.* Neurocomputing, vol. 71, no. 10, pages 1842–1849, Elsevier, 2008. 21

[Zhao10] Zheng Zhao, Fred Morstatter, Shashvata Sharma, Salem Alelyani, Aneeth Anand & Huan Liu. *Advancing feature selection research.* ASU feature selection repository, pages 1–28, 2010. 51

[Zhou07] Zhi-Hua Zhou & Min-Ling Zhang. *Multi-instance multi-label learning with application to scene classification.* Advances in neural information processing systems, vol. 19, page 1609, MIT; 1998, 2007. 48

[Zhu05] Xiaojin Zhu. *Semi-supervised learning literature survey.* Citeseer, 2005. 11