

Universiteit
Leiden
The Netherlands

Exceptional Model Mining

Duivesteijn, W.

Citation
Duivesteijn, W. (2013, September 17). *Exceptional Model Mining*. Retrieved from https://hdl.handle.net/1887/21760

Version:	Corrected Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/21760

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle http://hdl.handle.net/1887/21760 holds various files of this Leiden University dissertation.

Author: Duivesteijn, Wouter
Title: Exceptional model mining
Issue Date: 2013-09-17


Exceptional
Model
Mining
Wouter Duivesteijn

Exceptional Model Mining

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 17 september 2013
klokke 11.15 uur

door

Wouter Duivesteijn

geboren te Rotterdam
in 1984

Promotiecommissie

Promotor:	prof. dr. J. N. Kok	
Co-promotor:	dr. A. J. Knobbe	
Overige leden:	prof. dr. P. A. Flach	(University of Bristol)
	prof. dr. H. Blockeel	(Katholieke Universiteit Leuven)
	dr. W. A. Kosters	

Cover photo: ochre sea stars (*Pisaster ochraceus*), taken at Ganges Harbour, Salt Spring Island, British Columbia, Canada. Licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license by D. Gordon E. Robertson.

This research is financially supported by the Netherlands Organisation for Scientific Research (NWO) under project number 612.065.822 (Exceptional Model Mining).



Aan mijn grootouders, in liefdevolle herinnering.

Contents

1 Introduction 1
- 1.1 Overview . . . 4

2 Motivation and Preliminaries 7
- 2.1 Preliminaries . . . 10

3 The Exceptional Model Mining Framework 13
- 3.1 Search Strategy . . . 15
  - 3.1.1 Refinement Operator and Description Language . . . 16
  - 3.1.2 Beam Search Algorithm for Top-q EMM . . . 18
  - 3.1.3 Alternatives to Beam Search . . . 21
- 3.2 How to Define an EMM Instance? . . . 22
  - 3.2.1 Quality Measure Concepts . . . 22
  - 3.2.2 Compared to what? . . . 24
- 3.3 Related Work . . . 26
  - 3.3.1 Search Strategies for SD/EMM . . . 26
  - 3.3.2 Similar Local Pattern Mining Tasks . . . 27
  - 3.3.3 Similar Tasks with a Broader Scope . . . 29
- 3.4 Software . . . 31

4 Deviating Interactions – Correlation Model 33
- 4.1 Quality Measure φ_{scd} . . . 33
- 4.2 Experiments . . . 34
  - 4.2.1 Datasets . . . 34
  - 4.2.2 Experimental Results . . . 35
- 4.3 Alternatives . . . 38
- 4.4 Conclusions . . . 40

5 Deviating Predictive Performance – Classification Model 41
5.1 Quality Measure φ_{sed} . . . 42
5.2 Experiments . . . 42
5.2.1 Datasets . . . 42
5.2.2 Experimental Results . . . 42
5.3 Alternatives . . . 43
5.3.1 BDeu Score (φ_{BDeu}) . . . 44
5.3.2 Hellinger (φ_{Hel}) . . . 44
5.3.3 Experimental Results . . . 45
5.4 Conclusions . . . 47

6 Unusual Conditional Interactions – Bayesian Network Model 49
6.1 Quality Measure φ_{weed} . . . 50
6.1.1 Independence Relations in Bayesian Networks . . . 51
6.1.2 Edit Distance for Bayesian Networks . . . 52
6.2 Experiments . . . 54
6.2.1 Datasets . . . 54
6.2.2 Experimental Results . . . 55
6.3 Alternatives . . . 63
6.4 Conclusions . . . 66

7 Different Slopes for Different Folks – Regression Model 69
7.1 Quality Measure φ_{Cook} . . . 70
7.2 Experiments . . . 73
7.2.1 Datasets . . . 73
7.2.2 Experimental Results . . . 76
7.3 Pruning with Bounds for Cook's Distance . . . 80
7.3.1 Empirical bound evaluation . . . 83
7.4 Alternatives . . . 86
7.5 Conclusions . . . 87

8 Exploiting False Discoveries – Validating Found Descriptions 91
8.1 Problem Statement . . . 92
8.2 Validation Method . . . 93
8.2.1 Randomization Techniques . . . 94
8.2.2 Building a Statistical Model . . . 96
8.2.3 Comparing Quality Measures . . . 97

8.3 Experiments . . . 97
8.3.1 Validating Descriptions . . . 101
8.3.2 Validating Quality Measures . . . 102
8.3.3 Validating EMM Results . . . 105
8.4 Discussion . . . 107
8.4.1 Validating Descriptions . . . 108
8.4.2 Validating Quality Measures . . . 108
8.4.3 Validating EMM Results . . . 110
8.5 Related Work . . . 110
8.6 Conclusions . . . 112

9 Multi-label LeGo – Enhancing Multi-label Classifiers with Local Patterns 115
9.1 The LeGo Framework . . . 116
9.2 Multi-label Classification . . . 118
9.3 LeGo Components . . . 120
9.3.1 Local Pattern Mining Phase . . . 120
9.3.2 Pattern Subset Discovery Phase . . . 120
9.3.3 Global Modeling Phase . . . 122
9.4 Experimental Setup . . . 123
9.4.1 Evaluation Measures . . . 124
9.4.2 Statistical Testing . . . 125
9.5 Experimental Evaluation . . . 126
9.5.1 Feature Selection Methods . . . 126
9.5.2 Evaluation of the LeGo Approach . . . 127
9.5.3 Evaluation of the Decompositive Approaches . . . 131
9.5.4 Efficiency . . . 133
9.6 Discussion and Related Work . . . 134
9.7 Conclusions . . . 136

10 Conclusions 139

References 145

Nederlandse Samenvatting 157

English Summary 159

Acknowledgments 161

Curriculum Vitae 163

Chapter 1

Introduction

In their seminal 1996 paper [30], Fayyad, Piatetsky-Shapiro, and Smyth outlined their view on data mining, and what they called *KDD*, the then-emerging field of *Knowledge Discovery in Databases*. The basic problem that KDD strives to solve is the following: when presented with a set of raw data (which is usually too voluminous to inspect manually), distill information out of that dataset that is more compact, more abstract, or more useful. The authors wrote: "KDD is an attempt to address a problem that the digital information era made a fact of life for all of us: data overload." Since then, the internet has evolved from an additional source of information that we would occasionally dial into, to an always available vital necessity. Add to that the recent smartphone penetration into everyone's daily life, and we see that every person and company in the world generates more and more data. Hence the need for KDD methods has become evermore pressing.

Fayyad et al. divide the KDD process into nine stages, the seventh of which is *Data Mining*. After understanding the application domain, creating a dataset, cleaning and projecting the dataset, hypothesis selection and a few other preparatory steps, we arrive at the stage where we can search within a given dataset for "patterns of interest in a particular representational form or a set of such representations", before going to subsequent stages where patterns are interpreted and acted upon. In this dissertation we are mainly occupied with a subfield of data mining (the seventh stage of KDD), with some additional pattern interpretation (the eighth stage of KDD).

In the data mining phase, a given dataset is assumed. One can distinguish several methods to mine the dataset. The following were discussed by Fayyad et al.

Classification: mapping records of the dataset into one or several classes;

Regression: mapping records of the dataset to a real-valued prediction variable;

Clustering: identifying a finite set of categories to describe the dataset;

Summarization: finding a compact description for a subset of the dataset;

Dependency Modeling: finding a model that describes significant dependencies between variables;

Change and Deviation Detection: discovering substantial deviations in the data from the normative, or from previously measured values.

The data mining task we consider in this dissertation combines aspects of the last three methods, and has an application in the first.

The goal of *Local Pattern Mining* (LPM) is to find subsets of the dataset at hand, that are *interesting* in some sense. The goal is not to partition the dataset, and not to classify the dataset. We rather strive to pinpoint multiple (potentially overlapping) interesting subsets at the same time. The interestingness of a subset is gauged without considering the (lack of) coherence of records in the complement of the dataset, and without considering to what extent its interestingness is already represented by other found subsets: subsets are judged purely on their own merit. In LPM we are not quite interested in just any subset of the dataset; we are usually striving to find *subgroups*: subsets of the dataset that can be succinctly described in terms of conditions on attributes of the dataset. In this respect, LPM resembles the **Summarization** method introduced above. Originally, LPM was introduced as an *unsupervised* task where the *interestingness* was measured in terms of an unusually high frequency of a co-occurrence. In terms of such an interestingness definition, LPM resembles the **Deviation Detection** method introduced above.

The simplest form of *supervised* Local Pattern Mining is *Subgroup Discovery* (SD). In this task, one nominal attribute of the dataset is designated as the *target*. SD then strives to find subgroups of the dataset, for which

this target has an unusual distribution. Exceptionality of the distribution is usually gauged in terms of the relative frequencies of target values within the subgroup, compared to these frequencies on the whole dataset, and in terms of the size of the subgroup.

Unsupervised Local Pattern Mining (finding subgroups based on high frequency) and Subgroup Discovery (finding subgroups based on the distribution of one target) are interesting tasks. However, they do not encompass all possible forms of "interesting" subgroups of the dataset. In this dissertation we introduce the *Exceptional Model Mining* (EMM) framework, to accomodate a more general form of interestingness. In the EMM framework, the attributes of the dataset are partitioned into two: one part (the *descriptors*) is used to *define* subgroups on, and one part (the *targets*) is used to *evaluate* subgroups on. The concept of interest in subgroups is captured by learning, from (a subset of) the dataset, a *model* fitted on the targets. The goal of EMM in general is to find subgroups for which the model learned from the records belonging to the subgroup, has parameters that deviate substantially from the parameters of the model learned from the whole dataset. Alternatively, one can compare with the model learned from the complement of the subgroup; this choice will be discussed in detail in Section 3.2.2. EMM is instantiated by selecting two things: a *model class*, which indicates the type of interplay between targets we strive to find deviations for, and a *quality measure*, which quantifies the dissimilarity between two models from the model class. Striving to find unusual interplay between several targets, is where EMM resembles the **Dependency Modeling** method introduced by Fayyad et al.

To illustrate the difference between these Local Pattern Mining tasks, consider the following examples of subgroups one can find with them. In unsupervised LPM, there is no designated target attribute. One could find the subgroup of customers of a supermarket, that simultaneously buy coffee and milk. In Subgroup Discovery, suppose that the target is whether a person develops lung cancer. One could find the subgroup of smokers, whose lung cancer incidence is above average. In Exceptional Model Mining, suppose that the price of a house and its associated lot size are the two targets. One could then find the subgroup of inner city houses, for which the correlation between the two targets is substantially weaker than for the average house.

1.1 Overview

This dissertation consists of ten chapters, of which this introduction is **the first**. In this section, we shortly outline the remaining chapters, discussing the previous publications on which they are based, and giving the appropriate credits to (co-)authors.

In **Chapter 2: Motivation and Preliminaries**, we give motivating examples for Exceptional Model Mining, and introduce some notation. The examples have been discussed before in publications [23] and [25].

In **Chapter 3: The Exceptional Model Mining Framework**, we introduce the general Exceptional Model Mining framework. The EMM concept, including the introduction of the refinement operator, has appeared before in a paper by D. Leman, A. Feelders, and A. Knobbe, published in the proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2008) [71]. The remainder of Chapter 3, discussing our choices for the refinement operator and description language, algorithm and complexity analysis, how to define an EMM instance, related work, and the used software, is new.

The four subsequent chapters all introduce one choice of model class for EMM. None of these chapters explicitly discuss related work; since they instantiate the general framework of Chapter 3, we discuss all relevant related work there.

In **Chapter 4: Deviating Interactions – Correlation Model**, we discuss the EMM instance with the correlation between two numeric targets as model class. The original idea for this model class was first published by D. Leman, A. Feelders, and A. Knobbe, in the proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML / PKDD 2008) [71]. In Chapter 4, we reinterpret their work, and put it in the more general EMM context.

In **Chapter 5: Deviating Predictive Performance – Classification Model**, we discuss the EMM instance with a classifier on several unrestricted targets and one discrete output target as model class. Again, the original idea for this model class was first published by D. Leman, A. Feelders, and A. Knobbe [71], but the interpretation and EMM contextualization are new.

In **Chapter 6: Unusual Conditional Interactions – Bayesian Network Model**, we discuss the EMM instance with a Bayesian network on several nominal targets as model class. This work was published by W. Duivesteijn, A. Knobbe, A. Feelders, and M. van Leeuwen, in the proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010) [25].

In **Chapter 7: Different Slopes for Different Folks – Regression Model**, we discuss the EMM instance with a linear regression model on multiple targets as model class. In addition to the standard content of an EMM instance chapter, this chapter also contains a discussion of pruning the EMM search space with bounds on the developed quality measure. This work was published by W. Duivesteijn, A. Feelders, and A. Knobbe, in the proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2012) [23]. The idea for the simpler, alternative model class described in Section 7.4 was first published by D. Leman, A. Feelders, and A. Knobbe [71]; its interpretation and contextualization are new.

Having discussed Exceptional Model Mining instances, the following two chapters are dedicated to a related and an extended task. Contrary to the preceding four chapters, these following two chapters do come with their own related work discussions.

In **Chapter 8: Exploiting False Discoveries – Validating Found Descriptions**, we develop a method to determine the statistical significance of the outcome of supervised Local Pattern Mining tasks, such as Exceptional Model Mining. The quality of found descriptions is gauged against a model built over artificially generated false discoveries, to refute the hypothesis that a found description is also a false discovery. This method is additionally used to objectively compare different quality measures for the same task, by virtue of their capability to distinguish true from false discoveries. This work was published by W. Duivesteijn and A. Knobbe, in the proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2011) [24].

In **Chapter 9: Multi-label LeGo – Enhancing Multi-label Classifiers with Local Patterns**, we explore the additional value of descriptions found through EMM for the improvement of a global model. The descriptions found with the EMM instance in Chapter 6, with the Bayesian network model as target concept, highlight regions in the dataset where interplay between the targets is unusual. The ability to capture such interplay between labels is

what elevates a multi-label classifier over multiple single-label classifiers. Hence, employing the descriptions as binary attributes for a multi-label classifier should improve classifier performance. In this chapter we discuss the extent to which this *LeGo approach* [37, 57] indeed improves performance. This work was published by W. Duivesteijn, E. Loza Mencía, J. Fürnkranz, and A. Knobbe, in the proceedings of the 11$^{\text{th}}$ International Symposium on Intelligent Data Analysis (IDA 2012) [26]. An extended version was published by the same authors as a technical report of the Technische Universität Darmstadt [27]. This being joint work involving another PhD student, two reinterpretations and contextualizations of publications [26] and [27] are available. The one is Chapter 9 of this dissertation, and the other has appeared as a chapter in the Ph.D. dissertation of E. Loza Mencía [73].

In **Chapter 10: Conclusions**, we draw general conclusions from all preceding chapters. We discuss rationales why Exceptional Model Mining is not only a desireable, but also a practically useful framework to have.

Chapter 2

Motivation and Preliminaries

Finding elements that behave differently from the norm in a dataset is a task of paramount importance. Most data mining research in this direction focuses on *detecting* outliers: simply identifying the peculiarly-behaving records. The characteristic feature of local pattern mining techniques that separates them from such outlier detection methods, is that in local pattern mining, we are not just looking for any outlying record or set of records in the data. Instead, we are looking for subgroups: coherent subsets for which we can formulate a concise description in terms of conditions on attributes of the data. The existence of such descriptions makes the subgroups more actionable: if we can tell a drug manufacturer that ten of his patients react badly to a certain type of medication, this doesn't help him much, but if we can tell him instead that the group of smokers under the age of thirty react badly, this gives the manufacturer a clear indication in which direction to find a solution to his problem.

When the target concept in a dataset can no longer be captured by one particular attribute, but we still want to find exceptional subgroups in the dataset, we find a need for Exceptional Model Mining. As an example of a relatively complex target concept, consider the research performed by Robert T. Paine in 1963 and 1964 in Makah Bay, Washington [86]. It concerns the carnivore starfish *Pisaster ochraceus* whose presence kept a marine ecosystem consisting of 15 species stable. In this system, the sponge *Haliclona* was browsed upon by the nudibranch *Anisodoris*. When *Pisaster* was artificially removed, the bivalve *Mytilus californianus* and

the barnacles *Balanus glandula* and *Mitella polymerus* rapidly grew and crowded out other species. In total, only 8 species remained. Also, the sponge-nudibranch food chain was displaced, and the anemone population was reduced in density. Counterintuitively, when present, *Pisaster* did not eat any of these last three species.

In the studied ecosystem, *Pisaster* was the top carnivore: it consumed other species, but no other species consumed him, and *Pisaster* was the only species in the system for which both these statements held. This made Paine et al.'s research very relevant from a biological point of view; up until that point, it was generally assumed that removing the top carnivore from an ecosystem would increase diversity, but the *Pisaster* experiment proved that that was not necessarily the case.

Paine remarks that the food chains are strongly influenced by *Pisaster*, but by an indirect process. When dealing with a dataset detailing the presence of individual species, existing methods can probably detect simple patterns in the ecosystem, such as the growth of *Mytilus*, *Balanus* and *Mitella* and the decline in the number of species when *Pisaster* is removed. However, the more indirect influence of *Pisaster* on processes such as a food chain it is not direcly related to, for instance between *Haliclona* and *Anisodoris*, cannot be found by looking at single species or even correlations between pairs of species: the (in-)dependence between *Haliclona* and *Anisodoris* is conditional on the presence of *Pisaster*.

Paine models the food chains in the ecosystem as a Bayesian network. In order to find subgroups where the food chains between species are substantially different from the norm, we need to be able to detect the indirect processes that can be captured with a Bayesian network. Using an Exceptional Model Mining instance, we can for instance find subgroups defined by environmental parameters in which complex food chains are displaced. The ability to cope with Bayesian networks makes the same EMM instance applicable to datasets from such diverse fields as information retrieval [9], traffic accident reconstruction [18], medical expert systems [20], gene expression in computational biology [33], and financial operational risk [82].

Another EMM instance could for example be used to find evidence for the *Giffen effect* in data. This effect can be seen as a form of Simpson's

paradox for regression models. The economic law of demand states that, all else equal, if the price of a good increases, the demand for the product will decrease. Sir Robert Giffen described conditions under which this law does not hold [77]. The classic example concerns extremely poor households, who mainly consume cheap staple food, and relatively rich households in the same neighborhood, who can afford to enrich their meals with a luxury food. In this situation, when the price of the staple food increases, there will be a point where the relatively rich households can no longer afford the luxury food. These people need to uphold their calorie intake. Hence, they react by consuming more of the cheapest food available to them, which is the staple food whose price just increased. For the relatively rich households in this poor neighborhood, an increase in the price of the staple food, will lead to an increase in the demand for the staple food. Notice that this relation does not hold for the extremely poor households: they consume only the staple food to begin with, so when the price increases they can simply afford less of it.

For a long time, the Giffen effect was a controversial theory in Economics, since no real-life dataset featuring the effect was available. In 2008, more than a century after the theorem was formulated for the first time, Nolan and Jensen published a paper [53] containing the first real-world dataset containing the Giffen effect, for rice in Hunan, China. Their field study entailed distributing vouchers among randomly drawn households, with which the recipients could buy rice at a lower price. The authors monitored the price of and the demand for rice before, during, and after the voucher programme, as well as a plethora of alternative factors that could influence demand. The relation between the demand for rice and the influencing factors (including the price of rice) was captured by a regression model. Nolan and Jensen observed that the households consuming less than 80% of their calorie intake through rice, i.e. the relatively rich households in this poor neighborhood, displayed the Giffen effect, while the other households did not.

The group of relatively rich households in a poor neighborhood is a subgroup. The subgroup displays an unusual interaction between multiple targets, as captured by the regression model. Hence, subgroups displaying the Giffen effect can be automatically detected by an Exceptional Model Mining instance, mining for an unusual slope of a regression line.

2.1 Preliminaries

Having motivated Exceptional Model Mining in the previous section, we will formally introduce the framework in the next chapter. To that end, we first introduce some definitions and notations that will be used throughout the remainder of this dissertation. Any symbol introduced in this section may pop up at any given moment; we assume its meaning to be understood by the reader from this point on.

We assume a dataset Ω to be a bag of N *records* $r \in \Omega$ of the form

$$r = (a_1, \ldots, a_k, \ell_1, \ldots, \ell_m)$$

where k and m are positive integers. We call $a_1, \ldots, a_k$ the *descriptive attributes* or *descriptors* of r, and $\ell_1, \ldots, \ell_m$ the *target attributes* or *targets* of r. The descriptors are taken from an unrestricted domain $\mathcal{A}$. In later chapters we will learn models from a selected *model class* over the targets; restrictions on the type of each target may be imposed by the choice of model class. We refer to (elements of) the i^{th} record by superscript i.

For our definition of subgroups we need to define *descriptions*. In practice, descriptions will usually be taken from a description language $\mathcal{D}$, to be chosen by the user. We will leave this concept abstract for now; a particular choice we make for $\mathcal{D}$ will be discussed in Section 3.1.1. However, mathematically, we will define descriptions as functions $D : \mathcal{A} \to \{0, 1\}$. A description D *covers* a record r^i if and only if $D\left(a_1^i, \ldots, a_k^i\right) = 1$.

Definition (Subgroup). A *subgroup* corresponding to a description D is the bag of records $G_D \subseteq \Omega$ that D covers, i.e.

$$G_D = \left\{ r^i \in \Omega \;\middle|\; D\left(a_1^i, \ldots, a_k^i\right) = 1 \right\}$$

From now on we omit the D if no confusion can arise, and refer to a subgroup as G. We will freely associate subgroups with their descriptions and vice versa. Also, the 'patterns' in the commonly used term 'Local Pattern Mining' are equivalent to our descriptions, and hence imply subgroups. These terms will all be used interchangably when a clear separation between the concepts is not necessary. Whenever it is clear that we have a particular subgroup G in mind, we will write n for the number of records in

that subgroup: $n = |G|$, to which we will also refer as the *coverage* of the description. The complement of a subgroup is denoted by G^C, and for its number of records we write n^C. Hence, $G^C = \Omega \backslash G$, and $n^C = N - n$.

In order to objectively evaluate a candidate description in a given dataset, we need to define a *quality measure*. For each description D in the description language $\mathcal{D}$, this function quantifies the extent to which the subgroup G_D induced by the description deviates from the norm.

Definition (Quality Measure). A *quality measure* is a function $\varphi : \mathcal{D} \to \mathbb{R}$ that assigns a unique numeric value to a description D.

Since descriptions imply subgroups, we will occasionally write $\varphi(G)$ and refer to the quality of a subgroup.

Chapter 3

The Exceptional Model Mining Framework

Exceptional Model Mining [23, 25, 71] is a data mining framework that can be seen as a generalization of the Subgroup Discovery (SD) [50, 55, 114] framework. SD strives to find descriptions that satisfy certain user-specified constraints. Usually these constraints include lower bounds on the quality of the description ($\varphi(D) \geq lb_1$) and size of the induced subgroup ($|G_D| \geq lb_2$). More constraints may be imposed as the question at hand requires; domain experts may for instance request an upper bound on the complexity of the description. Most common SD algorithms traverse[1] the search space of candidate descriptions in a general-to-specific way: they treat the space as a lattice whose structure is defined by a *refinement operator* $\eta : \mathcal{D} \rightarrow 2^{\mathcal{D}}$. This operator determines how descriptions can be extended into more complex descriptions by atomic additions. Most applications (including ours) assume η to be a *specialization operator*: every description D_i that is an element of the set $\eta(D_j)$, is more specialized that the description D_j itself. The algorithm results in a ranked list of descriptions (or the corresponding subgroups) that satisfy the user-defined constraints.

In traditional SD, subgroup exceptionality is measured in terms of the distribution of only a single target variable. Hence, the typical quality measure contains a component indicating how different the distribution over the target variable in the subgroup is, compared to its distribution

[1] we consider the exact search strategy to be a parameter of the algorithm

in the whole dataset. Since unusual distributions are more easily achieved in small subsets of the dataset, the typical quality measure also contains a component indicating the size of the subgroup. Thus, whether a description is deemed interesting depends on both its exceptionality and the size of the corresponding subgroup.

EMM can be seen as a generalization of SD. Rather than one single target variable, EMM uses a more complex target concept. An instance of Exceptional Model Mining is defined by the combination of a *model class* over the targets, and a *quality measure* over this model class. When an instance has been defined, subgroups are generated (we will discuss how in the next section) to be evaluated. Then, for each subgroup under consideration, we induce a model on the targets. This model is learned from only the data belonging to the subgroup. Using the quality measure, the subgroup is evaluated based on model characteristics, to determine which subgroups are the most interesting ones. The typical quality measure in EMM indicates how exceptional the model fitted on the targets in the subgroup is, compared to either the model fitted on the targets in its complement, or the model fitted on the targets in the whole dataset — we will discuss this fundamental choice in Section 3.2.2. Just like in traditional SD, exceptional models are sometimes easily achieved in small subgroups, so if necessary, an EMM quality measure also contains a component indicating the size of the subgroup.

As we will explore in Section 3.2, there are several canonical choices that can be made when designing a quality measure for a selected model class. However, the framework allows a quality measure to be any function assigning a quality quantification to a description. This allows EMM to search for just about any imaginable instantiation of "interesting" subgroups.

So far, we have talked about Exceptional Model Mining in an informal, colloquial manner. This is deliberate. The goal is to find interesting subgroups of a dataset, for whatever instantiation of "interesting" the user of EMM cares for, which is intrinsically subjective. Therefore, any formal definition of the EMM task will only concern a subset of what we attempt to achieve with EMM. Nevertheless, to provide a more precise handle on what we will be concerned with in the following chapters, we can consider the following task definition

Problem Statement (Top-q Exceptional Model Mining). *Given a dataset* Ω*, description language* $\mathcal{D}$*, quality measure* φ*, positive integer* q*, and set of constraints* $\mathcal{C}$*. The Top-*q *Exceptional Model Mining task delivers the list* $\{D_1, \ldots, D_q\}$ *of descriptions in the language* $\mathcal{D}$ *such that*

* $\forall_{1 \leq i \leq q} : D_i$ *satisfies all constraints in* $\mathcal{C}$*;*
* $\forall_{i,j} : i < j \Rightarrow \varphi(D_i) \geq \varphi(D_j)$*;*
* $\forall_{D \in \mathcal{D} \setminus \{D_1, \ldots, D_q\}} : D$ *satisfies all constraints in* $\mathcal{C} \Rightarrow \varphi(D) \leq \varphi(D_q)$.

Informally, we find the q best-scoring descriptions in the description language that satisfy all constraints in $\mathcal{C}$. This set encompasses both user-induced constraints and search strategy limitations. These limitations include information about the exact choice we make for the refinement operator η, guiding how new candidate subgroups are generated out of other subgroups, and the limits to which we will explore the search space. In the following section we discuss the choices made for the search space traversal and the refinement operator in the remainder of this dissertation. Note that the general EMM *framework* leaves the choice for these matters open.

Also noteworthy is the fact that this problem statement includes the traditional Subgroup Discovery problem. This is a feature rather than a bug: we consider SD to be encompassed by EMM. In our view, Subgroup Discovery is simply a version of Exceptional Model Mining in which m, the number of targets as introduced in Section 2.1, is set to 1.

3.1 Search Strategy

Since the goal of SD/EMM is to find interesting subsets of the data, the corresponding search space is potentially exponentially large. Hence, we cannot simply explore this space by brute force; we need to find a more sophisticated search strategy. Part of the problem is already solved by only allowing subgroups. Since subgroups are subsets of the data for which a description exists, the set of subgroups is smaller than the set of subsets (although exactly *how much* smaller the set is, depends on the choice of description language $\mathcal{D}$). When many attributes in the dataset are numeric, the difference is not very substantial.

There are two main schools of thought in the community on how to overcome this problem, each with their own focus. The one, following canonical SD papers [55, 114], restricts the attributes in the dataset to be nominal and imposes an anti-monotonicity constraint on the used quality measure. Then the resulting search space can occasionally be explored exhaustively. The other resorts to heuristic search. This allows the attributes to be numeric as well, and facilitates a general quality measure. Since EMM is developed to capture any concept of interestingness in subgroups, we find allowing for any quality measure and numeric attributes more important than exhaustiveness. Hence we select the heuristic path. Exhaustive SD methods will be discussed in further detail in Section 3.3.1.

In the EMM setting, usually the *beam search* strategy is chosen, which performs a level-wise search. On each level, the best w (for *search width*) descriptions according to our quality measure φ are selected, and refined to create the candidate descriptions for the next level. The search is constrained by an upper bound on the complexity of the description (also known as the *search depth*, d) and a lower bound on the support of the corresponding subgroup. This search strategy combines the advantages of a greedy method with those of the implicit parallel search: as on each level w alternatives are considered, the search process is less likely to end up in a local optimum than a pure greedy approach, while selecting the w best descriptions at each level keeps the process focused, hence tractable.

3.1.1 Refinement Operator and Description Language

An important part of the beam search strategy is generating the set of candidate descriptions for the next level, by refining another description. This process is guided by the refinement operator η and the description language $\mathcal{D}$, for which we detail our choices in this section. Our description language $\mathcal{D}$ consists of logical conjunctions of conditions on single attributes.

We treat the numeric attributes with a particular kind of discretization, starting by fixing a positive integer $b \leq N$ (the number of *bins*) before the EMM process starts. On the first search level, when the generating description has no conditions, the discretization we apply is equal to static pre-algorithm discretization of the attribute into b bins of equal size. How-

ever, on each subsequent search level, our generating descriptions consist of a positive number of conditions, hence they cover strictly less than N records. Since on these levels we consider a discretization into b equal-sized bins of the attribute-values *within the generating non-empty description*, the bins may be different for each generating description. This *dynamic discretization* during the process draws more information from the attribute than we would use when statically discretizing it beforehand.

When η is presented with a description D to refine, it will build up the set $\eta(D)$ by looping over all the descriptive attributes $a_1, \ldots, a_k$. For each attribute, a number of descriptions will be added to the set $\eta(D)$, depending on the attribute type

if a_i is binary: add $D \cap (a_i = 0)$ and $D \cap (a_i = 1)$ to $\eta(D)$;

if a_i is nominal, with values $v_1, \ldots, v_g$: add $\{\ D \cap (a_i = v_j),\ D \cap (a_i \neq v_j)\ \}_{j=1}^{g}$ to $\eta(D)$;

if a_i is numeric: order the values of a_i that are covered by the description D; this gives us a list of ordered values $a_{(1)}, \ldots, a_{(n)}$ (where $n = |G_D|$). From this list we select the split points $s_1, \ldots, s_{b-1}$ by letting

$$\forall_{j=1}^{b-1} : s_j = a_{\left(\lfloor j\frac{n}{b} \rfloor\right)}$$

Then, add $\{\ D \cap (a_i \leq s_j),\ D \cap (a_i \geq s_j)\ \}_{j=1}^{b-1}$ to $\eta(D)$.

Informally, when presented with a description D, η will build a set of refinements by considering the descriptive attributes one by one. Each such refinement will consist of the conditions already present in D, plus one new condition. If an encountered attribute a_i is binary, 2 refined descriptions will be added to $\eta(D)$: one for which D holds and a_i is true, and one for which D holds and a_i is false. If the attribute a_i is nominal with g different values, $2g$ refined descriptions will be added to $\eta(D)$: for each of the g values, one where D holds and the value is present, and one where D holds and any of the $g-1$ other values is present. If the attribute a_i is numeric, we divide the values for a_i *that are covered by* D into a predefined number b of equal-sized bins. Then, using the $b-1$ *split points* $s_1, \ldots, s_{b-1}$ that separate the bins, $2(b-1)$ refined descriptions will be added to $\eta(D)$: for each split point s_j, one where D holds and a_i is less than or equal to s_j, and one where D holds and a_i is greater than or equal to s_j.

3.1.2 Beam Search Algorithm for Top-q EMM

Having described our choices for the search strategy and refinement operator that we will use in the remainder of this thesis, we can now describe and analyze an algorithm for the top-q Exceptional Model Mining problem stated earlier in this chapter. The pseudocode is given in Algorithm 1. In the algorithm, we assume that there is a subroutine called SATISFIESALL that tests whether a candidate description satisfies all conditions in a given set. Among the abstract datastructures we assume, the Queue is a standard queue with unbounded length. The PriorityQueue(x) is a queue containing at most x elements, where elements are stored and sorted with an associated quality; only the x elements with the highest qualities are retained, while other elements are discarded. In a straightforward but not too naive implementation, a PriorityQueue is built with a heap as its backbone. In this case the elementary operations, *insert_with_priority* for adding an element to the PriorityQueue and *get_front_element* for removing the element with the highest quality from the PriorityQueue, have a computational cost of $\mathcal{O}(\log x)$ [60, pp. 148–151].

Many statements in the algorithm control the beam search process in a straightforward manner. However, the process is also controlled by the interplay between the different (Priority-)Queues, which is more intricate and deserves attention. The resultSet is a PriorityQueue maintaining the q best found descriptions so far. Nothing is ever explicitly removed from the resultSet, but if the quality of a description is no longer among the q best, it is automatically discarded. Hence, the resultSet maintains the final result that we seek. The beam is a similar PriorityQueue, but with a different role. Here, the w best found descriptions so far *on the current search level* are maintained. When all candidates for a search level have been explored, the contents of the beam are moved into the unbounded but (by then) empty Queue candidateQueue, to generate the candidates for the next level.

Complexity

Since EMM is a highly parametrized algorithm, instantiated by a model class and quality measure, we need to introduce some notation before we

Algorithm 1 Beam Search for Top-q Exceptional Model Mining

Input: Dataset Ω, QualityMeasure φ, RefinementOperator η, Integer w, d, q, Constraints $\mathcal{C}$
Output: PriorityQueue resultSet

```
1 : candidateQueue ← new Queue;
2 : candidateQueue.enqueue({});              ▷ Start with empty description
3 : resultSet ← new PriorityQueue(q);
4 : for (Integer level ← 1; level ≤ d; level++) do
5 :     beam ← new PriorityQueue(w);
6 :     while (candidateQueue ≠ ∅) do
7 :         seed ← candidateQueue.dequeue();
8 :         set ← η(seed);
9 :         for all (desc ∈ set) do
10 :            quality ← φ(desc);
11 :            if (desc.SATISFIESALL(C)) then
12 :                resultSet.insert_with_priority(desc,quality);
13 :                beam.insert_with_priority(desc,quality);
14 :    while (beam ≠ ∅) do
15 :        candidateQueue.enqueue(beam.get_front_element());
16 : return resultSet;
```

can analyze the computational complexity of the algorithm. We write $M(n, m)$ for the cost of learning a model from n records on m targets, and c for the cost of comparing two models from the chosen model class.

Theorem 1. *The worst-case computational complexity of Algorithm 1 is*

$$\mathcal{O}\left(dwkN\left(c + M(N, m) + \log(wq)\right)\right)$$

Proof. We start our analysis at the innermost loop, working bottom-up. Line 12 inserts an element into a PriorityQueue of size q, which costs $\mathcal{O}(\log q)$. Line 13 does the same for a PriorityQueue of size w, and hence costs $\mathcal{O}(\log w)$. The conditions checked in line 11 are the user-induced constraints a domain expert may impose on the resulting descriptions. These usually are relatively simple conditions concerning for instance a minimal number of records covered by the descriptions. As such, they are relatively cheap to check. For all reasonable constraints a domain expert may come

up with, the necessary information can be extracted during the same scans of the dataset we need when, for instance, computing the quality of the description in the preceding line. As such, we assume the computational complexity of line 11 to be dominated by the complexity of line 10. The worst-case scenario is that all descriptions pass the test, hence the commands inside the if-loop need to be computed every time. Thus, the total complexity of lines 11 through 13 is $\mathcal{O}(\log w + \log q) = \mathcal{O}(\log(wq))$.

Line 10 computes the quality of a description. In the worst case, this requires the learning of two models: one on the description and one on its complement, and comparing these models. Hence: $\mathcal{O}(c + 2M(N, m))$ $= \mathcal{O}(c + M(N, m))$ (recall the definition of c and $M(N, m)$, as introduced just before Theorem 1). In line 9, a loop is run for all refinements of a seed description. By our choice of η, the worst case would be if every descriptive attribute were nominal (or numeric) having N distinct values. For each of the k descriptors (cf. Section 2.1), we would then generate $2N$ refinements. The loop is thus repeated $2kN$ times, which costs $\mathcal{O}(kN)$. Hence, the total complexity of lines 9 through 13 is $\mathcal{O}(kN(c + M(N, m) + \log(wq)))$.

Line 8 enumerates all refinements of one description, which we have just analyzed to cost $\mathcal{O}(kN)$. Line 7 dequeues an element from an ordinary Queue, which can be done in $\mathcal{O}(1)$. Line 6 loops all previously analyzed lines as many times as there are elements in the candidateQueue. This queue never has more than w elements, since it is always emptied before (in line 15) at most w new elements are added to the queue. Hence, the total complexity of lines 6 through 13 is $\mathcal{O}(w(kN + kN(c + M(N, m) + \log(wq))))$ $=$ $\mathcal{O}(wkN(c + M(N, m) + \log(wq)))$.

On the same level we find line 5, which costs $\mathcal{O}(1)$, and the while-loop of lines 14 through 15, which costs $\mathcal{O}(w \log w)$ if done extremely naively. These lines are dominated in complexity by lines 6 through 13. All these lines are enveloped by a for-loop starting at line 4, which is repeated d times. Lines 1 through 3 and 16 can be computed in constant time, and so the total computational complexity of Algorithm 1 becomes

$$\mathcal{O}(dwkN(c + M(N, m) + \log(wq))) \qquad \square$$

This complexity seems relatively benign; we see no factors with exponents higher than one, and the worst parameter has complexity $\mathcal{O}(w \log w)$,

which is tractable for a generous range of values for w. However, there are some variables in the complexity expression, which can lead to higher powers of parameters if we fill them in by selecting a model class and quality measure. For instance, if we would perform traditional Subgroup Discovery with this algorithm, we would be searching for descriptions having an unusually high mean for one designated target. Hence, the model computation complexity becomes $M(N, 1) = \mathcal{O}(N)$, and the model comparison cost becomes $c = \mathcal{O}(1)$. Thus, the total computational complexity of Beam Search for Top-q Subgroup Discovery would be $\mathcal{O}(dwkN(N + \log(wq)))$, which is quadratic in the number of records in the dataset.

Note that this computational complexity is in many respects a worst-case scenario, whose bounds a real-life run of the algorithm is unlikely to meet. Since data of such high cardinality is rarely obtained, the number of refinements of a seed description is usually much lower than $2kN$. Also, unlike in the worst-case scenario, the beam search converges in such a way that per search level the subgroups reduce in size, hence the modeling is done over progressively smaller parts of the dataset. Also noteworthy are the facts that when a dataset is extended with more data of the same cardinality, the algorithm scales linearly, and that the number of candidates under consideration is roughly equal per search level, except for level $d = 1$.

3.1.3 Alternatives to Beam Search

Whereas the traditional EMM framework strives to find exceptional descriptions by searching through the descriptive attribute space, and evaluating on the target attribute space, interesting results have been obtained by taking a more symmetrical approach to the two subspaces of the data. The EMDM algorithm [69] strives to effectively find exceptional models by iteratively improving candidate descriptions, exploiting structure in both spaces. Each iteration consists of two steps, one for Exception Maximization (EM) and one for Description Minimization (DM). In the EM step, a compression-based quality measure guides the search for subsets having an unusual model. In the DM step, a rule-based classifier is employed to find a concise description that crafts a subgroup from the found subset. Upon convergence, or when a threshold on the number of iterations is surpassed, the subgroups are reported.

The well-known FP-Growth algorithm has been adapted, to enable exhaustive EMM. Lemmerich et al.'s *generic pattern growth* algorithm (GP-Growth) [72] strives to avoid scanning the whole dataset to evaluate descriptions. Instead, it builds a special data structure, in which the key information of the model learned for a description is summarized. Such a summary is called a *valuation basis*. It contains enough information to determine the quality of any refinement of the description. The GP-Growth algorithm can reduce the memory requirement and runtime of an EMM instance by more than an order of magnitude, but only when a valuation basis can be found that is suitably condensed. This depends on the computational expense of the model class: if a parallel single-pass algorithm with sublinear memory requirements exists to compute the model from a given set of records, profit can be gained from GP-Growth. Most of the model classes we will discuss can benefit from GP-Growth, but in Chapter 6 we will see a model which cannot.

3.2 How to Define an EMM Instance?

As previously described, an EMM instance is defined by the choice of model class over the targets, and quality measure over the model class. In the following four chapters we define several such instances. Before that, we discuss some general themes that recur in EMM instance definitions.

The choice of model class is usually inspired by a real-life problem. For instance, when the goal is to find deviating dependencies between several species in an ecosystem, one is drawn towards graphical models such as Bayesian networks and Markov models. If we can formulate the relation between the targets for which we are interested in finding exceptions, this usually naturally directs our attention to a particular model class.

3.2.1 Quality Measure Concepts

Having chosen a model class, we need to define a quality measure that extracts characteristics from the learned models, and extracts from these characteristics a quantification of how different the models are from each

other. Usually such a quantification is relatively straightforward to design. For instance, if the model class is a regression model with two variables, one could take the difference between the estimated slopes in each model as quality measure. However, such a quantification is typically not enough to design a proper measure for the quality of a description. After all, deviations from the norm are easily achieved in very small subsets of the data. Hence, directly taking a difference quantification as quality measure probably leads to descriptions of very small subgroups, which are usually not the most interesting ones to domain experts. Therefore, we somehow want to represent the size of a subgroup in a quality measure.

In some of the canonical quality measures for Subgroup Discovery, such as Weighted Relative Accuracy (WRAcc) [35], the size of a subgroup is directly represented by a factor n or $\sqrt{n}$. Though their simplicity is appealing, we find it somewhat counter-intuitive to have a factor in a quality measure that explicitly favors subgroups covering the entire dataset over smaller subgroups. A slightly more sophisticated way to represent the subgroup size, is to multiply (i.e. weigh) the quantification of model difference with the *entropy* of the split between the subgroup and its complement. The entropy captures the information content of such a split, and favours balanced splits (1 bit of information for a 50/50 split) over skewed splits (0 bits for the extreme case of either subgroup or complement being empty). The entropy function $\varphi_{ef}(D)$ is defined (in this context) as

$$\varphi_{ef}(D) = -n/N \lg n/N - n^C/N \lg n^C/N$$

Another way to direct the search away from extremely small subgroups, is by employing a quality measure based on a statistical test. For certain models there may be hypotheses of the form

H_0 : model parameter for description = model parameter for complement
H_1 : model parameter for description $\neq$ model parameter for complement

which we can test, usually involving some statistical theory, to derive an expression for which we can compute a p-value. Then, using $1 - p$ as the quality measure, we have constructed a measure ranging from 0 to 1 for which higher values indicate more interesting descriptions.

Sections 4.1 (φ_{scd}), 5.1 (φ_{sed}) and 7.4 (φ_{ssd}) feature examples of quality measures that are directly based on a statistical test. In Sections 4.3 (φ_{ent})

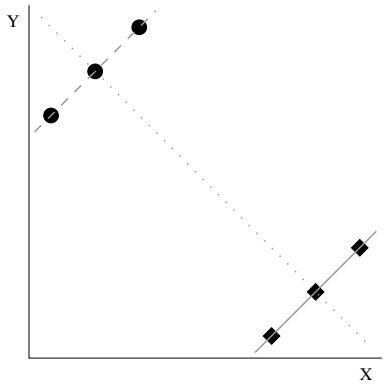


Figure 3.1: Should we compare a subgroup G_D to its complement G_D^C, or to the whole dataset Ω?

and 6.1 (φ_{weed}) we find examples of quality measures employing the entropy function. Quality measures from Sections 4.3 (φ_{abs}), 5.3 (φ_{BDeu} and φ_{Hel}), 6.3 (φ_{ed}), and 7.1 (φ_{Cook}) consist solely of a difference quantification (occasionally these are statistically inspired, but they are not directly based on an established statistical test).

3.2.2 Compared to what?

So far we have discussed quality measure development as a means of assessing how different two learned models are from one another, and how to ensure that subgroups have a substantial size. However, we have neglected a cardinal point. Since a quality measure should assign a quality to a description, its model should be compared, but to which other model? There are two options: we can compare the model for a description of a subgroup G_D either to the model for its complement G_D^C, or to the model for the whole dataset Ω. The simple constructed example from Figure 3.1 illustrates that these two comparisons can lead to very different outcomes.

Suppose that we have a two-dimensional target space, and we are concerned with finding descriptions having a deviating regression line in these two dimensions. Figure 3.1 depicts the target space, and the six records in the example dataset. The dotted grey line is the regression line of the whole dataset, with slope -1. Now suppose that we find the description D covering the records depicted as circles. The dashed grey line is the

regression line of G_D, with slope 1. The solid grey line is the regression line of G_D^C, also having slope 1. When gauging the exceptionality of a description solely by the slope of the regression line, we find G_D interesting when compared to Ω, but not at all when compared to G_D^C. Of course, the assessment changes when we include the intercept in the evaluation.

The problem as displayed in Figure 3.1 is underdetermined; we have not enough information to formulate an opinion on whether the subgroup should be deemed interesting. It can therefore not be used to illustrate whether comparing to G_D^C or to Ω is preferable; it merely illustrates that a different choice may lead to a different outcome.

There is not always a clear-cut preferred choice whether to compare to G_D^C or to Ω. Sometimes, the real-life problem at hand can point in one direction: if we are interested in deviations from a possibly inhomogeneous norm, it makes more sense to compare to Ω, whereas if we are interested in dichotomies, it makes more sense to compare to G_D^C. On other occasions, a statistically inspired quality measure may *require* choosing either Ω or G_D^C, to prevent violation of mathematical assumptions. Lastly, when the model class is so complicated that learning models from data covered by descriptions has a nontrivial computational expense, efficiency might dictate the choice: when comparing n descriptions to Ω, learning $n+1$ models suffices, but when comparing them to G_D^C, learning $2n$ models is required.

The previous two practical considerations supersede any personal preference that we outline; if the model class choice and quality measure design somehow require comparing to either G_D^C or Ω, then that is the way to go. However, when given the choice, we would consider comparing to Ω preferable. After all, Exceptional Model Mining is designed as a Local Pattern Mining task, where we strive to find coherent subsets of the data where something interesting is going on. The goal is to pinpoint many such deviations of the norm, possibly overlapping, without consideration for the coherence and model parameters occurring in the remainder of the dataset. When we compare the model for a subgroup G_D to the model for Ω, we evaluate a subgroup by comparing its behavior to the behavior for the entire dataset. This implies that we strive to find subgroups deviating from the norm. By contrast, when we compare the model for a subgroup G_D to the model for G_D^C, we evaluate a subgroup by comparing its behavior to the

behavior on the complement of the dataset. This implies that we strive to find schisms in the dataset: not necessarily one subgroup deviating from the norm, but rather a partitioning of Ω into two subgroups displaying clearly contrasting behavior. We think this is a very interesting task, but it may not strictly adhere to the goals of Exceptional Model Mining.

3.3 Related Work

Exceptional Model Mining extends a vast body of work, of which this section contains some highlights. First we discuss the search strategies developed to deal with the exponential search space. Then we look into other local pattern mining tasks, and other extensions of Subgroup Discovery. Finally, we discuss how similar questions arise in other data mining disciplines, and what distinguishes them from EMM.

3.3.1 Search Strategies for SD/EMM

When striving to find interesting subsets of a dataset, the search space is exponential in the number of records. By restricting the problem to finding interesting *subgroups*, i.e. subsets with a concise description, the search space remains theoretically exponential in size, but we obtain a handle with which we can tackle the problem. Traditionally [55], this is done by compelling all attributes in the dataset to be nominal. In this case, occasionally exhaustive search is possible, using filters akin to the anti-monotonicity constraints known from frequent itemset mining. When not all attributes are nominal, traditionally there was no other option than to resort to heuristic search.

Recently, Grosskreutz and Rüping developed a new pruning scheme with accompanying SD algorithm, MergeSD [45], which allows for exhaustive mining even when the attributes are taken from a numeric domain. Their key idea is to exploit bounds between related numeric descriptions to prune with optimistic estimates, thus reducing the search space to tractable levels. Unfortunately, the pruning scheme cannot be used with any quality measure; implicitly a constraint similar to anti-monotonicity is imposed.

In work dedicated to expanding the description language $\mathcal{D}$ available to Subgroup Discoverers, Mampaey et al. introduced an efficient treatment of numeric attributes [76]. The description space is not explored exhaustively. Instead, the algorithm finds richer descriptions efficiently, by finding an optimal interval for every numeric attribute, and an optimal value set for every nominal attribute. The efficiency comes from considering only descriptions that lie on a convex hull in ROC space, and evaluating them with a convex quality measure. Hence, the method is only suitable for a target concept that can be properly expressed in ROC space, i.e. traditional SD with a nominal target, and a convex concept of interestingness.

Another problem stemming from the exponential search space is the redundancy in a resulting description set. When a description is deemed interesting, small variations will very likely deliver other descriptions that are also quite interesting. Therefore it is not uncommon, especially when there are numeric attributes in the dataset, to find the top of a description chart dominated by many copies of what technically may all be slightly different descriptions, which in practice all indicate the same underlying concept. Van Leeuwen et al. [70] introduced three degrees of subgroup redundancy, and incorporated selection strategies based on these redundancies in a beam search algorithm. This results in non-exhaustive, but interestingly different search strategies.

The only work so far on exhaustive Exceptional Model Mining, is Lemmerich et al.'s GP-Growth algorithm [72], which was discussed in detail earlier in this chapter. It can severely reduce the memory requirement and runtime of an EMM instance, but only when a parallel single-pass algorithm with sublinear memory requirements exists to compute the model from a given set of records. This can be done for relatively simple model classes, but not for more computationally expensive model classes (cf. Section 3.1.3).

3.3.2 Similar Local Pattern Mining Tasks

Subgroup Discovery research orginated in the mid-nineties, in a simple single-table setting with a binary target attribute [55], and in a multi-relational setting [114]. Tasks that are very similar to, but slightly different

from Subgroup Discovery, include Contrast Set Mining [4], where the goal is to find "conjunctions of attributes and values that differ meaningfully in their distributions across groups", and Emerging Pattern Mining [21], which strives to find itemsets whose support increases substantially from one dataset to another. One could view the latter task as an amalgamation of two separate Subgroup Discovery runs (one for each dataset), followed by a search for classification rules (where a found subgroup has class 1 when found on dataset 1, and class 2 when found on dataset 2). Kralj Novak et al. provide a framework unifying Contrast Set Mining, Emerging Pattern Mining, and Subgroup Discovery [65].

Giving a full overview of all work related to Subgroup Discovery is beyond the scope of this dissertation; such overviews are available in the literature (for instance: [50]). In the remainder of this section we focus on work related to supervised local pattern mining with a more complex goal.

As the antithesis to Contrast Set Mining, Redescription Mining [39, 91] seeks multiple descriptions of the same subgroups, originally in itemset data. Recent extensions incorporate nominal and numeric data [38].

Umek et al. [109] consider Subgroup Discovery with a multi-dimensional output space. They approach this data by considering the output space first: agglomerative clustering in the output space proposes candidate subgroups that have records similar in outcomes. Then, a predictive modeling technique is used to test for each identified candidates whether they can be characterized by a description over the input space.

One of the few papers that explicitly seeks a deviating model over a target attribute, concerns Distribution Rules [54]. In this work, there is only one numeric target, and the goal is to find subgroups for which the distribution over this target is significantly different from the overall distribution, measured in terms of the Kolmogorov-Smirnov test for goodness of fit. Since rules are evaluated by assessing characteristics of a model, this can be seen as an early instance of Exceptional Model Mining, albeit considering only one target attribute.

3.3.3 Similar Tasks with a Broader Scope

General concepts from EMM, like fitting different models to different parts of the data, or identifying anomalies in a dataset, appear in tasks beyond Local Pattern Mining. In this section we discuss a few such tasks, and how they relate to EMM.

In Outlier Detection, traditionally the goal is to identify records that deviate from a general mechanism. Usually there is no desire to find a coherent set of such outliers, which can succinctly be described: identifying non-conforming records is enough. As Outlier Detection becomes more and more mature and sophisticated, we witness more attention towards the underlying mechanism making a point an outlier, for instance in recent work by Kriegel et al. [66]. Their method to detect outliers in arbitrarily oriented subspaces of the original attribute space also delivers an explanation with each outlier, consisting of two parts: an error vector, pointing towards the expected position of the outlier, and an outlier score, quantifying the likeliness that this point is an outlier. Searching for the reason for outliers is a step towards bridging the gap with finding coherent deviating subsets as done in EMM, although the approaches differ vastly. Alternatively, Konijn et al. [63] have designed a hybrid method, post-processing regular Outlier Detection results with a Subgroup Discovery run. This enables higher-level analysis of Outlier Detection results.

When fitting a regression function to a dataset with a complex underlying distribution, one could employ Regression Clustering [116]. The idea is to simultaneously apply $K > 1$ regression functions to the dataset, clustering the dataset into K subsets that each have a simpler distribution than the overall distribution. Each function is then regressed to its own subset, resulting in smaller residual errors, and the regressions and clustering optimize a common objective function. Catering for parts of the dataset where a fitted model is substantially different is a shared idea between Regression Clustering and EMM. However, in Regression Clustering the subsets are not necessarily coherent, easy to describe subgroups: the goal is not to explore exceptionalities, but to give a well-fitting partition.

A similar caveat holds for the well-known Classification And Regression Trees [7], where a nominal or numeric target concept is assigned a different class or outcome depending on conditions on attributes. While the recursive partitioning given by the tree ensures that every path from the root to a leaf constitutes a coherent, easy to describe subgroup, there is again no explicit search for exceptionalities. A partition that performs well is enough, and if multiple exceptional phenomena that happen to have similar effects on the target are found in the same cell of the partition, the CART algorithm judges this as a good outcome while from the Exceptional Model Mining viewpoint it is not.

As an extension of the regression tree algorithm provided by CART, where the leaves contain numeric values as opposed to the classes found in the leaves of a decision tree, the M5 system [90] produces trees having multivariate linear regression models in the leaves. Instead of learning a global model for the entire dataset, M5 partitions the dataset by means of the internal nodes of the tree, and learns a local model for each leaf. Essentially, the resulting tree can be seen as a piecewise linear regression model. M5 can also be seen as a sibling of Regression Clustering, but with an easy-to-describe partition and a hierarchical clustering. As is the case with CART and Regression Clustering, contrary to EMM the goal of M5 is not to find exceptionalities but to completely partition the data, and the focus is on the overall performance in the target space rather than separation of exceptional phenomena.

Contrary to ordinary decision trees, where the classes are found in the leaves of the tree and the internal nodes merely contain conditions for classification, a Predictive Clustering Tree (PCT) [5] has each internal node and each leaf corresponding to a cluster. A cluster is represented by a prototype, and a distance measure is assumed that computes the distance between prototypes hence clusters. Given all this, the decision tree algorithm is adapted to select in each node the condition maximizing the distance between the clusters in its children. Defining a quality measure that finds an optimal separation between a subset of the data and its complement, is a common concept in PCT and EMM. However, the goal of PCT is not to find global exceptionalities, but rather find a partition of the data that is optimal in some sense.

The work on PCTs has been generalized to concern the general problem of mining on a dataset with structure on the output classes, whether this structure takes the form of dependencies between classes (tree-shaped hierarchy, directed acyclic graph) or internal relations between classes (sequences). A tree ensemble method for such data was proposed by Kocev et al. [61]. Their method is able to give different predictions for parts of the dataset that behave differently from the norm. Contrary to EMM, there is no explicit identification of the deviating subgroup and model.

3.4 Software

In the following chapters, we will introduce model classes and quality measures, and run experiments with the corresponding Exceptional Model Mining instances. These experiments are primarily performed with the *Cortana* discovery package [78]: a Java implementation that is an open-source spin-off of the Safarii Data Mining system.

Cortana is not limited to Exceptional Model Mining; it provides multiple supervised Local Pattern Mining tasks. The user can set the task he/she wants Cortana to perform by selecting a *target concept*. For the simplest target concept, SINGLE_NOMINAL, the user must highlight one nominal attribute, and Cortana will perform Subgroup Discovery with that attribute as target. Similarly, for the SINGLE_NUMERIC target concept, one numeric attribute needs singling out, for Cortana to use as numeric target in a Subgroup Discovery run. Several Exceptional Model Mining instances are covered by other target concepts: DOUBLE_CORRELATION handles the Correlation model from Chapter 4, the MULTI_LABEL target concept corresponds to the Bayesian network model from Chapter 6, and the DOUBLE_REGRESSION target concept concerns the simple Regression model from Section 7.4. For each target concept, a range of quality measures is available that allow the user to define exactly what sort of exceptional subgroups Cortana should search for. The subgroup validation method we develop in Chapter 8 is also available in Cortana.

Independent of the choice of target concept and quality measure, Cortana provides a parametrized search algorithm. Several search strategies are included (breadth-first, depth-first, best-first, beam, and cover-based beam

[70]), and the user can select one of many strategies for dealing with numeric attributes. Furthermore, conditions can be set on the minimal subgroup size, the minimal subgroup quality, the maximal description length, the maximal number of subgroups to present at the end of the algorithm, and the maximal amount of total time the algorithm spends on the task.

Apart from many more things, Cortana provides a parametrized version of the Beam Search algorithm for Top-q Exceptional Model Mining, as detailed in Algorithm 1. It is available online, at `http://datamining.liacs.nl/cortana.html`.

Chapter 4

Deviating Interactions – Correlation Model

An Exceptional Model Mining instance strives to find subgroups, for which a particular kind of interaction between multiple target attributes is unusual, when compared to that same interaction between the same attributes on the entire dataset. Possibly the simplest such interaction is the correlation model. In this correlation model, we consider two numeric targets, ℓ_1 and ℓ_2. Within this model class, we will refer to them as $x = \ell_1$ and $y = \ell_2$. We are interested in their linear association as measured by the correlation coefficient ρ, estimated by the sample correlation coefficient

$$\hat{r} = \frac{\sum \left(x^i - \bar{x}\right)\left(y^i - \bar{y}\right)}{\sqrt{\sum \left(x^i - \bar{x}\right)^2 \sum \left(y^i - \bar{y}\right)^2}}$$

where x^i denotes the i^{th} observation on x, and $\bar{x}$ denotes its mean. We let ρ^G and ρ^{G^C} denote the population coefficients of correlation for G and G^C, respectively, and let $\hat{r}^G$ and $\hat{r}^{G^C}$ denote their sample estimates.

4.1 Quality Measure $\varphi_{\mathbf{scd}}$

To find descriptions with a substantial coverage and deviating correlation coefficient, we develop a statistically-oriented quality measure, based on the test

$$H_0 : \rho^G = \rho^{G^C} \quad \text{against} \quad H_1 : \rho^G \neq \rho^{G^C}$$

Generally, the sampling distribution of $\hat{r}$ is unknown. If x and y follow a bivariate normal distribution, we can apply the Fisher z transformation

$$z' = \frac{1}{2} \ln \left(\frac{1+\hat{r}}{1-\hat{r}} \right)$$

The sampling distribution of z' is approximately normal [84]. Its standard error is given by

$$\frac{1}{\sqrt{\xi - 3}}$$

where ξ is the size of the sample. As a consequence

$$z^* = \frac{z' - z^{C\prime}}{\sqrt{\frac{1}{n-3} + \frac{1}{n^C-3}}}$$

approximately follows a standard normal distribution under H_0. Here z' and $z^{C\prime}$ are the z-scores obtained through the Fisher z transformation for G and G^C, respectively. If both n and n^C are greater than 25, then the normal approximation is quite accurate, and can safely be used to compute the p-values. As quality measure φ_{scd} (acronym for Significance of Correlation Difference) we take 1 minus the computed p-value. Because we have to introduce the normality assumption to be able to compute the p-values, φ_{scd} should be viewed as a heuristic measure. Transformation of the original data (for example, taking their logarithm) may make the normality assumption more reasonable.

4.2 Experiments

4.2.1 Datasets

The *Windsor Housing* dataset [2] concerns 546 houses that were sold in Windsor, Canada in the summer of 1987. The information for each house includes the two attributes of interest, $\ell_1 = x = lot_size$ and $\ell_2 = y = sales_price$. An additional 10 attributes are available as descriptive attributes, including the number of bedrooms and bathrooms and whether the house is located at a desirable location. The correlation between lot size and sale price is 0.536, which implies that a larger size of the lot

Table 4.1: Statistics concerning the datasets used in the Correlation model (this chapter), Classification model (Chapter 5), and alternative Regression model (Section 7.4) experiments. Here, N is the total number of records, k is the number of descriptive attributes, and m is the number of targets on which the model is fitted.

Dataset	Domain	N	k	m
Affymetrix	Bioinformatics	63	311	2
Windsor Housing	Residential property value	546	10	2

coincides with a higher sales price. The fitted regression function is $y = 34136 + 6.60 \cdot x$, showing that on average one extra square meter corresponds to a sales price increase of \$6.60.

The *Affymetrix* dataset comes from the domain of bioinformatics. In genetics, genes are organised in so-called *gene regulatory networks*. This means that the expression (its effective activity) of a gene may be influenced by the expression of other genes. Hence, if one gene is regulated by another, one can expect a linear correlation between the associated expression-levels. In many diseases, specifically cancer, this interaction between genes may be disturbed. The *Affymetrix* dataset shows the expression-levels of 313 genes as measured by an Affymetrix microarray, for 63 patients that suffer from a cancer known as neuroblastoma [64]. Additionally, the dataset contains clinical information about the patients, including age, sex, stage of the disease, etc. As targets, we consider the expressions of the two genes *ZHX3* ('Zinc fingers and homeoboxes 2') and *NAV3* ('Neuron navigator 3'), showing a slightly positive overall correlation of 0.218.

4.2.2 Experimental Results

On the *Windsor Housing* dataset, we run an experiment with φ_{scd}. As discussed in Section 4.1, in order to be confident about the test results for this quality measure, the coverage of a description has to be over 25. This number was used as minimum support threshold for a run of Cortana using φ_{scd}. The following description (and its complement) was found to show the most significant difference in correlation ($\varphi_{\text{scd}}(D_1) = 0.9993$)

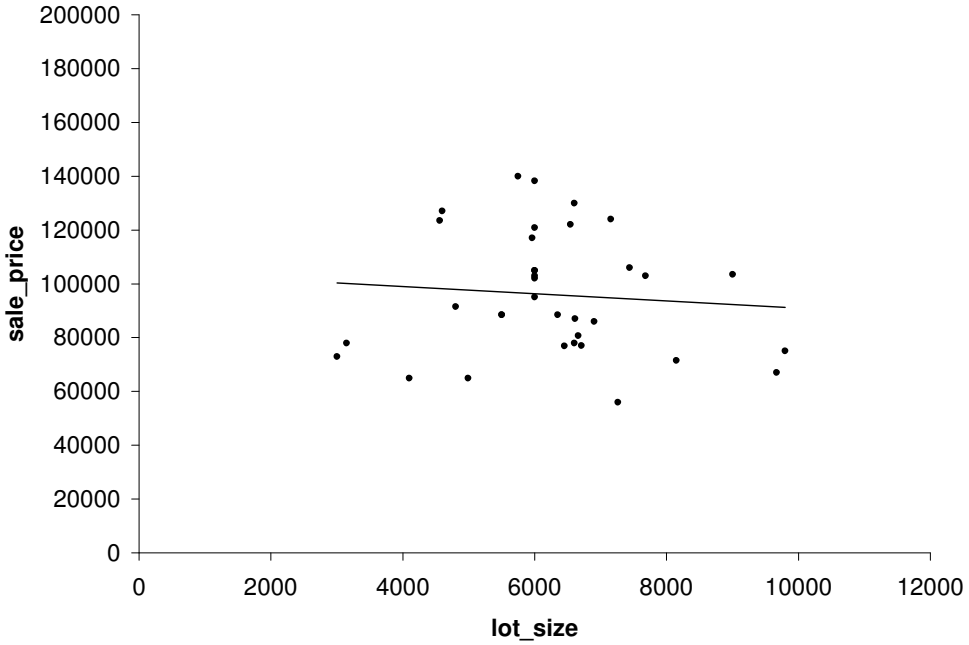
$$D_1 : drive = 1 \wedge rec_room = 1 \wedge nbath \geq 2$$

This is the group of 35 houses (covering 6.4% of the dataset) that have a driveway, a recreation room and at least two bathrooms. The scatter plots for the D_1 and D_1^C are given in Figure 4.1. The subgroup shows a correlation of $\hat{r}^{G_1} = -0.090$ compared to $\hat{r}^{G_1^C} = 0.549$ for the remaining 511 houses. A tentative interpretation could be that D_1 describes houses in the higher segments of the market where the price of a house is mostly determined by its location and facilities. The desirable location may provide a natural limit on the lot size, such that this is not a factor in the pricing. Figure 4.1 supports this hypothesis: houses in D_1 tend to have a higher price ($\$95,947$ on average, versus $\$68,122$ on the whole dataset).

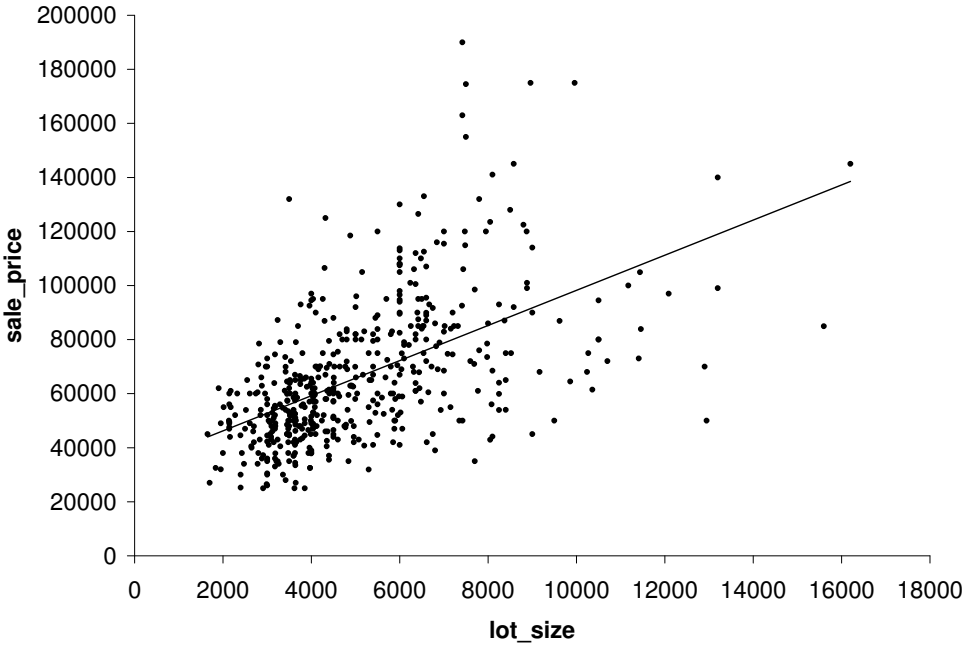
In general *sales_price* and *lot_size* are positively correlated, but EMM discovers a description with a slightly negative correlation. However, this value is not significantly different from zero: a test of

$$H_0 : \hat{r}^{G_1} = 0 \quad \text{against} \quad H_1 : \hat{r}^{G_1} \neq 0$$

yields a p-value of 0.61. The scatter plot confirms our impression that *sales_price* and *lot_size* are uncorrelated within the description. For purposes of interpretation, it is interesting to perform some post-processing. In Table 4.2 we give an overview of the correlations within different descriptions whose intersection produces the final result, as given in the last row. It is interesting to see that the condition $nbath \geq 2$ in itself actually leads to a slight increase in correlation compared to the whole database, but the combination with the presence of a recreation room leads to a substantial drop to $\hat{r} = 0.129$. When we add the condition that the house should also have a driveway we arrive at the final result with $\hat{r} = -0.090$. Note that adding this last condition only eliminates 3 records (the size of the subgroup goes from 38 to 35) and that the correlation between sales price and lot size in these three records (defined by the condition $nbath \geq 2 \wedge \neg drive = 1 \wedge rec_room = 1$) is -0.894. We witness a phenomenon similar to Simpson's paradox: splitting up a description with positive correlation (0.129) produces two descriptions both with a negative correlation (-0.090 and -0.894, respectively). This is a real-life occurrence of an effect similar to the one we witnessed in the artificial dataset of Figure 3.1, used in Chapter 3 for the sake of argument.



(a) G_1, with $\hat{r} = -0.090$.



(b) G_1^C, with $\hat{r} = 0.549$.

Figure 4.1: *Windsor Housing* - φ_{scd}: Scatter plot of *lot_size* and *sales_price* for the subgroup G_1 corresponding to description D_1 : $drive = 1 \wedge rec_room = 1 \wedge nbath \geq 2$ and its complement.

Table 4.2: Descriptions on the housing data, and their sample correlation coefficients and supports.

D	$\hat{r}^{G_D}$	$\lvert G_D \rvert$
Whole dataset	0.536	546
$nbath \geq 2$	0.564	144
$drive = 1$	0.502	469
$rec_room = 1$	0.375	97
$nbath \geq 2 \wedge drive = 1$	0.509	128
$nbath \geq 2 \wedge rec_room = 1$	0.129	38
$drive = 1 \wedge rec_room = 1$	0.304	90
$nbath \geq 2 \wedge rec_room = 1 \wedge \neg drive = 1$	−0.894	3
$nbath \geq 2 \wedge rec_room = 1 \wedge drive = 1$	−0.090	35

4.3 Alternatives

A logical consideration for a quality measure would be the absolute difference of the correlation for the description D and its complement, i.e.

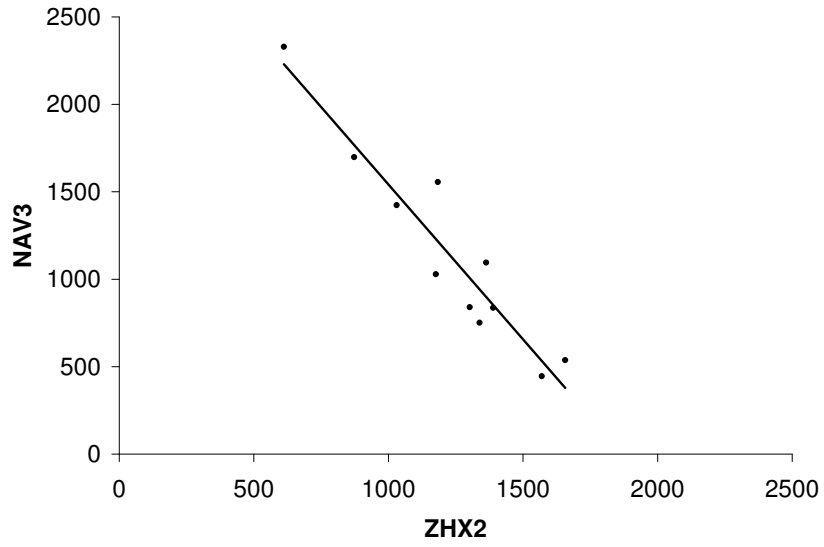
$$\varphi_{abs}(D) = \left| \hat{r}^{G_D} - \hat{r}^{G_D^C} \right|$$

Unfortunately, this measure does not take into account the coverage of the descriptions, and hence does not do anything to prevent overfitting.

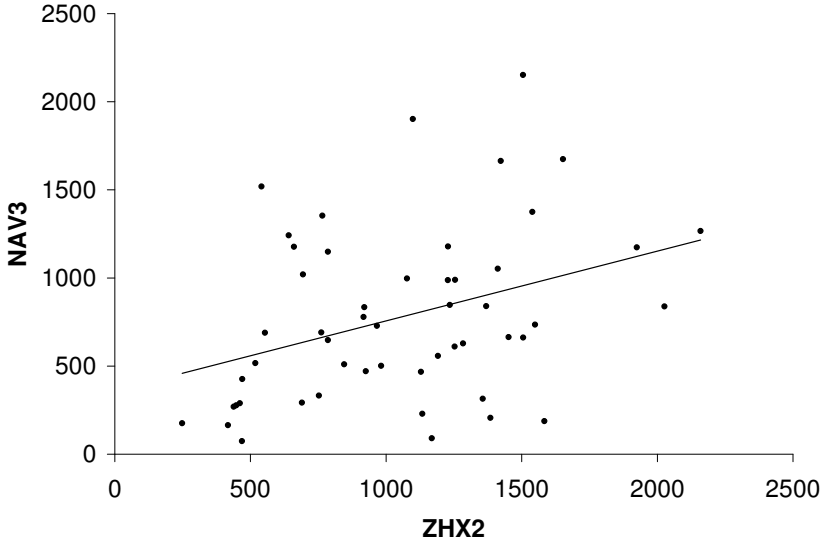
On the *Affymetrix* dataset, recall that we analyse the correlation between *ZHX3* and *NAV3*, showing a very slight correlation ($\hat{r} = 0.218$) on the whole dataset. We analyze this dataset in terms of the absolute difference of correlations φ_{abs}, allowing the use of all remaining attributes (both gene expression and clinical information) for building descriptions. As the φ_{abs} measure does not have any provisions for promoting larger subgroups, we use a minimum support threshold of 10 (15% of the patients). The largest distance ($\varphi_{abs}(D_2) = 1.313$) was found with the following description covering 11 records (17.5%) of the dataset

$D_2 : 11_band = \text{'no deletion'} \wedge survival\ time \leq 1919 \wedge XP_498569.1 \leq 57$

Figure 4.2 shows the plot for this description and its complement with the regression lines drawn in. The correlation for the description is $\hat{r}^{G_2} = -0.95$



(a) G_2, with $\hat{r} = -0.950$.



(b) G_2^C, with $\hat{r} = 0.363$.

Figure 4.2: *Affymetrix* - φ_{abs}: Scatter plot of the subgroup corresponding to description D_2 : *11_band = 'no deletion'* $\wedge$ *survival time* $\leq 1919 \wedge$ *XP_498569.1* ≤ 57 and its complement.

and the correlation in the remaining data is $\hat{r}^{G_2^C} = 0.363$. Note that the description displays a very "stable" behavior: all points are quite close to the regression line, with $R^2 \approx 0.9$.

As an improvement of φ_{abs}, the following quality function weighs the absolute difference between the correlations with the *entropy function* of the split between the description and its complement, as introduced in Section 3.2.1. Hence, when we find descriptions with φ_{abs}, but we find their coverage not substantial enough, we can solve this problem by running EMM with the alternative quality measure φ_{ent}, defined as

$$\varphi_{ent}(D) = \varphi_{ef}(D) \cdot \left| \hat{r}^{G} - \hat{r}^{G^C} \right|$$

4.4 Conclusions

In this chapter, we propose to use the correlation between two numeric targets as a measure of exceptionality for descriptions. This is probably the simplest form of target interplay for which Exceptional Model Mining can find deviating descriptions. As such, a domain expert should be able to easily interpret not only a found description, but also the associated model. As we have seen, particularly in discussing description D_1 found on the *Windsor Housing* dataset, a rationale for a subgroup can relatively easily be given based on the domain-specific interpretation of attributes on which the description is defined. This rationale can be fortified straightforwardly by inspecting the corresponding sample correlation coefficients. The statistical test, yielding the impression that the targets are uncorrelated within D_1, gives us confidence that the rationale makes sense. Also, a domain expert could learn a lot from observations such as the Simpson's paradox observed in Table 4.2. Thus, having only one parameter of interest in gauging the interesting interplay between targets, even though it restricts the EMM framework to relatively simple models, can enhance the analysis of the experimental results.

Chapter 5

Deviating Predictive Performance – Classification Model

As a more complex Exceptional Model Mining (EMM) instance, we turn to a classification model. We strive to find subgroups of the dataset for which the performance of a classifier is way off target, or particularly spot-on. In classification models, the output target attribute $y = \ell_m$ is discrete. Generally speaking, the other targets $\ell_1, \ldots, \ell_{m-1}$ can be of any type (binary, nominal, numeric, etc.), though a particular choice of classifier may restrict this. Our EMM framework allows for any classification method, as long as some quality measure can be defined in order to judge the models induced. Although we allow arbitrarily complex methods, such as decision trees, support vector machines or even ensembles of classifiers, we only consider a relatively simple classifier here, for reasons of simplicity and efficiency: we consider the logistic regression model

$$\text{logit}(P(y_i = 1|x_i)) = \ln\left(\frac{P(y_i = 1|x_i)}{P(y_i = 0|x_i)}\right) = \beta_0 + \beta_1 \cdot x_i$$

where $y \in \{0, 1\}$ is a binary class label and $x \in \{\ell_1, \ldots, \ell_{m-1}\}$. The coefficient β_1 tells us something about the effect of x on the probability that y occurs, and hence may be of interest to domain experts. A positive value for β_1 indicates that an increase in x leads to an increase of $P(y = 1|x)$. The strength of influence can be quantified in terms of the change in the odds of $y = 1$ when x increases with, say, one unit.

5.1 Quality Measure φ_{sed}

To judge whether the effect of x is substantially different in a particular subgroup G_D (built from a description D), we fit the model

$$\text{logit}(P(y_i = 1|x_i)) = \beta_0 + \beta_1 \cdot D(i) + \beta_2 \cdot x_i + \beta_3 \cdot (D(i) \cdot x_i) \qquad (5.1)$$

where $D(i)$ is shorthand for $D\left(a_1^i, \ldots, a_k^i\right)$ (cf. Section 2.1). Note that

$$\text{logit}(P(y_i = 1|x_i)) = \begin{cases} (\beta_0 + \beta_1) + (\beta_2 + \beta_3) \cdot x_i & \text{if } D(i) = 1 \\ \beta_0 + \beta_2 \cdot x_i & \text{if } D(i) = 0 \end{cases}$$

Hence, we allow both the slope and the intercept to be different for the description and its complement. As a quality measure, we propose to use one minus the p-value of a test on $\beta_3 = 0$ against a two-sided alternative in the model of Equation (5.1). This is a standard test in the literature on logistic regression [84]. We refer to this quality measure as φ_{sed}, an acronym whose meaning is lost in time, but maintained in order to correspond to the acronym in the original paper [71].

5.2 Experiments

5.2.1 Datasets

We demonstrate the classification model on the *Affymetrix* dataset, which was also used in the correlation model experiments. For more details on the dataset, see Section 4.2.1.

5.2.2 Experimental Results

In the logistic regression experiment, we take *NBstatus* as the output $\ell_2 = y$, and *age* (age at diagnosis in days) as the predictor $\ell_1 = x$. The descriptions are created using the gene expression level variables. Hence, the model specification is

$$\text{logit}\{\ P(\ y_i = \textit{`relapse or deceased'} \mid x_i\ )\ \}$$
$$=$$
$$\beta_0 + \beta_1 \cdot D(i) + \beta_2 \cdot x_i + \beta_3 \cdot (D(i) \cdot x_i)$$

We find the description

$$D_3 : SMPD1 \geq 840 \wedge HOXB6 \leq 370.75$$

with a coverage of 33 (52.4%), and quality $\varphi_{sed}(D_3) = 0.994$. We find a positive coefficient of x for the description, and a slightly negative coefficient for its complement. Within the description, the odds of *NBstatus = 'relapse or deceased'* increase with 44% when the age at diagnosis increases with 100 days, whereas in the complement the odds decrease with 8%. Within the description, an increase in age at diagnosis decreases the probability of survival, whereas within the complement, an increase in age slightly increases the probability of survival. Such reversals of the direction of influence may be of particular interest to the domain expert.

5.3 Alternatives

Another classifier we can consider is the *Decision Table Majority* (DTM) classifier [58, 62], also known as a *simple decision table*. The idea behind this classifier is to compute the relative frequencies of the ℓ_m values for each possible combination of values for $\ell_1, \ldots, \ell_{m-1}$. For combinations that do not appear in the dataset, the relative frequency estimates are based on that of the whole dataset. The predicted ℓ_m value for a new individual is simply the one with the highest probability estimate for the given combination of input values.

Example 1. *As an example of a DTM classifier, consider a hypothetical dataset of* 100 *people applying for a mortgage. The dataset contains two attributes describing the age (divided into three suitable categories) and marital status of the applicant. A third attribute indicates whether the application was successful, and is used as the output. Out of the* 100 *applications,* 61 *were successful. The following decision table lists the estimated probabilities of success for each combination of age and married. The support for each combination is indicated between brackets.*

	married = 'no'	*married = 'yes'*
age = 'low'	0.25 (20)	0.61 (0)
age = 'medium'	0.4 (15)	0.686 (35)
age = 'high'	0.733 (15)	1.0 (15)

As this table shows, the combination married = 'yes' ∧ age = 'low' does not appear in this particular dataset, and hence the probability estimate is based on the complete dataset (0.61). This classifier predicts a positive outcome in all cases except when married = 'no' and age is either 'low' or 'medium'.

For this instance of the classification model we discuss two different quality measures. The *Bayesian Dirichlet equivalent uniform* (BDeu) score, which can be used as a measure for the performance of the DTM classifier on G_D, and the *Hellinger distance*, which assigns a value to the distance between the conditional probabilities estimated on G_D and G_D^C.

5.3.1 BDeu Score (φ_{BDeu})

The BDeu score φ_{BDeu} is a measure from Bayesian theory [48] and is used to estimate the performance of a classifier for a description, with a penalty for small contingencies that may lead to overfitting. Note that this measure ignores how the classifier performs on the complement. It merely captures how "predictable" a particular description is.

The BDeu score is defined as

$$\prod_{\ell_1,\ldots,\ell_{m-1}} \frac{\Gamma(\theta/\mathcal{I})}{\Gamma(\theta/\mathcal{I} + \#(\ell_1,\ldots,\ell_{m-1}))} \prod_{\ell_m} \frac{\Gamma(\theta/\mathcal{I}\mathcal{J} + \#(\ell_1,\ldots,\ell_m))}{\Gamma(\theta/\mathcal{I}\mathcal{J})}$$

where Γ denotes the gamma function, $\mathcal{I}$ denotes the number of value combinations of the input variables, $\mathcal{J}$ the number of values of the output variable, and $\#(\ell_1,\ldots,\ell_m)$ denotes the number of records with that value combination. The parameter θ denotes the *equivalent sample size*. Its value can be chosen by the user.

5.3.2 Hellinger (φ_{Hel})

Another possibility is to use the Hellinger distance [115]. It defines the distance between two probability distributions $P(x)$ and $Q(x)$ as follows

$$H(P,Q) = \sum_x \left(\sqrt{P(x)} - \sqrt{Q(x)}\right)^2$$

where the sum is taken over all possible values x. In our case, the distributions of interest are

$$P(\ell_m \mid \ell_1, \ldots, \ell_{m-1})$$

for each possible value combination $\ell_1, \ldots, \ell_{m-1}$. The overall distance measure becomes

$$\varphi_{\text{Hel}}(D) = H\left(\hat{P}^{G_D}, \hat{P}^{G_D^C}\right) =$$

$$\sum_{\ell_1,\ldots,\ell_{m-1}} \sum_{\ell_m} \left(\sqrt{\hat{P}^{G_D}(\ell_m|\ell_1,\ldots,\ell_{m-1})} - \sqrt{\hat{P}^{G_D^C}(\ell_m|\ell_1,\ldots,\ell_{m-1})}\right)^2$$

where $\hat{P}^{G_D}$ denotes the probability estimates on G_D. Intuitively, we measure the distance between the conditional distribution of ℓ_m in G_D and G_D^C for each possible combination of input values, and add these distances to obtain an overall distance.

5.3.3 Experimental Results

For the DTM classification experiments on the *Affymetrix* dataset, we have selected three binary attributes. The first two attributes, which serve as input variables of the decision table, are related to genomic alterations that may be observed within the tumor tissues. The attribute *1p_band* (ℓ_1) describes whether the small arm ('p') of the first chromosome has been deleted. The second attribute, *mycn* (ℓ_2), describes whether one specific gene is amplified or not (multiple copies introduced in the genome). Both attributes are known to potentially influence the genesis and prognosis of neuroblastoma. The output attibute for the classification model is *NBstatus* (ℓ_3), which can be either *'no event'* or *'relapse or deceased'*. The following decision table describes the conditional distribution of *NBstatus* given *1p_band* and *mycn* on the whole dataset

1p_band =	*mycn* = *'amplified'*	*mycn* = *'not amplified'*
'deletion'	0.333 (3)	0.667 (3)
'no change'	0.625 (8)	0.204 (49)

In order to find descriptions for which the distribution is significantly different, we run EMM with the Hellinger distance φ_{Hel} as quality measure. As our quality measures for classification do not specifically promote descriptions with larger coverage, we have selected a slightly higher minimum support constraint: $minsup = 16$, which corresponds to 25% of the data. The following subgroup of 17 patients (27.0%) was the best found ($\varphi_{\text{Hel}}(D_4) = 3.803$)

$$D_4 : prognosis = \text{'unknown'}$$

1p_band =	*mycn* = *'amplified'*	*mycn* = *'not amplified'*
'deletion'	1.0 (1)	0.833 (6)
'no change'	1.0 (1)	0.333 (9)

Note that for each combination of input values, the probability of *'relapse or deceased'* is increased, which makes sense when the prognosis is uncertain. Note furthermore that the overall dataset does not yield a pure classifier: for every combination of input values, there is still some confusion in the predictions.

In our second alternative classification experiment, we are interested in "predictable" descriptions. Therefore, we run EMM with the φ_{BDeu} measure. All other settings are kept the same. The following subgroup ($|G_5| = 16$ (25.4%), $\varphi_{\text{BDeu}}(D_5) = -1.075$) is based on the expression of the gene *RIF1* ('RAP1 interacting factor homolog (yeast)')

$$D_5 : RIF1 \geq 160.45$$

1p_band =	*mycn* = *'amplified'*	*mycn* = *'not amplified'*
'deletion'	0.0 (0)	0.0 (0)
'no change'	0.0 (0)	0.0 (16)

For this description, the predictiveness is optimal, as all patients turn out to be tumor-free. In fact, the decision table ends up being rather trivial, as all cells indicate the same decision.

5.4 Conclusions

In this chapter, we propose to find descriptions for which a classifier learned from the targets has deviating performance. In theory, this can be done with any classification algorithm, which can be as complex as one desires. In practice, we have developed statistically and probabilistically inspired quality measures for a few relatively simple classification algorithms: logistic regression with merely one predictor, and a multi-predictor Decision Table Majority classifier.

As we have seen in our analysis of description D_3, similarly to some of the findings in the previous chapter, the classification model allows for extensive model inspection. The description merely indicates that a combination of expression level constraints corresponds to deviating behavior. From further model analysis, however, we can learn that the value of one of the predictors has a positive influence on the output value *within* D_3, while the influence is negative *outside of* D_3. This Simpson's paradox is invaluable knowledge for a domain expert.

Specific interest in the resulting subgroups on the dataset domain aside, EMM with this particular model class is potentially extremely interesting within our own field of study, by delivering meta learning information. When data miners are working with, or developing their own, classification algorithms, this instance of EMM can deliver important indications when the algorithm works particularly well, and when it performs not so well. Classification algorithm developers can incorporate this knowledge to improve their algorithms. For classification algorithm users, particularly descriptions such as D_5 found with φ_{BDeu} are potentially interesting. This measure aims to find predictable descriptions. The resulting decision table shows that description D_5 highlights a part of the dataset where predictiveness is optimal: the corresponding subspace of the total search space has essentially been solved. Considering this part of the problem to be solved, we can then focus our attention on classifying the rest of the input space, making the hypothesis space smaller and potentially reducing the computational burden of subsequent classifier runs.

Chapter 6

Unusual Conditional Interactions – Bayesian Network Model

In Chapter 4, we discussed an EMM instance with an internally unsupervised model class, regarding the correlation between two attributes. In Chapter 5, we discussed an EMM instance with an internally supervised model class, classifying a single output target attribute based on one or several input target attributes. Depending on the choice of classifier, this may or may not incorporate complex interactions between sets of input target attributes; in any case, such complex interactions have not yet been considered for an unsupervised model class. In this chapter we fill that void, by considering the Exceptional Model Mining instance with a Bayesian network as model class.

In the Bayesian network model class we allow multiple nominal targets $\ell_1, \ldots, \ell_m$. A description is deemed interesting, when the conditional dependence relations between the targets are substantially different for the description from these relations on the whole dataset. Hence we validate the descriptions on the conditional interdependencies between the targets, rather than the target values themselves. To capture these interdependencies, we learn a Bayesian network between the targets, from data.

The choice to capture complex interactions between larger sets of unsupervised target attributes by means of conditional dependence relations, is inspired by the *Pisaster* example discussed in Chapter 2. Recall that the field study of Robert T. Paine [86] yields, among many other results, that a

conditional dependence relation exists between the sponge *Haliclona*, the nudibranch *Anisodoris*, and the starfish *Pisaster ochraceus*. This study gives a real-life example of multiple-target interactions that require the complexity of a Bayesian network.

There are many algorithms to learn a Directed Acyclic Graph (DAG) model, such as a Bayesian network, from data; see for instance [8, 47, 67]. We use a non-deterministic hill climbing algorithm; using a hill climbing method makes the algorithm speedy enough for use in an EMM setting, while its non-deterministic nature decreases the chance that the algorithm will end up in a local optimum.

We start with a Bayesian network with m vertices and no edges, and compute the quality of that model. We choose the Bayesian Dirichlet equivalent uniform (BDeu) score (see Section 5.3.1), because it assigns equal scores to equivalent models and assumes no prior information. Then we hill-climb through the space of Bayesian networks by applying the best single-edge change in the model. At each step, we apply a random number of covered arc reversals [12], in order to escape from a maximum that may be local. For more details on this combination of methods, see [95].

Notice that this process is non-deterministic: at every step in the hill climbing, and whenever we try to escape a maximum, a random number of randomly selected covered edges is reversed. During our experiments we occasionally find different Bayesian networks for the same data with different random seeds. However, these variations were modest: few edges change, and resulting networks for the same data are usually equivalent.

We consider the choice of method to learn a Bayesian network from data a parameter of this EMM instance.

6.1 Quality Measure φ_{weed}

Having chosen a method to learn a Bayesian network from data, we would like to employ such networks to capture deviating conditional dependence relations between targets. Our quality measure uses the structure of the learned networks to this end. The main idea is to start the EMM process by learning a Bayesian network BN^{Ω} between the targets from the entire

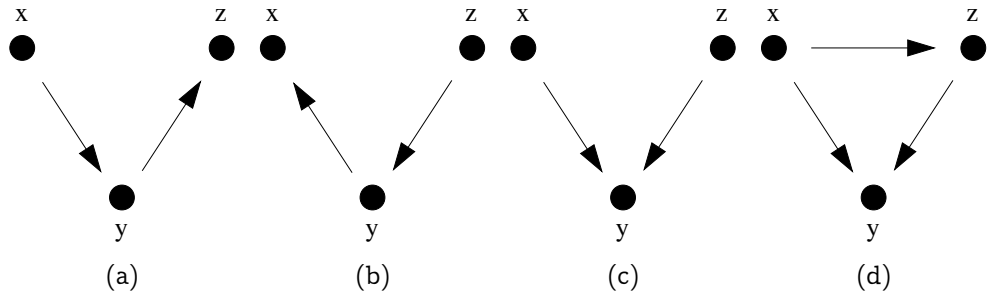


Figure 6.1: Example Bayesian networks.

dataset. Then, for each description D under consideration, we learn another Bayesian network BN^D, but we learn it *only from the records covered by* D. Comparing the structure of the networks BN^Ω and BN^D then gives us a measure for the quality of the description D. One might be tempted to consider traditional edit distance between graphs to make this comparison, but then we would not take into account some peculiarities about how Bayesian networks represent independence relations.

6.1.1 Independence Relations in Bayesian Networks

There are two important peculiarities about the independence relations in Bayesian networks, which we illustrate by the example networks in Figure 6.1. First, seemingly different Bayesian networks may represent the same independence relations. If we look at network (b), we find that in this network only one independence relation holds: x and z are conditionally independent given y. By symmetry of conditional independence, this is the same independence relation as the one in network (a). Bayesian networks that represent the same independence relations are called *equivalent*. Note that this relation partitions Bayesian networks into equivalence classes. Second, Bayesian networks with the same skeleton (the network when we drop the directions) are not necessarily equivalent. In network (c), x and z are marginally independent, unlike in networks (a) and (b).

We identify a special configuration of vertices and edges in a Bayesian network that is relevant for the discussion in the rest of this chapter. It is a structure as seen in network (c): a *v-structure*.

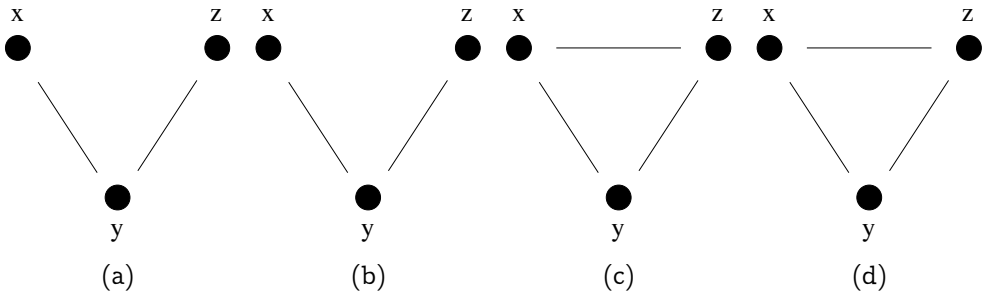


Figure 6.2: Moralized graphs for the networks in Figure 6.1.

Definition (V-structure). A *v-structure* in a Bayesian network is a set of three vertices $\{x, y, z\}$ such that the network contains edges $x \rightarrow y$ and $z \rightarrow y$, but no edge between x and z.

The probabilistic interpretation of this v-structure is that x and z are marginally independent, but conditionally dependent given y. A v-structure is also known as an *immorality*, since the parents of vertex y are 'unmarried', i.e. there is no edge between them. A graph can be *moralized* [17] by first marrying all unmarried parents (i.e. draw an edge between all pairs of vertices that have a common child but no common edge), and then dropping directions. Thus, moralizing a graph removes all v-structures. The moralized versions of the networks of Figure 6.1 are depicted in Figure 6.2. As one can see, the moralized version of network (c) has an extra edge, which corresponds to removing the v-structure from the original network.

Notice that the moral graph also is not sufficient to capture all information about the underlying independence relations; x and z are marginally independent in network (c) and marginally dependent in network (d), but these networks have the same moral graph.

6.1.2 Edit Distance for Bayesian Networks

To overcome the peculiarities of Bayesian networks, we propose a heuristic quality measure based on the following well-known result by Verma and Pearl [111]

Theorem 2 (Equivalent DAGs). *Two DAGs are equivalent if and only if they have the same skeleton and the same v-structures.*

Since these two conditions determine whether two DAGs are equivalent, it makes sense to consider the number of potential edges violating the conditions as a measure of how different two DAGs are.

Definition (Edit distance for Bayesian networks). Let two Bayesian networks BN^1 and BN^2 be given with the same set of vertices. Denote the edge set of their skeletons by S^1 and S^2, and the edge set of their moralized graphs by M^1 and M^2. Let

$$\zeta = \left| \left[S^1 \ominus S^2\right] \cup \left[M^1 \ominus M^2\right] \right|$$

The distance between BN^1 and BN^2 is defined as

$$\delta(BN^1, BN^2) = \frac{2\zeta}{m(m-1)}$$

As usual in set theory, $\ominus$ denotes a symmetric difference: $X \ominus Y = (X \cup Y) - (X \cap Y)$. The factor $\frac{2}{m(m-1)}$ causes the distance to range between 0 and 1: it is the expanded reciprocal of $\binom{m}{2}$, the number of distinct pairs of targets in the dataset, hence vertices in the Bayesian networks.

We illustrate the edit distance by computing the mutual distances between the networks in Figure 6.1. We find that $\delta(a, b) = 0$ and $\delta(a, c) = \delta(a, d) = \delta(b, c) = \delta(b, d) = \delta(c, d) = 1/3$. Only for the two networks that are equivalent, distance 0 is obtained. If we compare the networks to the independence model $\varnothing$ which has no edges at all, we obtain $\delta(a, \varnothing) = \delta(b, \varnothing) = 2/3$, and $\delta(c, \varnothing) = \delta(d, \varnothing) = 1$.

The edit distance can now be used to quantify the exceptionality of a description

Definition (Edit distance based quality measure). Let a description D be given. Denote the Bayesian network we learn from Ω by BN^{Ω}, and denote the Bayesian network we learn from G_D by BN^D. Then the quality of D is

$$\varphi_{ed}(D) = \delta\left(BN^{\Omega}, BN^D\right)$$

If we would plug φ_{ed} into the EMM framework, a familiar problem would occur: unusual interdependencies between the targets are easily achieved in very small subsets of the dataset. Thus, using φ_{ed} would result in small subgroups. For this reason, we combine the measure with the entropy function φ_{ef} (cf. Section 3.2), to obtain the following aggregate measure.

Definition (Weighed Entropy and Edit Distance).

$$\varphi_{\text{weed}}(D) = \sqrt{\varphi_{\text{ef}}(D)} \cdot \varphi_{\text{ed}}(D)$$

The original components ranged from 0 to 1, hence the new quality measure does so too. We take the square root of the entropy, thus reducing its bias towards 50/50 splits, since we are primarily interested in a description with large edit distance, while mediocre entropy is acceptable.

6.2 Experiments

6.2.1 Datasets

The *Emotions* dataset [103] consists of 593 songs, from which 8 rhythmic and 64 timbre features were extracted. Domain experts assigned the songs to any number of six main emotional clusters from the Tellegen-Watson-Clark model of mood [102]: *'amazed-surprised'*, *'happy-pleased'*, *'relaxing-calm'*, *'quiet-still'*, *'sad-lonely'*, and *'angry-fearful'*.

The *Scene* dataset [6] is from the semantic scene classification domain, in which a photo can be classified into one or more of 6 classes. It contains 2407 photos, each of which is divided into 49 blocks using a 7×7 grid. For each block the first two spatial color moments of each band of the LUV color space are computed. This space identifies a color by its lightness (the L* band) and two chromatic valences (the u* and v* band). The photos can have the classes *'beach'*, *'field'*, *'fall foliage'*, *'mountain'*, *'sunset'*, and *'urban'*.

From the biological field we consider the *Yeast* dataset [28]. It consists of micro-array expression data and phylogenetic profiles with 2417 genes of the yeast *Saccharomyces cerevisiae*. Each gene is annotated with any number of 14 functional classes.

Table 6.1: Statistics concerning the datasets used in the Bayesian Network Model and Multi-label LeGo experiments (cf. Chapter 9). Here, N is the total number of records, k is the number of descriptive attributes, and m is the number of nodes in the fitted Bayesian network model. The column *Cardinality* displays the average number of positive targets per record.

Dataset	Domain	N	k	m	Cardinality
Emotions	Music	593	72	6	1.87
Mammals	Zoogeography	2221	69	101	24.43
Scene	Vision	2407	294	6	1.07
Yeast	Biology	2417	103	14	4.24

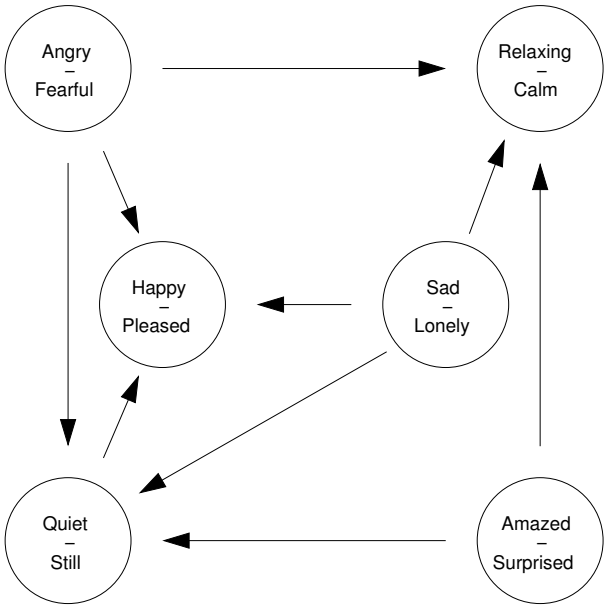
The three introduced datasets all have a relatively small number of targets. Hence the fitted Bayesian networks are easy to interpret, and experiments on these datasets form a nice proof of concept for our method. However, EMM with the Bayesian Network model class can also handle larger, more complex target systems. Hence, in addition to the MLC datasets, we analyse the *Mammals* dataset [40, 80]. It focuses on subdividing the geography of Europe into clusters based on their fauna, which is a core activity of biology. The dataset was created by combining two datasets: one documenting presence or absence of 101 mammals for a set of 2221 grid cells covering Europe, and one documenting climate and elevation of the corresponding land areas. We define candidate subgroups by conditions on the climate and elevation data, and fit Bayesian networks on the mammals. We use a version of this dataset that was pre-processed by Heikinheimo et al. [49].

Some statistics regarding these datasets can be found in Table 6.1.

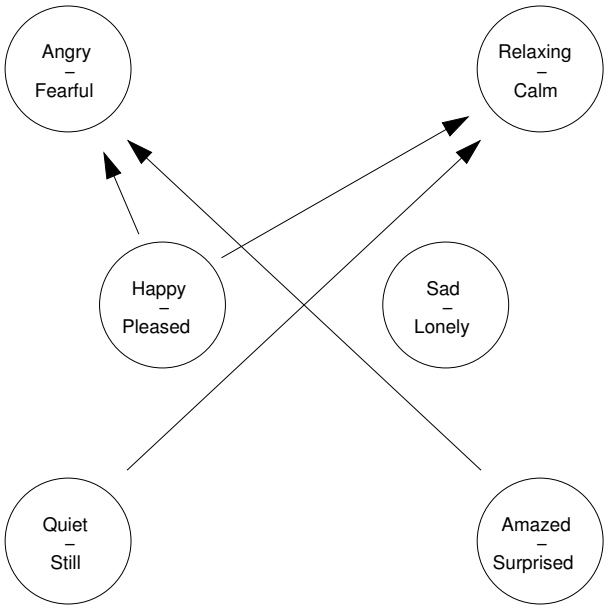
6.2.2 Experimental Results

Emotions Data

On the *Emotions* dataset, we obtained the networks shown in Figure 6.3. Figure 6.3a depicts a network learned from the whole dataset, and Figure 6.3b displays a network learned from a subgroup of size 94 (15.9%) corresponding to description $D_6 : STD_MFCC_7 \leq 0.203 \wedge Mean_Centroid \geq 0.066$, with quality $\varphi_{\text{weed}}(D_6) = 0.675$. The first condition says that coef-



(a) Whole dataset.



(b) D_6 : $STD_MFCC_7 \leq 0.203 \wedge Mean_Centroid \geq 0.066$.

Figure 6.3: Bayesian networks for the *Emotions* data.

ficient 7 of the 13-band Mel Frequency Cepstrum has a low standard deviation, which has a nontrivial interpretation. The second condition says that the songs in the subgroup have a moderate to high mean spectral centroid. This correlates with the impression of a bright sound [96].

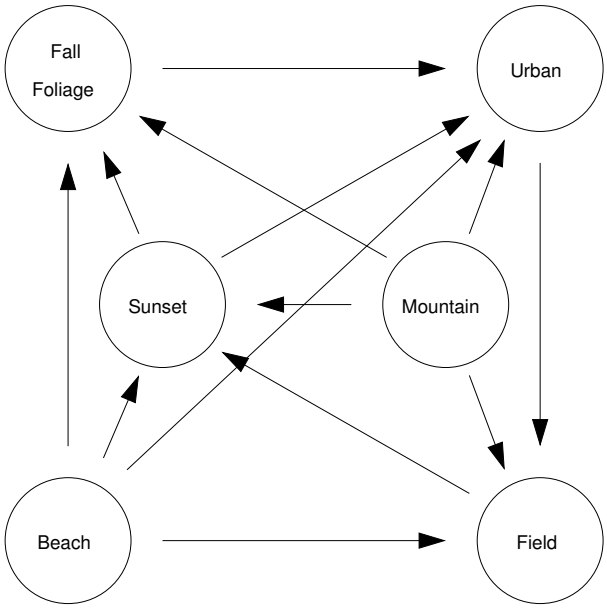
From Figure 6.3a we find that on the whole dataset, the emotion *sad-lonely* is correlated with all other emotions: it shares marginal dependence relations with *happy-pleased*, *relaxing-calm* and *quiet-still*, and conditional dependence relations given both *relaxing-calm* and *quiet-still* with *angry-fearful* and *amazed-surprised*. When restricted to the description, *sad-lonely* is correlated with none of the other emotions (cf. Figure 6.3b). This seems reasonable: we would expect that bright sounds in music have a great influence on whether humans perceive a song as *sad-lonely* or not. Hence for songs with bright sounds it is more likely that *sad-lonely* is less correlated with other factors (such as the other emotions); we already have an explanation for the distribution of *sad-lonely*, so the probability increases that it does not depend on the other emotions.

Scene Data

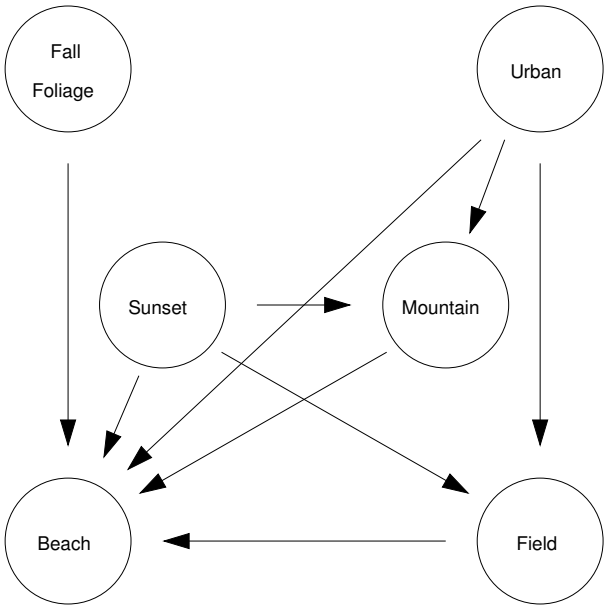
Figure 6.4 shows the networks fitted on the *Scene* dataset. In this dataset, we found a description with quality $\varphi_{\text{weed}}(D_7) = 0.545$, covering 452 records (18.8%). The conditions indicate a high mean lightness in the upper right corner of the photo, and a low mean u* chromatic valence in a more centrally located area.

Yeast Data

The first-ranked description on the *Yeast* dataset has quality $\varphi_{\text{weed}}(D_8) = 0.437$, and is defined by conditions on its 79-element gene expression data: *probe* $3 \leq -0.025 \wedge$ *probe* $66 \geq -0.071$. The three subsequent descriptions in the ranking each share their first condition with the top-ranked descriptions, hence they are not that interesting to present here. The fifth-ranked description has quality $\varphi_{\text{weed}}(D_9) = 0.369$ and conditions *probe* $9 \leq -0.063 \wedge$ *probe* $53 \geq -0.081$. The subgroup sizes are $|G_8| = 681$ (28.2%) and $|G_9| = 530$ (21.9%).



(a) Whole dataset.



(b) D_7 : *Mean L* band block 7* $\geq 0.699 \wedge$ *Mean u* band block 19* ≤ 0.336.

Figure 6.4: Bayesian networks for the *Scene* data.

From the fitted Bayesian networks, many changes in dependence relations can be deduced; we will outline a few. In G_8 the functional class *cell growth, cell division, DNA synthesis* has four dependence relations less than on the whole dataset, and *protein destination* has five less. On the other hand, *energy* and *ionic homeostasis* both have an extra dependence relation. In G_9, the functional classes *cellular organization* and *cell rescue, defence, death and aging* have fewer dependence relations than on the whole dataset (six and three, respectively), while *metabolism* and *cellular biogenesis* have one more.

Mammals Data

On the *Mammals* dataset, the first-ranked description D_{10} is defined by conditions $latitude \geq 49.85 \wedge prec_feb \geq 28.75$, i.e. northern areas with a fair amount of precipitation in February. Two other interesting descriptions (ranked sixth and eighth) are defined by meteorological conditions only. In description D_{11} we have $max_temp_nov \leq 7.66 \wedge prec_feb \leq 45.38$, i.e. November is not warm and precipitation in February is low, while in description D_{12} we have $max_temp_mar \leq 7.97 \wedge max_temp_sep \leq 17.65$, i.e. the temperatures in both March and September do not reach high levels. The descriptions have quality $\varphi_{\text{weed}}(D_{10}) = 0.122$, $\varphi_{\text{weed}}(D_{11}) = 0.121 = \varphi_{\text{weed}}(D_{12})$, and coverage $|G_{10}| = 839$ (37.8%), $|G_{11}| = 835$ (37.6%), and $|G_{12}| = 834$ (37.6%).

The Figures 6.5, 6.6, and 6.7 chart the regions in Europe that belong to the descriptions. Areas that are unique to one description within this set are Ireland and the Benelux for D_{10} (which had the condition that it is wet in February), Romania and Poland for D_{11} (cold in November, dry in February), and the Alps and Pyrenees for D_{12} (cold in both March and September).

Among the relations between mammals that distinguish the descriptions from each other and the whole dataset Ω are the following: the European Water Vole (*Arvicola terrestris*) and the Mountain Hare (*Lepus timidus*) are conditionally dependent given the Ermelin (*Mustela erminea*) on Ω but not on any of the descriptions, only on D_{10} the Wildcat (*Felis silvestris*) and the Beech Marten (*Martes foina*) are conditionally depen-

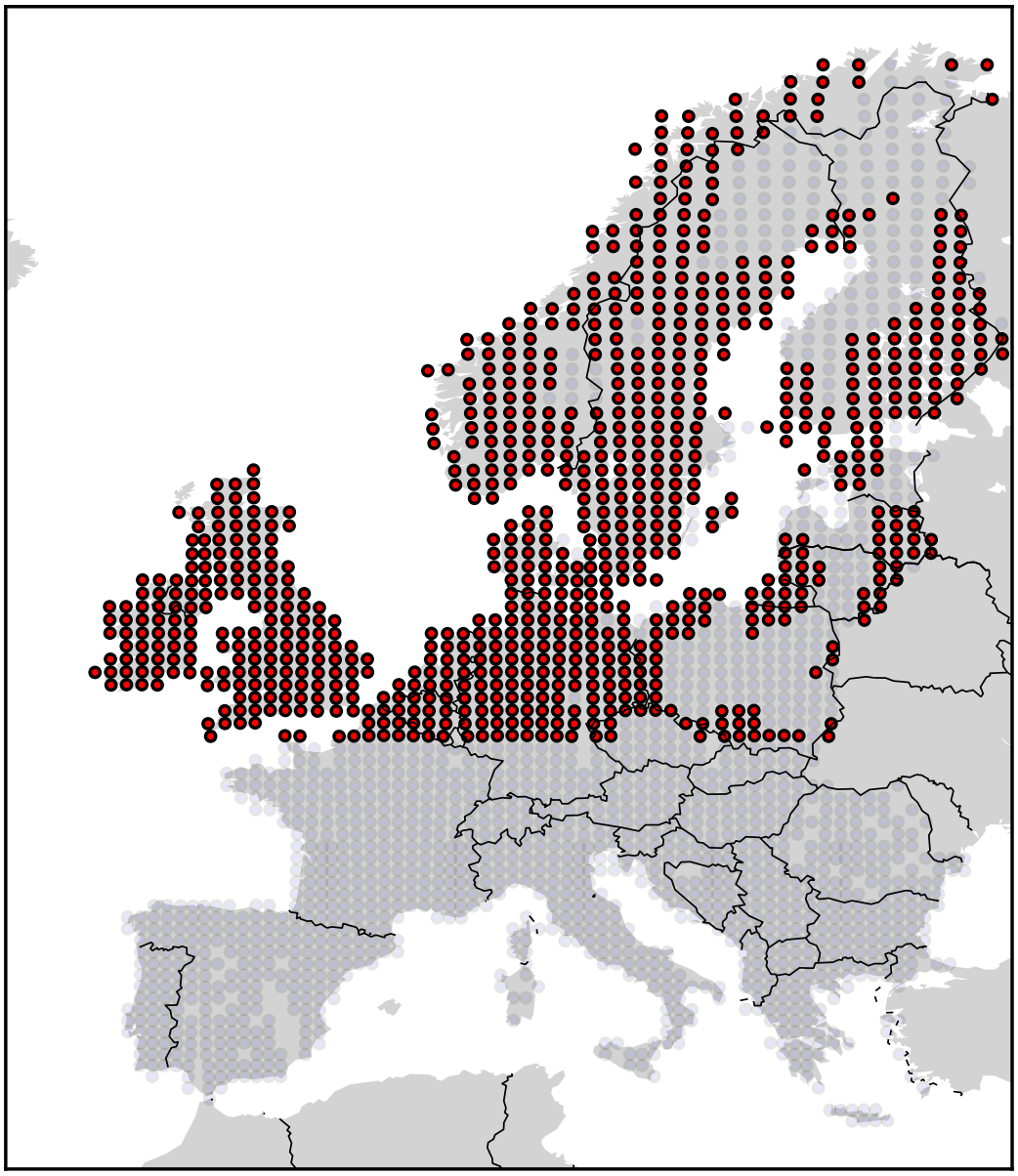


Figure 6.5: Regions in Europe that belong to the subgroup corresponding to D_{10} : $latitude \geq 49.85 \wedge prec_feb \geq 28.75$ ($|G_{10}| = 839$).

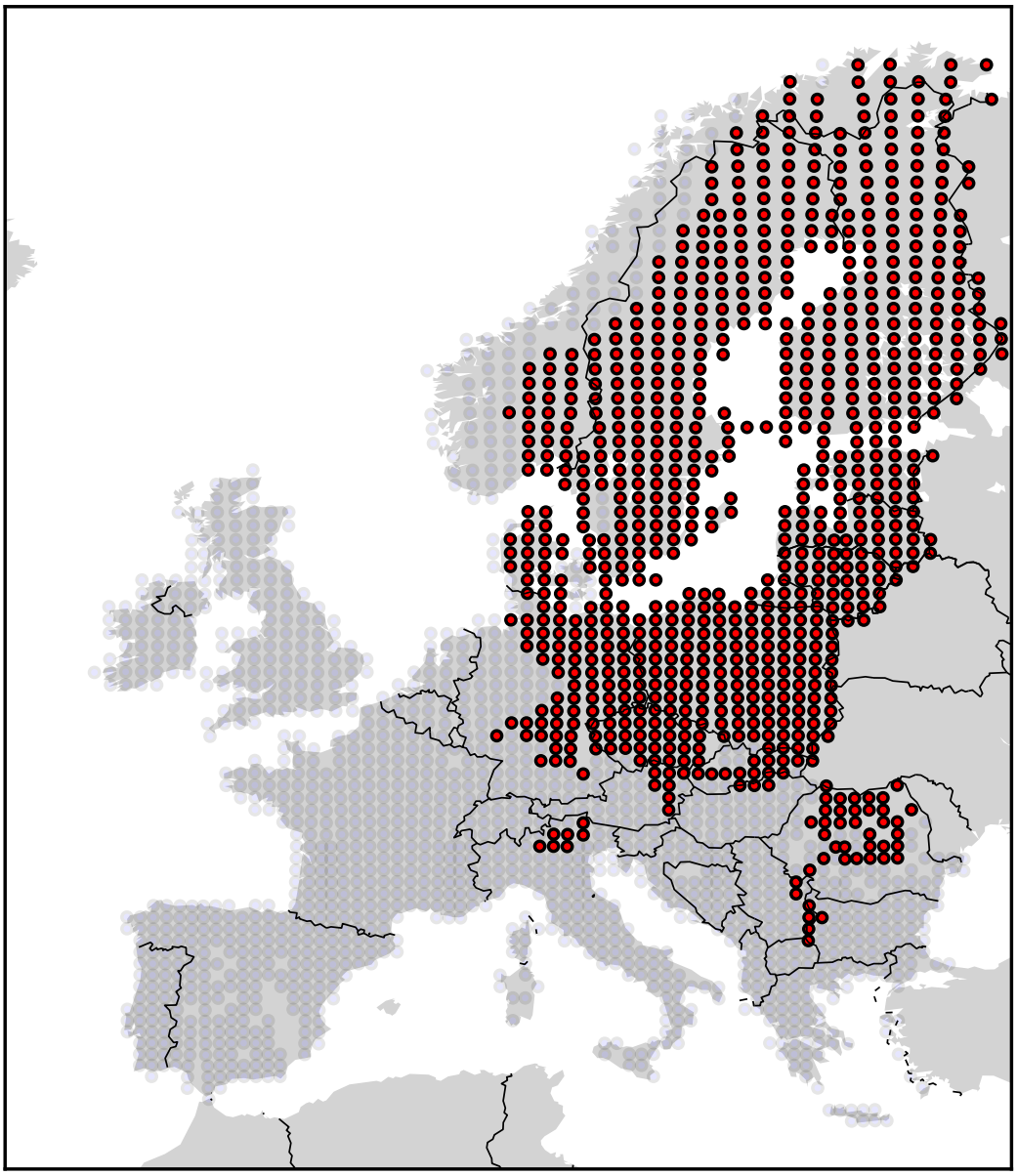


Figure 6.6: Regions in Europe that belong to the subgroup corresponding to D_{11} : $max_temp_nov \leq 7.66 \wedge prec_feb \leq 45.38$ ($|G_{11}| = 835$).

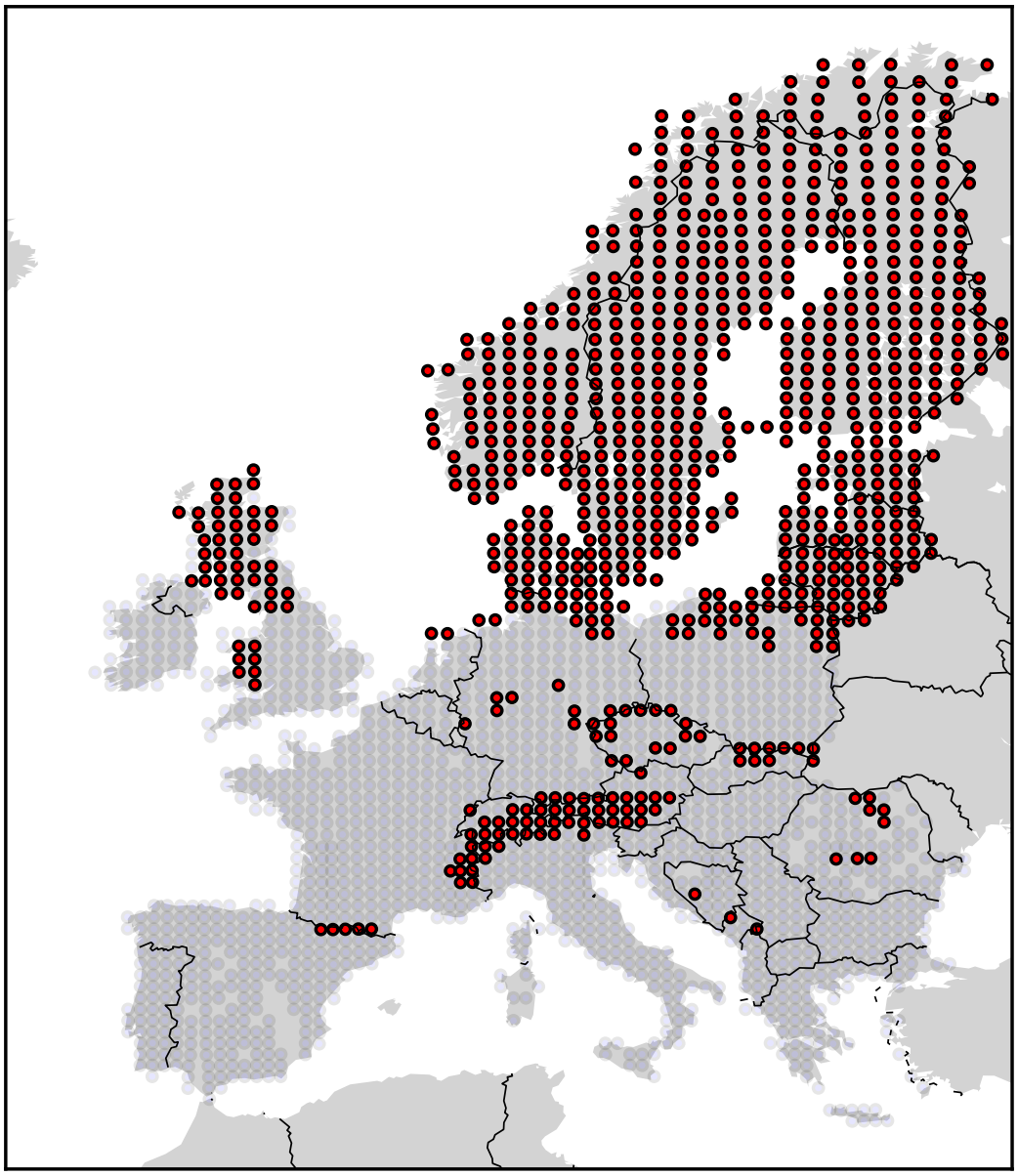


Figure 6.7: Regions in Europe that belong to the subgroup corresponding to $D_{12} : max_temp_mar \leq 7.97 \wedge max_temp_sep \leq 17.65$ ($|G_{12}| = 834$).

dent given the Western Roe Deer (*Capreolus capreolus*), only on D_{11} the Broad-toothed Field Mouse (*Apodemus mysticanus*) and the Lesser Mole Rat (*Nannospalax leucodon*) are conditionally dependent given the Marbled Polecat (*Vormela peregusna*), and only on D_{12} the Red Squirrel (*Sciurus vulgaris*) and the Least Weasel (*Mustela nivalis*) are conditionally dependent given the European Badger (*Meles meles*).

6.3 Alternatives

In Section 6.1.2, we discussed how we incorporated an entropy term in our quality measure φ_{weed}, in order to avoid obtaining small subgroups. If small subgroups are required, we can also run this EMM instance with the non-composite quality measure φ_{ed}, selecting the good descriptions only by virtue of their edit distance on Bayesian networks. To illustrate what the outcome of such a run can be, we repeated the experiments from the previous section on the *Mammals* dataset with φ_{ed} instead of φ_{weed}. The first-ranked description we found with this distance is D_{13} : *mean_temp_apr* $\geq 11.86 \wedge$ *mean_temp_aug* ≤ 23.28. Its quality is $\varphi_{\text{ed}}(D_{13}) = 0.147$, and its coverage is $|G_{13}| = 105$ (4.7%). The regions in Europe that belong to this description are displayed in Figure 6.8.

The relations between mammals that distinguish D_{13} from Ω include the following. On Ω, but not on D_{13}, the Alpine Marmot (*Marmota marmota*) and the Alpine Field Mouse (*Apodemus alpicola*) are conditionally dependent given the Alpine Ibex (*Capra ibex*), and the Beech Marten (*Martes foina*) and the Red Fox (*Vulpes vulpes*) are conditionally dependent given the Least Weasel (*Mustela nivalis*). On D_{13}, but not on Ω, the Common Genet (*Genetta genetta*) and the European Mink (*Mustela lutreola*) are conditionally dependent given the Crowned Shrew (*Sorex coronatus*), and the European Snow Vole (*Chionomys nivalis*) and the Iberian Shrew (*Sorex granarius*) are conditionally dependent given the Lusitanian Pine Vole (*Microtus lusitanicus*).

Using plain φ_{ed} instead of the composite φ_{weed} has its benefits and its drawbacks. When we compare the description D_{13} found with φ_{ed}, with the descriptions D_{10}, D_{11}, and D_{12} found with φ_{weed}, there are several things to remark. As expected, using the plain edit distance leads EMM to report

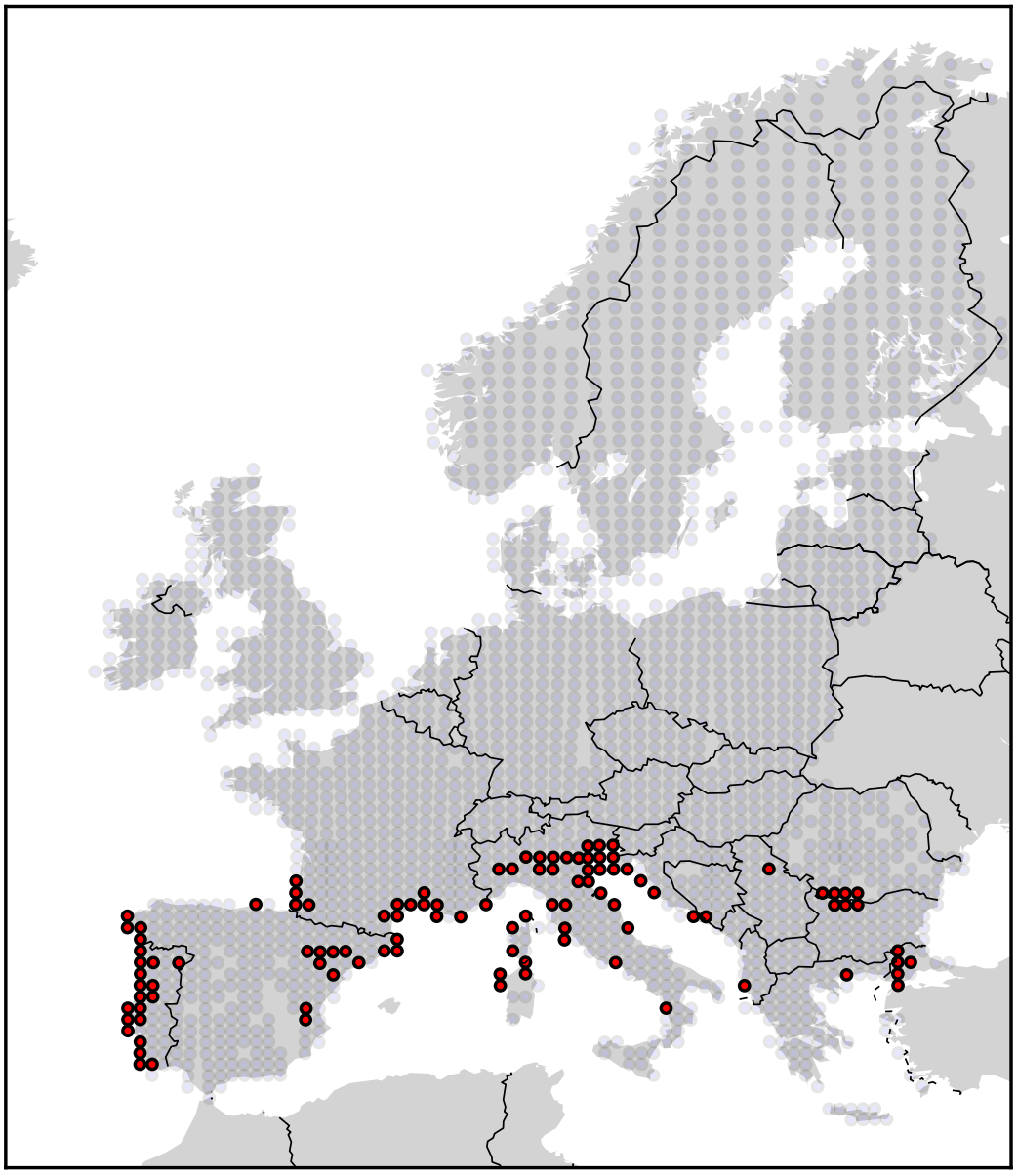


Figure 6.8: Regions in Europe that belong to the subgroup corresponding to D_{13} : $mean_temp_apr \geq 11.86 \wedge mean_temp_aug \leq 23.28$ ($|G_{13}| = 105$).

smaller subgroups than we obtain when using the edit distance weighted with entropy. Whether this is an argument for using φ_{ed} or φ_{weed} depends on the problem statement or domain expert at hand.

When we look at the deviating conditional dependence relations between the mammals, we find that particularly in the description found with the plain edit distance, the relations tend to focus on mammals that appear only in a very small subarea of Europe. For instance, within the parts of Europe covered by the dataset, the European Mink only occurs in a small area in the South West of France and the North of Spain, while the Iberian Shrew and the Lusitanian Pine Vole are confined to the Iberian peninsula. So, roughly speaking, φ_{ed} can be seen as more focused than φ_{weed}.

On the other hand, if we look at the maps of regions of Europe belonging to the subgroups, we see that φ_{weed} finds subgroups that are, geographically speaking, more coherent than the subgroup found with φ_{ed}. As we can see in Figure 6.5, subgroup G_{10} spans the North West of Europe, and as we can see in Figure 6.6, subgroup G_{11} spans the North East of Europe. At first glance, the area depicted in Figure 6.7 seems to indicate that subgroups G_{12} spans a dichotomous part of Europe: part is coherent, spanning Scandinavia, Scotland, Wales, and the Baltic countries, but to the South of that we find what appears to be rubble. However, if we compare this chart to a map of Europe indicating altitude, we find that the "rubble" actually largely overlaps with mountainous areas: we have found the Alps, the Pyrenees, the Harz, and the Carpathians. So, G_{12} spans some Northern areas, and some mountainous areas. By contrast, the regions belonging to subgroup G_{13}, as depicted in Figure 6.8, are far more scattershot. The coastal line of Portugal is a fairly coherent part of the subgroup, but the remaining areas seem relatively random. Although "mediterranean coastal" is a recurring theme, the selection of parts of the mediterranean coast seems incoherent, as does the isolated grid cell in Serbia and the small chunks in Bulgaria and Turkey. Hence, roughly speaking, φ_{weed} seems to deliver more substantially coherent subgroups than φ_{ed}.

6.4 Conclusions

In this chapter, we propose to use the interdependencies between discrete target variables as an exceptionality measure for descriptions. These interdependencies are modeled by Bayesian networks, and the quality of a description is defined as the difference between the network on the whole dataset and the network on the subgroup. To quantify this difference and thus the exceptionality of the model, we define a distance metric on Bayesian networks with the same vertex set. Experiments show that substantial findings on four domains can be made.

Compared to the previous two chapters, the model class in the current chapter is substantially more complex. This allows EMM to search for deviations in sophisticated interplay between multiple targets simultaneously. However, the price we pay for this advantage, is that interpreting results becomes problematic. As always, the found descriptions themselves can still be interpreted easily by a domain expert. Whether interpretation of the associated models is possible, however, depends on the number of targets in the dataset at hand.

As we have seen in our analysis of the results on the *Emotions* and *Scene* datasets, we can obtain meaningful insights from comparing Bayesian networks having six vertices. However, on the *Yeast* dataset the Bayesian network contains fourteen vertices, and on the *Mammals* dataset the network contains 101 vertices. For such large networks, we can still analyze the models associated with descriptions in a limited way, by highlighting dependence relations in small subsets of the vertices that differ between the description and the whole dataset. Having an overview of deviating (conditional) dependence relations between entire networks, however, has become impossible.

In such cases, it helps when the dataset has a third set of attributes, in addition to the descriptors and the targets. In the *Mammals* dataset, such a third set is available: the location information of grid cells throughout Europe. If a description, defined on the first set of attributes and evaluated on the second set, also displays coherence on the third set of attributes, then this reinforces our belief that we have found something substantial in our dataset. For instance, the fact that the geographically coherent region of

the Alps is highlighted in Figure 6.7, even though D_{12} was neither defined nor evaluated on location information, is strong corroborating evidence that this description indicates an actual underlying phenomenon in the dataset.

The work presented in this chapter can be extended in various ways. For instance, we could integrate our approach with the Hellinger distance introduced in Section 5.3.2, to determine the exceptionality of a description by comparing underlying probability distributions. Considering the Bayesian network parameters, or merely the signs of the correlations for ordered variables, could also improve our method.

Perhaps the most promising direction in which this EMM approach could be employed will be explored in Chapter 9: as a building block to be used in the Local Pattern Discovery phase in the LeGo framework [57]. As our descriptions identify parts of the input space where exceptional sets of dependencies hold, they can be thought of as a means to simplify a given multi-label classification problem, by allowing for different classification models in different descriptions. As descriptions may represent more coherent samples of the data, compared to the whole database, it can be expected that the LeGo building blocks can be employed to improve predictive accuracy.

Acknowledgments

The European mammals data was kindly provided by Tony Mitchell-Jones and the Societas Europaea Mammalogica.

Chapter 7

Different Slopes for Different Folks – Regression Model

In Chapter 2, we have discussed the Giffen effect. This effect concerns circumstances under which the economic law of demand is broken. Normally, all else equal, demand for a product will decrease if its price increases. However, given certain conditions on the different kinds and relative prices of available food sources (cf. Chapter 2), this relation reverses for the poor but not too poor households: for them the demand for a certain product will *increase* if its price increases. The relation between price of and demand for products is captured by a regression model.

Inspired by this example, we consider the Exceptional Model Mining instance with regression as model class: seeking descriptions for which (a subset of) the parameter vector β significantly deviates from the parameter vector estimated on the whole dataset. The targets $\ell_1, \ldots, \ell_m$ are internally supervised: ℓ_m is the output of the regression model, and $\ell_1, \ldots, \ell_{m-1}$ are the input variables. Formally, we learn the model $Y = X\beta + \varepsilon$, where Y is the $N \times 1$ vector[1] of ℓ_m-values from our dataset, and X is the $N \times m$ full rank matrix of which column 1 consists of N times the value 1 (representing the intercept in the regression model) and the other columns contain the ℓ_i-values from our dataset. So, in matrix form, we have

[1] We explicitly give both dimensions of all vectors for two reasons: on the one hand, to clearly indicate whether the vector comes in row or column form; on the other hand, to facilitate checking that the dimensions match in subsequent matrix products.

$$Y = \begin{pmatrix} \ell_m^1 \\ \ell_m^2 \\ \vdots \\ \ell_m^N \end{pmatrix} \qquad X = \begin{pmatrix} 1 & \ell_1^1 & \ell_2^1 & \cdots & \ell_{m-1}^1 \\ 1 & \ell_1^2 & \ell_2^2 & \cdots & \ell_{m-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \ell_1^N & \ell_2^N & \cdots & \ell_{m-1}^N \end{pmatrix} \qquad Y = X\beta + \varepsilon$$

Here, β is the $m \times 1$ vector of the unknown regression parameters, and ε is the $N \times 1$ vector of randomly distributed errors such that $\mathbb{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \mathrm{diag}(\sigma^2 I)$ (where I denotes the $N \times N$ identity matrix).

Given an estimate of the vector β, denoted $\hat{\beta}$, one can compute the vector of fitted values $\hat{Y}$. These quantities can be used to assess the appropriateness of the fitted model, by looking at the residuals $e = Y - \hat{Y}$. We will estimate β with the ordinary least squares method, which minimizes the sum of squared residuals. This leads [41] to the estimate

$$\hat{\beta} = (\hat{\beta}_i) = \left(X^\top X\right)^{-1} X^\top Y$$

After computing the vector of fitted values, we find that we can now write the corresponding residual vector as

$$e = (e_i) = Y - \hat{Y} = \left(I - X\left(X^\top X\right)^{-1} X^\top\right) Y$$

We will denote a part of this equation by V, i.e.

$$V = (v_{ij}) = X\left(X^\top X\right)^{-1} X^\top$$

This matrix was dubbed the *hat matrix* by John W. Tukey, since $\hat{Y} = VY$, i.e. the hat matrix transforms Y into $\hat{Y}$ [51].

7.1 Quality Measure φ_{Cook}

In order to define a proper quality measure for comparing estimated parameter vectors, we need to take into account the variance of the estimator $\hat{\beta}$, and the covariances between $\hat{\beta}_i$ and $\hat{\beta}_j$. For example, if $\hat{\beta}_i$ has a large variance compared to $\hat{\beta}_j$, then a given change in $\hat{\beta}_i$ should contribute less

to the overall quality than the same change in $\hat{\beta}_j$, because the change in $\hat{\beta}_i$ is more likely to be caused by random variation. This suggest that

$$\left(\hat{\beta}^G - \hat{\beta}\right)^\top \left[\mathrm{Cov}\left(\hat{\beta}\right)\right]^{-1} \left(\hat{\beta}^G - \hat{\beta}\right)$$

might be a better distance measure than the normal Euclidian distance. In fact this expression is equivalent to Cook's distance up to a constant scale factor. Cook originally introduced his distance [13] in 1977 for determining the contribution of single records to $\hat{\beta}$. He states that according to normal theory [42], the $(1-\alpha)\times 100\%$ confidence ellipsoid for the unknown vector, β, is given by the set of all vectors β^* satisfying

$$\frac{\left(\beta^* - \hat{\beta}\right)^\top \left[\widehat{\mathrm{Cov}}\left(\hat{\beta}\right)\right]^{-1} \left(\beta^* - \hat{\beta}\right)}{m} =$$

$$\frac{\left(\beta^* - \hat{\beta}\right)^\top X^\top X \left(\beta^* - \hat{\beta}\right)}{m s^2} \leq F(m, N-m, 1-\alpha)$$

where

$$s^2 = \frac{e^\top e}{N-m} \qquad \widehat{\mathrm{Cov}}\left(\hat{\beta}\right) = s^2 \left(X^\top X\right)^{-1}$$

and $F(m, N-m, 1-\alpha)$ is the $1-\alpha$ probability point of the central F-distribution with m and $N-m$ degrees of freedom. Here, s^2 is the unbiased estimator for σ^2.

We can exploit the confidence ellipsoid and F-distribution to define a quality measure suitable for the current EMM instance, with some desirable properties. On the one hand, it respects the (co-)variances present in the data, as discussed at the beginning of this section. On the other hand, it comes with theoretical upper bounds that can be utilized to prune the search space, as we will discuss in Section 7.3. To arrive at the definition of the quality measure, however, we first need to determine the degree of influence of single records on the parameter vector. Then we will discuss generalizing this to the influence of deleting multiple records simultaneously. After that, we can give the definition.

Suppose we want to know how a single record r^i influences $\hat{\beta}$. Then one could naturally compute the least squares estimate for β with the record removed from the dataset. Let us denote this estimate by $\hat{\beta}_{(i)}$. We can

adapt the confidence ellipsoid as an easily interpretable measure of the distance between $\hat{\beta}_{(i)}$ and $\hat{\beta}$. Hence, *Cook's distance* is defined as

$$\Delta_i = \frac{\left(\hat{\beta}_{(i)} - \hat{\beta}\right)^\top X^\top X \left(\hat{\beta}_{(i)} - \hat{\beta}\right)}{ms^2}$$

Suppose for example that for a certain record r^i we find that $\Delta_i \approx F(m, N-m, 0.5)$. Then removing r^i moves the least squares estimate to the edge of the 50% confidence region for β based on $\hat{\beta}$.

Cook and Weisberg extended Cook's distance to the case where multiple records are deleted simultaneously [14]. Let I be a vector of indices that specify the h records to be deleted. From now on, we let the subscript (I) denote "with the h cases indexed by I deleted", while the subscript I without parentheses denotes "with only the h cases indexed by I remaining". The only notation that deviates from this rule of thumb is the definition of Cook's distance for multiple observations, which becomes

$$\Delta_I = \frac{\left(\hat{\beta}_{(I)} - \hat{\beta}\right)^\top X^\top X \left(\hat{\beta}_{(I)} - \hat{\beta}\right)}{ms^2} \tag{7.1}$$

Its geometric interpretation is identical to the geometrical interpretation of Δ_i. Any subset that has a large joint influence on the estimation of β will result in a large Δ_I.

The fact that the definition of Cook's distance does not follow the notational rule of thumb can be very confusing. We choose to retain the definition in this form to make our work compatible with previously released papers and books. However, it is important to stress the notational anomaly: whenever we write D_I, Cook's distance is computed for the case where the records indexed by I are *deleted*. Whenever we write *anything else* with a subscript I, it is computed for the case where the records indexed by I are *retained*, and all other records are deleted.

For practical purposes one might not be interested in computing Cook's distance based on the entire parameter vector $\hat{\beta}$. For instance, one might be interested in the influence records have on the regression coefficient corresponding to one particular attribute, while excluding the intercept and other coefficients from the evaluation. To this end, Cook and Weisberg [15] introduce the zero/one-matrix Z, with dimensions $m' \times m$, where m'

is the number of elements of $\hat{\beta}$ that we are interested in (hence $m' \leq m$). The matrix Z is defined in such a way that $\psi = Z\beta$ are the coefficients of interest. Hence, if we are interested in the last m' elements of $\hat{\beta}$, Z will start from the left with $m - m'$ columns containing all zeroes, followed by a $m' \times m'$ identity matrix ($Z = (\mathbf{0}, \mathbf{I}_{m'})$).

When using this transformation, Cook's distance (Equation (7.1)) becomes

$$\Delta_I^{\psi} = \frac{\left(\hat{\beta}_{(I)} - \hat{\beta}\right)^{\top} Z^{\top} \left(Z \left(X^{\top} X\right)^{-1} Z^{\top}\right)^{-1} Z \left(\hat{\beta}_{(I)} - \hat{\beta}\right)}{m' s^2}$$

Since Cook's distance is invariant to changes in scale of the variables involved [13], it would make an excellent quality measure for use in EMM

Definition (φ_{Cook}). Let D be a description. Its *quality according to Cook's distance* is given by

$$\varphi_{\text{Cook}}(D) = \Delta_I^{\psi}, \text{ where } I = \left\{ i \;\middle|\; r^i \in \Omega, D\left(a_1^i, \ldots, a_k^i\right) = 0 \right\}$$

The quality of a description according to Cook's distance is the distance bridged when the records *not covered by the description* are simultaneously *discarded*. Hence, Cook's distance is computed for the case where the records covered by the description D are *retained*.

7.2 Experiments

7.2.1 Datasets

The *Giffen Behavior* dataset was used for a study that provided the first real-world evidence of Giffen behavior, i.e. an upward sloping demand curve [77]. As common sense suggests, the demand for a product will normally decrease as its price increases. According to economic textbooks, there are circumstances however, for which the demand curve should slope upward. The common example is that of poor families that spend most of their income on a relatively inexpensive staple food (e.g. rice or wheat) and a small part on a more expensive type of food (e.g. meat). If the price of the staple food rises, people can no longer afford to supplement their diet with the more expensive food, and must consume more of the staple food.

The dataset we analyze [53] was collected in a field study in different counties in the Chinese province Hunan, where rice is the staple food. The price changes were brought about by subsidizing the purchase of rice. Each household was randomly assigned to either a control group, or one of three treatment groups. Households in the treatment groups were given vouchers worth ¥0.10, ¥0.20, or ¥0.30, redeemable at selected vendors for a reduction off the price of each *jin* (1 *jin* equals 500 grams) of rice. The average price of rice in Hunan is ¥1.20 per *jin*, so the vouchers represented substantial price changes. The programme provided vouchers for a time period of five months, and subsidized for each person an amount of rice, equal to roughly twice the average per capita consumption.

Data were gathered on three points in time: before the voucher programme started, while the voucher programme was running, and after the voucher programme had ended. Hence, for each household, two changes are observed: the change between the first and second period ($t = 2$), capturing the effect of giving the subsidy; and the change between the second and third period ($t = 3$), capturing the effect of removing the subsidy. The global model estimated in [53] is

$$\%\Delta staple_{i,t} = \beta_0 + \beta_1 \%\Delta p_{i,t} + \sum \beta_2 \%\Delta Z_{i,t} + \sum \beta_3 County \times Time_{i,t} + \Delta\varepsilon_{i,t}$$

where $\%\Delta staple_{i,t}$ denotes the percent change in household i's consumption of rice, $\%\Delta p_{i,t}$ is the percent change in the price of rice due to the subsidy (negative for $t = 2$ and positive for $t = 3$), $\%\Delta Z_{i,t}$ is a vector of percent changes in other control variables including income and household size, and $County \times Time$ denotes a set of dummy variables included to control for any county-level factors that change over time. For further details about the design of the study and the estimation strategy, we refer to [53].

The *Ames Housing* dataset contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, Iowa from 2006 to 2010. It consists of 2930 observations on 82 variables. The global model is

$$Price = \beta_0 + \beta_1 \times Lot\ Area + \beta_2 \times Quality$$

where *Price* is the sales price of the house in dollars, *Lot Area* is the lot size in square feet, and *Quality* rates the overall material and finish of the house on a scale from 1 to 10.

The *Auction* dataset was analyzed in [93]. It concerns eBay auctions of Apple iPod mini players from June 27 to July 18, 2006. The goal is to model the final price reached in the auction in terms of auction, seller, and product characteristics. The global model is

$$\begin{aligned} Price &= \beta_0 + \beta_1 \times Nbid + \beta_2 \times PositiveFeedback + \beta_3 \times Time \\ &\quad + \beta_4 \times FeedbackScore + \beta_5 \times Memory + \beta_6 \times ResPrice \end{aligned}$$

where *Price* is the final price of the auction in US dollars, *Nbid* is the number of distinct people who bid in the auction, *PositiveFeedback* is the seller's positive feedback percentage (the coefficient is nonzero from the fourth decimal place), *Time* is the time of he final bid expressed in seconds after Dec. 31 1969, 22:00:00 PDT (the coefficient is nonzero from the fifth decimal place), *FeedbackScore* is the seller's feedback score, *Memory* is the reported memory of the iPod in gigabytes, and *ResPrice* is the auction reservation price in US dollars.

The *EAEF* dataset was extracted from the National Longitudinal Survey of Youth 1979 (NLSY79). It contains information about hourly earnings of men and women, their education, and other information. For more details, see [22, Appendix B]. We fit a model relating years of schooling and years of work experience to earnings in US dollars per hour. The model fitted on the whole dataset is

$$Earnings = \beta_0 + \beta_1 \times YrsOfSchool + \beta_2 \times YrsWorkExp$$

The *Personal Computer* dataset was analyzed in [99]. The data was collected from advertisements in PC Magazine. Each observation consists of the advertised price and features of personal computers. We have learned the following model from the complete dataset

$$\begin{aligned} Price &= \beta_0 + \beta_1 \times Spd + \beta_2 \times HD + \beta_3 \times RAM \\ &\quad + \beta_4 \times Scr + \beta_5 \times Ads + \beta_6 \times Trend \end{aligned}$$

where *Price* is the price in US dollars of a 486 PC, *Spd* is the clock speed in MHz, *HD* is the size of the hard drive in MB, *RAM* is the size of RAM in MB, *Scr* is the size of the screen in inches, *Ads* is the number of 486 price listings in the month the advertisement was placed, and *Trend* is a time trend starting from January of 1993 to November of 1995.

Table 7.1: Statistics concerning the datasets used in the Regression Model experiments. Here, N is the total number of records, k is the number of descriptive attributes , and m is the number of coefficients in the fitted regression model.

Dataset	Domain	N	k	m
Ames Housing	Residential property value	2930	77	3
Auction	eBay auctions	1225	3	7
EAEF	Employment	2714	32	3
Giffen Behavior	Food economics	1254	6	16
Personal Computer	PC pricing	6259	3	7
Wine	Wine pricing	5000	6	4

Finally, the *Wine* dataset was analyzed in [16]. It is composed of 9600 observations derived from 10 years (1991–2000) of tasting ratings reported in the Wine Spectator Magazine (online version) for California and Washington red wines. Our analysis uses a random sample of size 5000 from the original data. For a detailed description of the data we refer to [16]. The global model is

$$Price = \beta_0 + \beta_1 \times Cases + \beta_2 \times Score + \beta_3 \times Age$$

where *Price* is the retail price suggested by the winery, *Score* is the score from the Wine Spectator, *Age* is the years of aging before commercialization, and *Cases* is the number of cases produced (in thousands). All coefficients have the sign that one would expect based on common sense.

Table 7.1 lists some elementary properties of these datasets.

7.2.2 Experimental Results

Giffen Behavior Data

The global model estimated on the *Giffen Behavior* dataset is

$$\%\Delta staple_{i,t} = \beta_0 + \beta_1 \%\Delta p_{i,t} + \sum \beta_2 \%\Delta Z_{i,t} + \sum \beta_3 County \times Time_{i,t} + \Delta\varepsilon_{i,t}$$

The coefficient of primary interest is β_1. If $\beta_1 > 0$ we observe Giffen behavior. The other variables are included in the model to control for other

possible influences on demand, so that the effect of price can be reliably estimated. Therefore it makes sense to restrict our quality measure to the coefficient β_1.

The authors of [53] suggest that for the extremely poor, one might not observe Giffen behavior, because they consumed rice almost exclusively anyway, and therefore have no other possibility than buying less of it in case of a price increase. The *Initial Staple Calorie Share (ISCS)* was also measured in the study, and the hypothesis is that families with a high value for this variable do not display Giffen behavior. The authors of [53] tried different manually selected thresholds on *ISCS*; for example, for the subgroup of households with $ISCS > 0.8$, indeed it is observed that $\hat{\beta}_1 = -0.585$ (no Giffen behavior) whereas for $ISCS \leq 0.8$ they find $\hat{\beta}_1 = 0.466$ (Giffen behavior).

We analyzed this dataset with *ISCS* as one of the descriptive attributes. The best description we found was D_{14} : $ISCS \geq 0.87$ with $\hat{\beta}_1 = -0.96$ (against $\hat{\beta}_1 = 0.22$ for the complete dataset). The coverage of this description is $|G_{14}| = 106$ (3.9%). This confirms the conclusion that Giffen behavior does not occur for families that almost exclusively consume rice anyway. This conclusion can also be reached by defining subgroups on *income per capita* rather than *ISCS*. Particularly illustrative examples are the 4^{th}-ranked description D_{15} : *Income per Capita* ≤ 64.67, with a slope of -0.41, and the 6^{th}-ranked description D_{16} : *Income per Capita* ≥ 803.75, with a slope of 0.79 (strong Giffen behavior).

Ames Housing Data

The global model for the *Ames Housing* dataset is

$$Price = -108225.05 + 1.93 \times Lot\ Area + 44201.87 \times Quality$$

By far the most deviating description we find is D_{17}, where the building type is a *'townhouse inside unit'*. For D_{17}, the learned model is

$$Price = -17674.20 + 24.62 \times Lot\ Area + 15786.88 \times Quality$$

The coverage of this description is $|G_{17}| = 101$ (3.4%). The dependence of *Price* on *Lot Area* is much stronger for town houses, whereas the dependence of price on overall *Quality* is less strong than in general. In an

attempt to explain this pattern, we note that the average lot area of town houses (2353 square feet) is much smaller than the overall average (10148 square feet) which is largely determined by the predominant building type *'single family detached'*. Furthermore, it stands to reason that for townhouses a larger part of the lot area is actually occupied by the house itself than for the single family detached houses. This is consistent with a much stronger dependence of their price on the lot area.

EAEF Data

The global model fitted on the $EAEF$ dataset is

$$Earnings = -29.15 + 2.78 \times YrsOfSchool + 0.63 \times YrsWorkExp$$

The 4^{th} ranked description we found was $D_{18} : COLLBARG = 1$, meaning that the pay was set by collective bargaining. The learned model for this description with coverage $|G_{18}| = 533$ (19.6%) is

$$Earnings = -8.93 + 1.57 \times YrsOfSchool + 0.43 \times YrsWorkExp$$

This suggests that for this group an extra year of schooling on average leads to an increase of just \$1.57 in hourly earnings, compared to \$2.78 for the whole dataset. The same is true for the influence of an extra year of work experience: just \$0.43 for the collective bargaining subgroup, against \$0.63 in the complete dataset. This is consistent with the finding that unions tend to equalize the income distribution, especially between skilled and unskilled workers [1].

Personal Computer Data

The global model for the *Personal Computer* dataset is

$$\begin{aligned} Price = &-246.68 + 8.89 \times Spd + 0.71 \times HD + 47.39 \times RAM \\ &+ 126.70 \times Scr + 0.97 \times Ads - 47.08 \times Trend \end{aligned}$$

By far the most important attribute for defining descriptions was whether or not the company was a *premium firm* (IBM or COMPAQ). The most deviating description was D_{19}, the *non-premium firms*, with model

$$\begin{aligned} Price = &-2130.21 + 13.15 \times Spd + 2.31 \times HD + 22.20 \times RAM \\ &+ 252.80 \times Scr + 0.79 \times Ads - 46.45 \times Trend \end{aligned}$$

The coverage of this description is $|G_{19}| = 612$ (9.8%). We get the clearest picture when we contrast this with the regression model fitted to the *premium firms*, which is

$$\begin{aligned} Price = &\ 165.69 + 8.50 \times Spd + 0.67 \times HD + 53.66 \times RAM \\ &+ 99.96 \times Scr + 0.65 \times Ads - 47.87 \times Trend \end{aligned}$$

The coverage of this description is $|G_{19}^{C}| = 5647$ (90.2%). We find mostly reasonable behavior in these subgroups: the price of computers from premium firms is based on a far higher intercept, since the premium brand name ensures a vast price upkeep. Consequently, other factors have a substantially smaller impact on the price than for computers from non-premium firms. Oddly, the size of RAM memory does matter more strongly for premium brands than for non-premium brands.

Wine Data

On the *Wine* dataset, the global model is

$$Price = -186.61 - 0.0002 \times Cases + 2.35 \times Score + 5.51 \times Age$$

The most deviating description is D_{20} : $Variety = \text{'Non-varietal'}$ (alternatives are *'Pinot noir'*, *'Cabernet'*, *'Merlot'*, *'Zinfandel'* and *'Syrah'*). The regression model for D_{20} is

$$Price = -341.92 - 0.0004 \times Cases + 4.16 \times Score + 7.22 \times Age$$

'Non-varietal' means that multiple varieties of grapes are used, and on average these wines are more expensive than the single-variety wines (average price of \$44.16 against \$28.89). People buying those more expensive wines tend to be better informed (e.g. read Wine Spectator Magazine) than the average buyer. This explains to a certain extent why the price of those more expensive wines is more sensitive to its score and age: their buyers are more critical.

7.3 Pruning with Bounds for Cook's Distance

Cook's distance is a theoretically well-founded quality measure for mining descriptions for which the slope vector deviates. The bad news is that its computation involves the computation of $\hat{\beta}^G$, which implies that we need to invert a matrix for each candidate description. This is computationally very expensive. Fortunately, some upper bounds have been derived for Cook's distance, which we can use to discard some candidates without having to invert a matrix.

The upper bounds for Cook's distance are derived [15, p. 136] by rewriting the numerator of the right hand side of Equation (7.1) in terms of e_I and V_I. Then the spectral decomposition of V_I is used, rewriting the sub-matrix of the hat matrix in terms of its eigenvalues and eigenvectors. We denote those eigenvalues by $\lambda_1, \ldots, \lambda_h$, and can assume without loss of generality that $0 \leq \lambda_1 \leq \ldots \leq \lambda_h \leq 1$. Notice that if the last inequality is not strict, i.e. $\lambda_h = 1$, then removing the records indexed by I would lead to a rank deficient model, and we cannot properly perform the linear regression. Finally, a proper approximation for these λ_i is required; Cook proposes to use $\mathrm{tr}(V_I)$ here, but notes that this is only a good approximation under the condition that $\mathrm{tr}(V_I) < 1$. Assuming that this condition holds, we can bound D_I by

$$D_I \leq \frac{\mathrm{tr}(V_I)}{(1 - \mathrm{tr}(V_I))^2} \cdot \frac{\sum_{i \in I} e_i^2}{m s^2} \tag{7.2}$$

Unfortunately, this bound is potentially different for each I. Cook also gives bounds that hold for all subsets I of a fixed size h. When we fix h and let I vary over all such subsets, we can either use $R^2 = \max_I \left(\sum_{i \in I} e_i^2\right)$, which turns Equation (7.2) into

$$D_I \leq \frac{\mathrm{tr}(V_I)}{(1 - \mathrm{tr}(V_I))^2} \cdot \frac{R^2}{m s^2} \tag{7.3}$$

or we could use $T = \max_I (\mathrm{tr}(V_I))$, which turns Equation (7.2) into

$$D_I \leq \frac{T}{(1 - T)^2} \cdot \frac{\sum_{i \in I} e_i^2}{m s^2} \tag{7.4}$$

Both estimates can be combined to turn Equation (7.2) into

$$D_I \leq \frac{T}{(1-T)^2} \cdot \frac{R^2}{ms^2} \tag{7.5}$$

Rather obviously, there are relations between the bounds: bound (7.2) is stricter than both bound (7.3) and bound (7.4), and those are both stricter than bound (7.5).

Whenever one has the possibility to enumerate all candidate descriptions for mining with Cook's distance, the bounds (7.2)–(7.5) can be used for pruning. In combination with the beam search strategy for top-q EMM, we propose to do this in the following way.

Per search level, we enumerate all candidate descriptions in descending order according to bound (7.5). Then we consider the subgroups in this order. For each description, we compute the bounds in order of decreasing ease of computation, i.e. first bound (7.5), then bound (7.4), then bound (7.3), and finally bound (7.2). We check whether any of these bounds has a value that is lower than Cook's distance for the q^{th} best evaluated description so far. If so, we know that Cook's distance for this new description can not enter the top-q, since the bound is an upper bound for Cook's distance. Hence we can skip computing Cook's distance for this description, which saves us the computation of a relatively expensive regression. If none of the bounds help us out, we compute Cook's distance for the new description.

To illustrate what can reasonably be expected from pruning with the bounds, we simulate their behavior on random subsets of the *EAEF* dataset. For each possible subgroup size, we draw a random sample of the data with that size. Then we compute the values of the bounds for these subsets, when fitting the model

$$Earnings = \beta_0 + \beta_1 \times YrsOfSchool$$

The results can be found in Figure 7.1. The figure depicts the subset size on the x-axis (linear scale), and the values of the bounds on the y-axis (logarithmic scale).

The *EAEF* dataset has 2714 records, so when a subset approaches this size it will correspond to deleting very few records, and as one would expect, Cook's distance becomes very small, as do the bounds. Furthermore, one notices that the bound quality lines do not extend all the way to subset

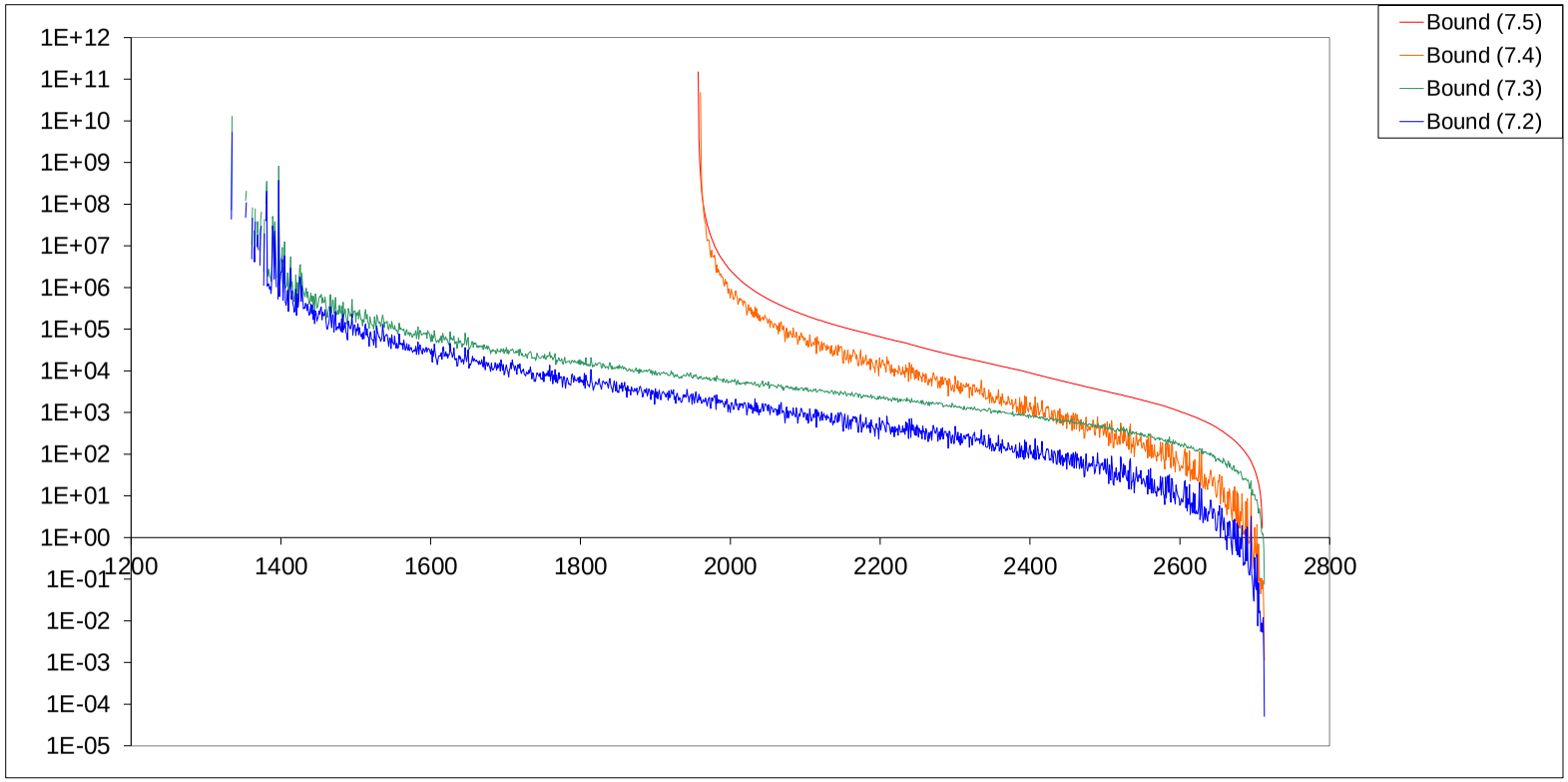


Figure 7.1: Bound values (logarithmic scale) for random subsets of different sizes on the *EAEF* dataset. Fitted model: $Earnings = \beta_0 + \beta_1 \times YrsOfSchool$.

size 0. This is caused by limitations in the approximations used in the bounds. As mentioned before, the bounds are only good approximations whenever $\mathrm{tr}(V_I) < 1$. When this constraint is not satisfied, the bounds cannot be computed. For bounds (7.4) and (7.5), the quantity T is used as an estimate for $\mathrm{tr}(V_I)$, but this too only makes sense when $T < 1$, or else the bounds cannot be computed.

The practical upshot is that for subsets having less than 1960 records, bounds (7.4) and (7.5) cannot be computed. For subsets having less than roughly 1250 records, this also holds for bounds (7.2) and (7.3). When viewed as a percentage of the number of records in the datasets, we find that these borders are roughly the same over all datasets: bounds (7.2) and (7.3) can only be computed when the subset contains at least 50% of the records, and bounds (7.4) and (7.5) only when the subset contains at least 75% of the records. We also find that the more complex the model we fit, the further these thresholds move towards larger percentages.

The bounds can not be computed for at least half of the subsets we consider, and the bound values tend to increase enormously just before these threshold values are reached. However, the bounds are computable for the largest subsets, and the computation of the hat matrix is quadratic in the subset size. Hence whenever we can prune a subset, it always takes a relatively expensive regression computation out of the total runtime.

7.3.1 Empirical bound evaluation

To empirically see how the bounds function, we performed a depth-1 EMM run on each dataset, with the goal to find the top-1 description. When numeric attributes were used to generate candidate descriptions, we split them into 12 equal-sized bins. We discarded any description covering fewer than 100 records, since we consider these too small to be considered interesting from a statistical point of view. For each bound we counted how often it was computed, and how often it caused a description to be pruned.

The results can be found in Table 7.2a. This table features the datasets, dataset characteristics, number of times every bound is computed, number of descriptions pruned with every bound, fraction of candidate descriptions for which at least one bound was computable, and fraction of candidate

Table 7.2: Pruning results for depth-1 EMM runs. Here, N is the total number of records, $\mathcal{C}$ is the set of candidate descriptions considered, and m is the number of coefficients in the fitted regression model. The set "bounded $\mathcal{C}$" consists of the candidate descriptions for which at least one bound could be computed, and the set "pruned $\mathcal{C}$" consists of the candidate descriptions that were pruned using one of the bounds.

(a) Results when looking for the top-1 description.

Dataset	N	$\lvert\mathcal{C}\rvert$	m	Bounds computed				Descriptions pruned				$\frac{\lvert\text{bounded } \mathcal{C}\rvert}{\lvert\mathcal{C}\rvert}$	$\frac{\lvert\text{pruned } \mathcal{C}\rvert}{\lvert\mathcal{C}\rvert}$
				(7.5)	(7.4)	(7.3)	(7.2)	(7.5)	(7.4)	(7.3)	(7.2)		
Ames Housing	2930	980	3	196	41	228	37	155	28	191	11	0.419	0.393
Auction	1225	40	7	5	0	9	5	5	0	4	0	0.350	0.225
EAEF	2714	204	3	35	29	68	68	6	9	0	21	0.407	0.176
Giffen Behavior	1254	100	16	1	1	1	1	0	0	0	1	0.010	0.010
PC486	6259	6	7	0	0	2	1	0	0	1	0	0.333	0.167
Wine	5000	56	4	2	2	26	20	0	0	6	11	0.464	0.304

(b) Results when looking for the top-50 descriptions.

Dataset	N	$\lvert\mathcal{C}\rvert$	m	Bounds computed				Descriptions pruned				$\frac{\lvert\text{bounded } \mathcal{C}\rvert}{\lvert\mathcal{C}\rvert}$	$\frac{\lvert\text{pruned } \mathcal{C}\rvert}{\lvert\mathcal{C}\rvert}$
				(7.5)	(7.4)	(7.3)	(7.2)	(7.5)	(7.4)	(7.3)	(7.2)		
Ames Housing	2930	980	3	196	125	272	122	71	68	150	44	0.419	0.340
EAEF	2714	204	3	35	34	77	77	1	5	0	11	0.407	0.083

descriptions that were pruned. Notice that there is a strong dependency between the "Bound computed" and "Descriptions pruned" columns: in the *Ames Housing* dataset we can compute bound (7.5) for 196 descriptions, of which we can prune 155, so only 41 subgroups remain for which we compute bound (7.4). However, the number of descriptions for which we compute bound (7.3) is larger, since the condition under which this bound is computable is less strict than the condition for bound (7.4) and (7.5). Of the 228 descriptions for which we compute bound (7.3) we can prune 191, leaving 37 descriptions for which we compute bound (7.2).

As we indicated earlier, the fraction of descriptions for which we can compute the bounds is strongly dependent on the complexity of the fitted model. As we can see from the table, in the datasets for which $3 \leq m \leq 4$ we can compute bounds for over 40% of the descriptions, in the datasets for which $m = 7$ we can compute bounds for $33 - 35\%$ of the descriptions, and in the dataset for which $m = 16$ we can compute bounds for just 1% of the descriptions. This dependency becomes somewhat less direct when we look at the percentage of descriptions we can actually prune, since this is relatively low for the *EAEF* dataset on which we fit a relatively simple model. However, apart from this one dataset, we still see a strong relation between model simplicity and pruning success.

Since we are rarely interested in only the one best-performing description, we replicate these experiments with the goal to find the top-50 descriptions. Since we need to have considered at least 50 descriptions before we can make sure others will not enter the top-50 based on their bounds, we know in advance that there will be little or no pruning possible for the *Auction*, *PC486*, and *Wine* datasets. We also expect to gain little information from the *Giffen Behavior* dataset, hence Table 7.2b encompasses the results of these experiments on merely the *Ames Housing* and *EAEF* dataset. Notice that the fraction of descriptions we can prune on the *Ames Housing* dataset has only decreased slightly, while the fraction of descriptions we can prune on the *EAEF* dataset is cut in half.

We repeat all these experiments with depth-2 EMM runs with beam width $w = 10$. We find that in these experiments, we can barely compute bounds for any level-2 descriptions, let alone prune them. This is caused by the fact that level-2 descriptions are refinements of well-scoring level-1 descriptions,

which usually cover relatively few records. Such descriptions scarcely ever cover more than 50% of the records, hence their refinements also scarcely ever do so. Fortunately, that also means that the regression computations for these level-2 descriptions is relatively cheap.

7.4 Alternatives

We can define a simpler, statistically founded quality measure when we restrict ourselves to a simpler regression model, allowing only one input ($y = \ell_2$) and one output variable ($x = \ell_1$) in the regression, i.e.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{7.6}$$

Consider model (7.6) learned from a subgroup G and its complement G^C. Of course, there is a choice of distance measures between the fitted models. We propose to look at the difference in the slope β_1 between the two models, because this parameter is usually of primary interest when fitting a regression model: it indicates the change in the expected value of y, when x increases with one unit. Another possibility would be to look at the intercept β_0, if it has a sensible interpretation in the application concerned. As with the correlation model, we use significance testing to measure the distance between the fitted models. Let β_1^G be the slope for the regression function of G and $\beta_1^{G^C}$ the slope for the regression function of G^C. The hypothesis to be tested is

$$H_0 : \beta_1^G = \beta_1^{G^C} \quad \text{against} \quad H_1 : \beta_1^G \neq \beta_1^{G^C}$$

We use the least squares estimate $\hat{\beta}_1$ for the slope β_1, and unbiased estimator s^2 for the variance of $\hat{\beta}_1$, i.e.

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \qquad s^2 = \frac{\sum \hat{\varepsilon}_i^2}{(\xi - 2)\sum (x_i - \bar{x})^2}$$

where $\hat{\varepsilon}_i$ is the regression residual for individual i, and ξ is the sample size. Finally, we define our test statistic t'. Although it does not have a t distribution, its distribution can be approximated quite well by one, with degrees of freedom given at the top of the next page (cf. [81])

$$t' = \frac{\hat{\beta}_1^G - \hat{\beta}_1^{G^C}}{\sqrt{s^{G^2} + s^{G^{C\,2}}}} \qquad df = \frac{\left(s^{G^2} + s^{G^{C\,2}}\right)^2}{\frac{s^{G^4}}{n-2} + \frac{s^{G^{C\,4}}}{n^C-2}}$$

The approximation is accurate when $n + n^C \geq 40$ (cf. [81]), so unless we analyze a very small dataset we should be confident to base p-value computation on it. Our quality measure φ_{ssd} (acronym for Significance of Slope Difference) is one minus this p-value.

Running EMM on the *Windsor Housing* dataset (cf. Table 4.1) using φ_{ssd} as quality measure, we find as first-ranked description D_{21} the 226 houses (41.3% of the dataset) that have a *driveway*, *no basement* and *at most one bathroom*

$$D_{21} : drive = 1 \wedge basement = 0 \wedge nbath \leq 1$$

From the subgroup G_{21} and its complement G_{21}^C (320 houses, 58.7%) we learn the following two regression functions, respectively

$$G_{21} : \; y = 41568 + 3.31 \cdot x$$
$$G_{21}^C : \; y = 30723 + 8.45 \cdot x$$

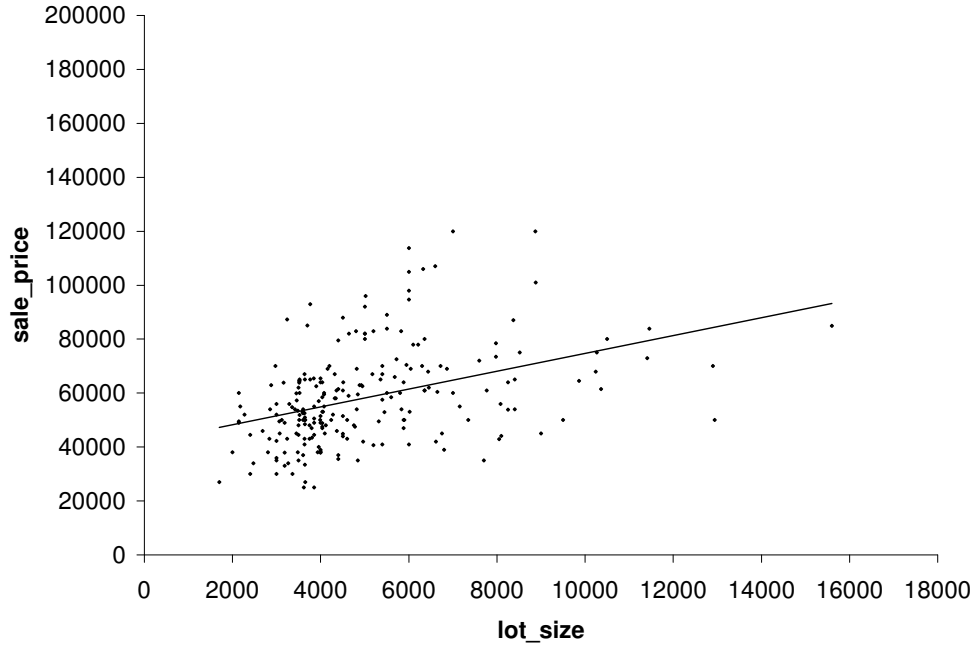
The description quality is $\varphi_{ssd}(D_{21}) > 0.9999$, meaning that the p-value of the test

$$H_0 : \beta_1^{G_2} = \beta_1^{G_2^C} \quad \text{against} \quad H_1 : \beta_1^{G_2} \neq \beta_1^{G_2^C}$$

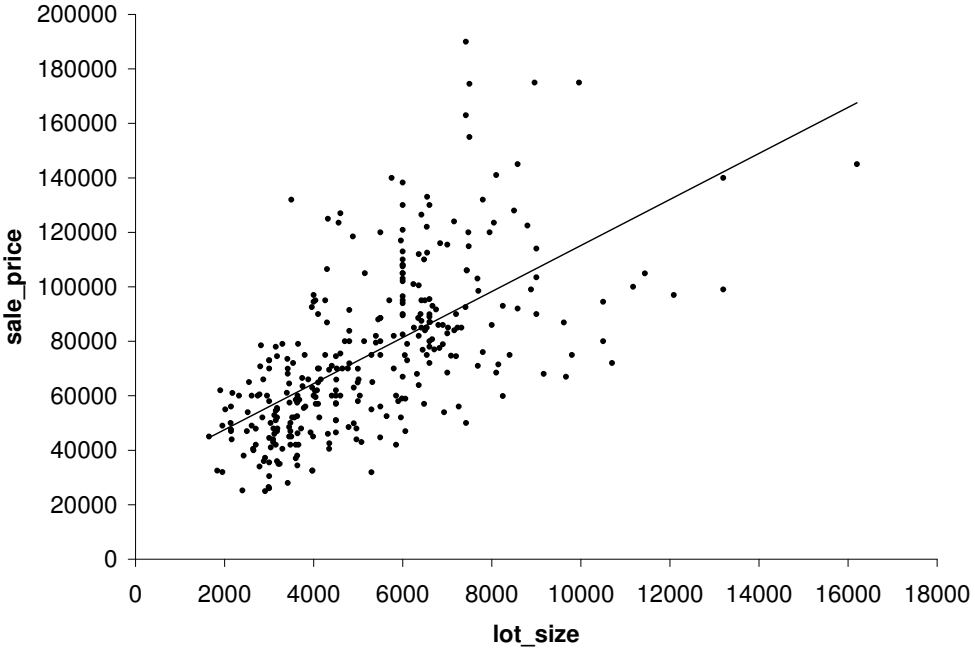
is virtually zero. There are descriptions with a larger difference in slope, but the reported description scores higher because its coverage is quite big. Figure 7.2 shows the scatter plots of *lot_size* and *sales_price* for the description and its complement.

7.5 Conclusions

In this chapter, we propose to use Cook's distance in an Exceptional Model Mining setting. This allows us to find descriptions, for which a regression model fitted on the targets is substantially different from that model for the whole dataset. The use of Cook's distance has two benefits.



(a) $G_{21}, y = 41568 + 3.31 \cdot x$.



(b) $G_{21}^{C}, y = 30723 + 8.45 \cdot x$.

Figure 7.2: *Windsor Housing* - φ_{ssd}: Scatter plot of *lot_size* and *sales_price* for the subgroup corresponding to D_{21} : $drive = 1 \wedge basement = 0 \wedge nbath \leq 1$ and its complement.

On the one hand, Cook's distance has some desirable properties. It is invariant under changes in the scale of a variable, and it explicitly takes the covariance matrix of $\hat{\beta}$ into account. Hence, when using Cook's distance, we need not worry whether the outcome of the EMM algorithm is influenced by the scale ones attributes happen to arrive in (attributes need not be normalised), or the interactions that happen to be present between the regression parameters.

On the other hand, there are some theoretical upper bounds on Cook's distance, that can be computed without actually performing the relatively expensive regression computations. As we have seen, these bounds can only be computed under certain constraints, which correspond to the description covering at least 50% of the records. On the one hand, this means that we can compute the bounds for relatively few descriptions, but on the other hand, whenever we can prune a description, we always prune a relatively expensive regression computation. In future research, we would like to develop bounds for Cook's distance that can be computed for descriptions with small coverage as well.

As we have seen in Section 7.3, the fraction of descriptions that can be pruned is strongly dependent on the complexity of the regression model we fit. We have seen some datasets (Ames Housing and Wine) for which the model complexities are modest, on which we can prune almost 40% and 30% of the descriptions, respectively. On datasets whose model complexities are mediocre, we can still prune approximately 20%, and on the dataset for which the model complexity is high, we can prune only 1%.

In Section 7.2.2 we have discussed some illustrative examples of descriptions found on datasets from different domains. The models fitted on these descriptions are discussed. These examples show the versatility of the problems which EMM with Cook's distance can solve.

Theoretically, the joint influence of records makes Cook's distance for single observations theoretically unsuitable for use in a setting where multiple observations are removed simultaneously. However, it may very well be that this problem is not that serious on real-life datasets. Hence, in future research, we would like to see whether we can use Cook's distance for single observations as a proxy for Cook's distance for multiple observations, for instance by summing over D_i for all $i \in I$.

Also in future work, we would like to explore whether we can improve pruning for complex models. Often one is not interested in the influence of all model coefficients, and at the end of Section 7.1 we have seen an adaptation of Cook's distance such that it is evaluated on a subset of the coefficients. Modifying the bounds accordingly is done in a rather blunt way. We plan to study whether more sophisticated bounds can be derived, with which we can prune more descriptions.

Finally, this chapter was motivated by the Giffen behavior example, in which coefficients not only substantially change in magnitude, but additionally change in sign. Such sign changes can be found on other datasets as well, and the descriptions to which such models are fitted are usually among the most striking deviations we can find. In future work, we would like to develop a quality measure that explicitly seeks for such sign changes.

Chapter 8

Exploiting False Discoveries – Validating Found Descriptions

In Exceptional Model Mining, just like in Subgroup Discovery, we explore a large search space to find subsets of the data that have a relatively high value for a designed or selected quality measure. As we addressed in Chapter 3, the magnitude of the candidate space is potentially exponential in the number of records. Therefore, the process suffers from the multiple comparisons problem [52], which roughly states that, when considering a large number of candidates for a statistical hypothesis, some candidates will inevitably be incorrectly labeled as passing the test. Hence one of the many practical problems in SD/EMM is that it is nontrivial to determine whether discovered descriptions are actual discoveries, or *false discoveries* caused by random artifacts.

In this chapter, we draw upon statistical theory to build a model for false discoveries. Using this model, a number of practical problems can be solved. When applying SD/EMM to a dataset, one is often faced with the nontrivial task of choosing the right parameters for the discovery algorithm, in order to obtain a reasonable collection of results. The problems we intend to address are related to these parameter-setting issues. First of all, with the gradually extending range of quality measures available, for 'classical' Subgroup Discovery [35, 68] but also for non-standard variants such as regression [46, 89] and Exceptional Model Mining [25, 71], the issue of selecting the right measure for the task at hand is often hard. Users of

discovery tools often choose the measure based on their personal familiarity, or simply proceed with the default choice. We aim to provide more objective guidelines for selecting the measure that is most likely to produce interesting and exceptional results, and present empirical results that indicate a partial order amongst quality measures.

A second algorithm-tuning question we intend to address is that of setting a minimum threshold for the selected quality measure. Different measures have different domains, and end-users find it hard to set a reasonable value. Ideally, one would like to choose a minimum quality, such that all descriptions exceeding this value are reliably exceptional, and do not include "random" results that stem from the potentially large search space and the multiple comparisons problem inherent to all discovery methods. In other words, given some desired significance level α (typically 5% to 1%), we would like to obtain the corresponding minimum quality for the measure and dataset in question. As a converse, but very related task, one would like to compute a p-value for each reported description, that indicates to what extent the result is statistically significant.

As mentioned, SD and EMM potentially suffer from the multiple comparison problem. The main contribution of this chapter is the introduction of a method that employs a randomization technique to build a statistical model for the false discoveries caused by the multiple comparisons problem. Using this statistical model, we can refute many insignificant results returned by the discovery algorithm, and thus identify a set of on average more interesting descriptions. Furthermore, we employ the statistical validation to provide an experimental comparison of measures, and propose a suitable choice of measure.

8.1 Problem Statement

As mentioned in the introduction, the main contribution of this chapter is a method that builds a statistical model for false discoveries. This model can be used to solve a plethora of practical problems, of which we will empirically illustrate the following two

1. given a dataset Ω, a quality measure φ and a set $\mathcal{S}$ of descriptions found with this measure through SD/EMM, determine the statistical significance of each description $D \in \mathcal{S}$;
2. given datasets $\Omega_1, \ldots, \Omega_t$, determine which of the given quality measures $\varphi_1, \ldots, \varphi_g$ are better in distinguishing the top-q descriptions found with that measure from a random baseline, for a given q.

8.2 Validation Method

Philosophically speaking, we deal with the multiple comparisons problem (MCP) by a strategy akin to the informal saying "if you can't beat 'em, join 'em". The MCP is caused by random artifacts generating false discoveries. Our plan is to manage this process, generating some false discoveries *of our own*. Then, we can put these *artificial false discoveries* to work, to distinguish the true from the false discoveries in real SD/EMM results.

Informally speaking, to solve problem 1 from the previous section, we perform the following actions. First, we generate a number of false discoveries, by running SD/EMM on datasets obtained using a randomization technique. Then, we build a global model over the qualities of these false discoveries. Finally, we compare the qualities of the descriptions found on the original dataset with this global model. Descriptions that perform substantially better than the model for artificial false discoveries, are deemed likely to be true discoveries.

To solve problem 2 from the previous section, we perform these preceding actions for each quality measure. Then we compare the significance assessments of the top-q descriptions between the quality measures φ_i, distilling a preferential ranking of the quality measures for a dataset Ω_j. We do this for all datasets $\Omega_1, \ldots, \Omega_t$, which allows us to draw conclusions on whether quality measures are consistently preferential over one another.

The solutions to the two problems can be incorporated into one encompassing validation method, consisting of three consecutive steps. Specific choices for each of these steps will be thoroughly examined in the subsequent three sections. For now, we express the validation method in the following, somewhat formal manner

Validation Method. *Suppose a dataset Ω, quality measures $\varphi_1, \ldots, \varphi_g$, and sets of descriptions $\mathcal{S}_1, \ldots, \mathcal{S}_g$ where $\forall_{i=1}^{g} : \mathcal{S}_i$ is found through SD/EMM using quality measure φ_i. The method consists of the following steps*

I. $\forall_{i=1}^{g}$: *use a randomization technique to generate baseline descriptions $B_1^i, \ldots, B_x^i \subseteq \Omega$ for arbitrarily large x;*

II. $\forall_{i=1}^{g}$: *build a statistical model for false discoveries based on the qualities $\varphi_i\left(B_1^i\right), \ldots, \varphi_i\left(B_x^i\right)$ of the baseline descriptions. Then determine for each $D \in \mathcal{S}_i$ how much $\varphi_i(D)$ deviates from the model;*

III. choose any positive integer q, and determine preference between the quality measures by comparing the deviations corresponding to the top-q descriptions in $\mathcal{S}_i$.

In order to solve problem 1, we need steps I and II of the method, for $g = 1$ (since there is only one quality measure). Solving problem 2 requires taking all three steps, and repeating them for each dataset $\Omega_1, \ldots, \Omega_t$.

Since we determine the statistical soundness of quality measures in terms of their ability to deviate from a random baseline, we could interpret this as a test to what extent a quality measure is also a measure for exceptionality. Notice that our method does not consider the coherence of a set $\mathcal{S}$ of descriptions: we do not solve the problem of redundancy within such a set, we do not solve the problem of selecting a small subset of jointly interesting descriptions in $\mathcal{S}$, we merely consider for every single description in $\mathcal{S}$ the likelihood that it is deemed interesting because of the multiple comparisons problem.

8.2.1 Randomization Techniques

There are several randomization techniques we can use to generate the baseline descriptions $B_1^i, \ldots, B_x^i$ (i.e. perform step I of the method). We will employ the randomization technique that is currently the most popular in data mining: swap randomization. Gionis et al. have published a paper detailing its use in data mining [43]. In its most radical form for zero-one matrices, swap randomization shuffles the elements of the dataset in such a way that all row and column sums are kept intact, which is what

the authors of [43] have done for tests involving itemset mining. Swap randomization is also frequently used for validating classifiers in a more moderate form: only the column containing the class labels is replaced by a random permutation of itself. For SD/EMM, it seems reasonable to use this moderate form of swap randomization: each target column is shuffled, independently (cf. Section 8.6) from the other target columns.

We generate the baseline descriptions in the following way. For each B_j^i to be generated, we create a swap-randomized version of the data, by keeping all descriptors intact but applying a random permutation to each target column. Then we run our SD/EMM algorithm on the resulting dataset using quality measure φ_i, and let B_j^i be the best description found.

The rationale behind this process is that by swap-randomizing each target column, we keep its distribution intact. However, we randomize all dependencies between the target columns and the descriptors, and we randomize all internal dependencies between the target columns. Hence the best description found on the swap-randomized data represents the best-quality discovery made while there is no connection between target column and other attributes, and no connection between multiple target columns, apart from connections caused by random artifacts. In other words, this best description represents a false discovery, and its quality is among the highest qualities a false discovery can have.

Another reason why a description found on the swap-randomized data is a good representation of a false discovery is the fact that its discovery has resulted from the same search process as employed while discovering actual descriptions on the original dataset. Alternatively one could easily choose a method to directly generate some random baseline description for use in step I of our method. However, a description found on swap-randomized data goes through the same motions of the SD/EMM algorithm as the actual descriptions found on the original dataset, i.e. the same hypothesis space is traversed, the traversal is performed in the same way, and the search is bounded by the same constraints. Hence the generated false discovery can reasonably be considered a false discovery of the search process.

8.2.2 Building a Statistical Model

When we have generated the baseline descriptions, there are plenty of ways to build a statistical model from them (i.e. perform step II of the method). The most straightforward technique, and the simplest in terms of statistical interpretability, is a direct application of the central limit theorem (CLT) [75]. Under the assumption that x (the number of baseline descriptions) is sufficiently large, according to the central limit theorem, the mean of $\varphi_i\left(B_1^i\right), \ldots, \varphi_i\left(B_x^i\right)$ follows a normal distribution, since these are independent and identically distributed random variables. We use the sample mean ($\hat{\mu}$) and sample standard deviation ($\hat{\sigma}$) as distribution parameters, as suggested by the method of moments [88]. We call this distribution, $\mathcal{N}(\hat{\mu}, \hat{\sigma})$, the *Distribution of False Discoveries* (DFD). Let $D \in \mathcal{S}$ be a description under consideration. We can now formulate the null hypothesis

$$H_0 : \varphi(D) \text{ is generated by the DFD}$$

We can compute a p-value corresponding to this null hypothesis for each $D \in \mathcal{S}$, and this p-value gives us the deviation required as result of step II of the method.

Notice that although the null hypothesis is fixed, its interpretation may vary depending on the randomization technique employed in step I of the method.

Using the DFD, we can not only validate a found description, but also compute threshold values for the quality measure at given significance levels, prior to the actual mining run. Such a threshold could be used as lower bound on the quality of a description in the SD/EMM process. This is a nontrivial contribution to the process, since it is generally not easy for an end-user to set a sensible lower bound for any given quality measure. Additionally, such sensible values for a lower bound depend heavily on the dataset at hand. Until now, it was common to use a default value for such a lower bound by lack of a better method; the DFD gives us more sensible threshold values.

8.2.3 Comparing Quality Measures

For performing step III, comparing the relative performance of the quality measures, we use a technique recently described by Demšar in an article [19] on statistical comparisons of classifiers over multiple datasets. First a Friedman test [31, 32] is performed to determine whether the quality measures all perform equivalently. This is a non-parametric version of the repeated-measures ANOVA. For each test case the quality measures are ranked by their performance; in case of ties we assign average ranks. Let r_i denote the average rank over all test cases for quality measure φ_i, $\forall_{i\in\{1,\ldots,g\}}$, and let T denote the number of test cases. The null hypothesis states that all measures perform similarly, hence their average ranks should be equal. Under this null hypothesis, the Friedman statistic

$$\chi_F^2 = \frac{12T}{g(g+1)} \cdot \sum_i \left(r_i - \frac{g+1}{2} \right)^2$$

follows a chi-squared distribution with $g-1$ degrees of freedom.[1]

If the null hypothesis of the Friedman test is rejected, we can determine which quality measures are significantly better than others with a post-hoc test. Following Demšar's proposal, we use the Nemenyi test [83], which is similar to the Tukey test for ANOVA. In this test a critical difference (CD) is computed

$$CD = q_\alpha \sqrt{\frac{g(g+1)}{6T}}$$

where the critical values q_α are based on the Studentized range statistic [94, pp. 451–452] divided by $\sqrt{2}$. If the difference between the average ranks of two quality measures surpasses this CD, then the better-ranked measure performs significantly better.

8.3 Experiments

To illustrate how our method performs, we run Subgroup Discovery experiments on several datasets. The bulk of our empirical evaluation will

[1] Careful readers may notice that this formula is not the one given by Demšar. It is, however, the one given by Friedman himself. Equivalence can be shown in four lines of math; the equation shown here is slightly easier to compute, and easier on the eye.

Table 8.1: UCI datasets used for the DFD experiments.

	Dataset	N	# descriptors		$\|\ell\|$
			discrete	numeric	
1.	*Adult*	48842	8	6	2
2.	*Balance-scale*	625	0	4	3
3.	*Car*	1728	6	0	4
4.	*CMC*	1473	7	2	3
5.	*Contact-lenses*	24	4	0	3
6.	*Credit-a*	690	9	6	2
7.	*Dermatology*	366	33	1	6
8.	*Glass*	214	0	9	6
9.	*Haberman*	306	1	2	2
10.	*Hayes-roth*	132	0	4	3
11.	*Ionosphere*	351	0	34	2
12.	*Iris*	150	0	4	3
13.	*Labor*	57	8	8	2
14.	*Mushroom*	8124	22	0	2
15.	*Pima-indians*	768	0	8	2
16.	*Soybean*	683	35	0	19
17.	*Tic-tac-toe*	958	9	0	2
18.	*Wisconsin*	699	0	9	2
19.	*Yeast*	1484	1	7	10
20.	*Zoo*	101	16	1	7

be done in terms of Subgroup Discovery with only one discrete target, but this is by no means essential to the method. In fact, it can be applied to any supervised Local Pattern Mining technique. We will briefly illustrate this by applying our method not only to traditional Subgroup Discovery, but also to the instance of Exceptional Model Mining with the correlation model class, as introduced in Chapter 4. Results of the experiments on traditional SD are described in Sections 8.3.1 and 8.3.2, and results on the EMM instance in Section 8.3.3.

We pick the following parameters for the beam search process. On each level, we select the $w = 25$ best descriptions, and refine these to create the candidate descriptions for the next level. To bound the complexity of the descriptions we use a search depth of $d = 3$. We let $minsup = \lfloor \frac{N}{10} \rfloor$, i.e. a description must be covered by at least 10% of the dataset. These parameter settings are somewhat arbitrary; we believe that this is not really relevant for the purpose of demonstrating our new method.

The 20 datasets we have used for our tests with traditional SD, can be found in the UCI Machine Learning Repository [3]. Table 8.1 contains details on the datasets considered. Here, $|\ell|$ denotes the number of distinct target values in the dataset. Notice that, rather unfortunately, this table features a second dataset named *Yeast*, after the one introduced in Table 6.1. The dataset considered in this chapter is the *Yeast* dataset that can be found through UCI [3], and not the *Yeast* dataset that was introduced in the paper by Elisseeff et al. [28].

Before experimenting with the method, let us empirically investigate a simpler solution: *empirical* p*-values*. Given a description D to be validated, one could simply assign as p-value the fraction of randomly generated results that outperform D. Though valid as a validation method for single descriptions, the empirical p-values lack the expressive power necessary to perform step III of our validation method, which is required when we want to validate quality measures. This is illustrated by the histogram (represented by the jagged line), displayed in Figure 8.1, of qualities of 1000 random subsets on the *CMC* dataset with target value *'no-use'*, normalized into Z-space (i.e. a subset has value one on the x-axis in the histogram when its quality is one standard deviation higher than the sample mean). The figure also contains our CLT-based normal distribution fitted to the

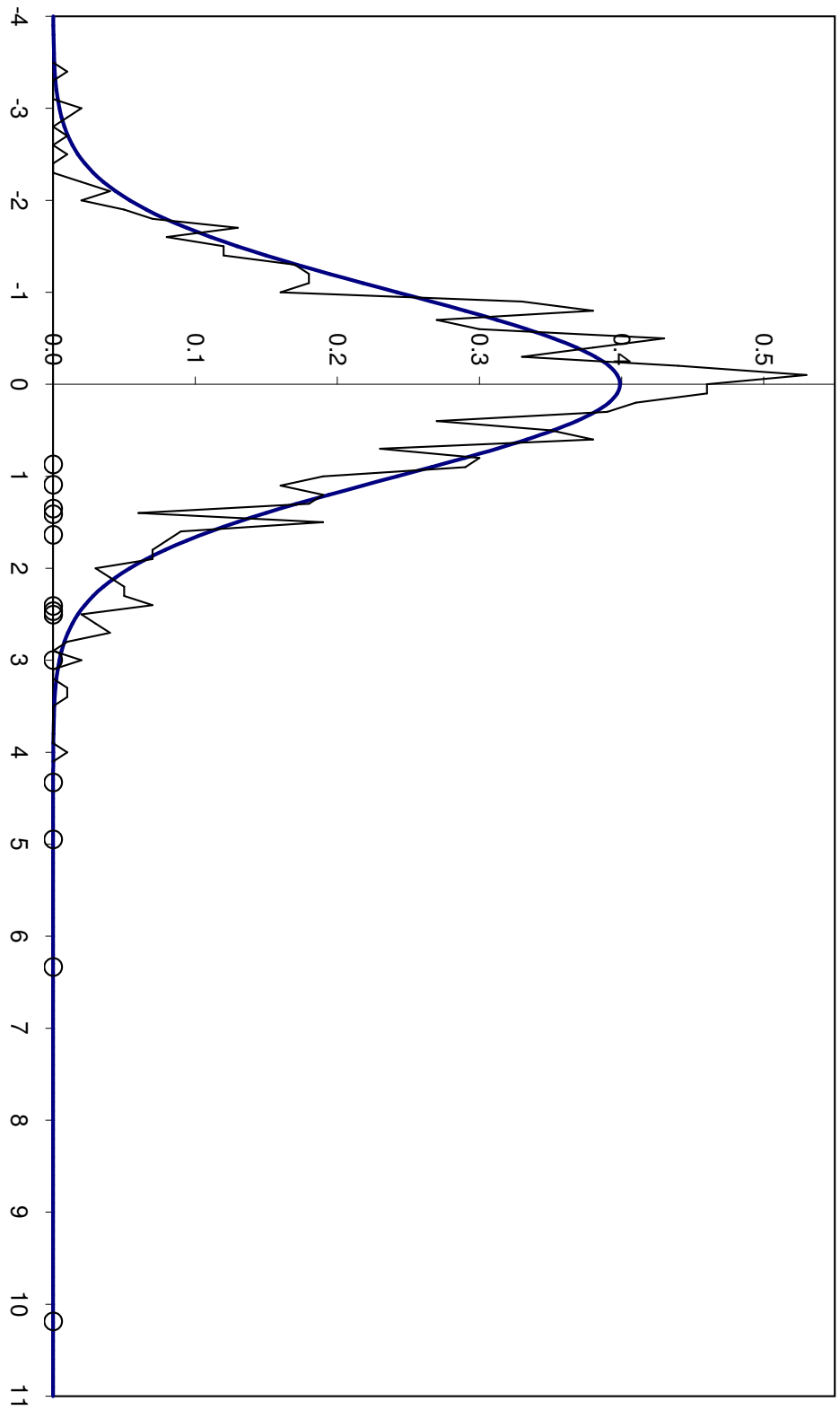


Figure 8.1: CLT-based model versus empirical p-values.

random qualities (represented by the smooth curve) and the 13 descriptions (represented by circles on the x-axis) found using a very shallow search of $d = 1$. The rightmost nonzero value of the histogram occurs at $x = 4$, hence all descriptions to the right of that point are indistinguishable by empirical p-values. The normal distribution never becomes zero, hence does not suffer from this problem. Hence, with the CLT-based model we can still compare the relative significance of the 4 best descriptions, whereas with empirical p-values we cannot.

8.3.1 Validating Descriptions

We will first illustrate how to use our method to solve problem 1 from Section 8.1: validating single descriptions. To this end, we only need the method's first two steps.

We consider just one quality measure: Weighted Relative Accuracy (WRAcc) [55], arguably the most popular quality measure in Subgroup Discovery. For each dataset, we perform an SD run for each target value, and report the 1000 best descriptions. We then run the first two steps of our method to determine how many of the found descriptions remain if insignificant descriptions are removed. We report the average fraction of descriptions that is retained per dataset for different significance levels in Table 8.2.

As stated in Section 8.2, one could also use the Distribution of False Discoveries to determine quality measure thresholds for given significance levels, a common practical issue with SD/EMM exercises. We illustrate this by determining thresholds on the *Contact-lenses* dataset with target value *'none'*. Notice that WRAcc can theoretically assume values between -0.25 and 0.25. We find that with significance level $\alpha = 10\%$ a subgroup needs to have a WRAcc of at least 0.054 to reject the null hypothesis that it is a false discovery, with $\alpha = 5\%$ a subgroup needs to have a WRAcc of at least 0.068, and with $\alpha = 1\%$ a value of at least 0.093. To provide context: the best description found on this dataset with this target value has a WRAcc of 0.188.

Table 8.2: Fraction of descriptions retained when removing insignificant descriptions.

Dataset	$\alpha = 10\%$	$\alpha = 5\%$	$\alpha = 1\%$
Adult	1.000	1.000	1.000
Balance-scale	0.561	0.554	0.548
Car	0.650	0.591	0.518
CMC	0.506	0.484	0.445
Contact-lenses	0.069	0.069	0.052
Credit-a	1.000	1.000	1.000
Dermatology	0.838	0.808	0.761
Glass	0.738	0.675	0.562
Haberman	0.427	0.392	0.327
Hayes-roth	0.388	0.313	0.210
Ionosphere	1.000	1.000	1.000
Iris	0.902	0.879	0.834
Labor	0.628	0.567	0.401
Mushroom	0.967	0.966	0.964
Pima-indians	1.000	1.000	1.000
Soybean	0.724	0.713	0.689
Tic-tac-toe	0.493	0.446	0.311
Wisconsin	1.000	1.000	1.000
Yeast	0.687	0.673	0.647
Zoo	0.600	0.582	0.524

8.3.2 Validating Quality Measures

We can build on the instantiation of our model that we used in the previous section to solve problem 2 from Section 8.1: validating quality measures. We select 12 quality measures for single discrete targets that are quite common in Subgroup Discovery, and test them against each other. The measures are WRAcc, |WRAcc|, χ^2, Confidence, Purity, Jaccard, Specificity, Sensitivity, Laplace, F-measure, G-measure, and Correlation. Details on these measures and their origins can be found in the paper by Fürnkranz and Flach [35].

Table 8.3: Average ranks of the quality measures.

	All datasets		Binary target	
Measure	$q = 1$	$q = 100$	$q = 1$	$q = 100$
χ^2	4.435	4.038	4.694	3.889
Jaccard	5.224	5.622	5.361	7.028
Correlation	5.235	4.679	5.361	4.667
|WRAcc|	5.288	4.571	5.306	4.333
G-measure	5.312	5.538	5.417	6.750
F-measure	5.582	5.718	5.250	6.778
WRAcc	5.800	5.027	5.417	4.722
Confidence	6.506	6.865	7.333	7.028
Laplace	6.553	6.654	7.278	6.139
Specificity	7.465	8.455	8.306	7.806
Purity	10.235	10.141	8.389	7.361
Sensitivity	10.365	10.692	9.889	11.500
χ^2_F $(\alpha = 1\%)$	261.916	292.001	40.674	57.618
CD $(\alpha = 1\%)$	2.069	2.160	4.496	4.496

For each dataset, we perform steps I and II of our method in the same way as in the previous section, with each of the 12 quality measures. We then compare the measures in step III by comparing the p-values of the q best descriptions, for both $q = 1$ and $q = 100$ (for $q = 100$ we take the average p-values over the top-100 descriptions). Hence for all measures we obtain for both choices of q one test score for each combination of dataset and target value within that dataset. For $q = 1$ this leads to a grand total of 85 test scores for each quality measure. On both the *Car* and the *Contact-lenses* dataset, no 100 descriptions are found that satisfy the *minsup* constraint. Hence there are no results on these datasets for $q = 100$, leaving a total of 78 test cases for $q = 100$.

The measures are subsequently ranked, where a lower test score (p-value) is better. The resulting average ranks can be found in the second and third columns of Table 8.3. This table also displays the results of the Friedman tests, the values for χ^2_F. With a significance level of $\alpha = 1\%$ we need χ^2_F to be at least 24.73 to reject the null hypothesis that all quality measures perform equally well. Hence we comfortably pass this test.

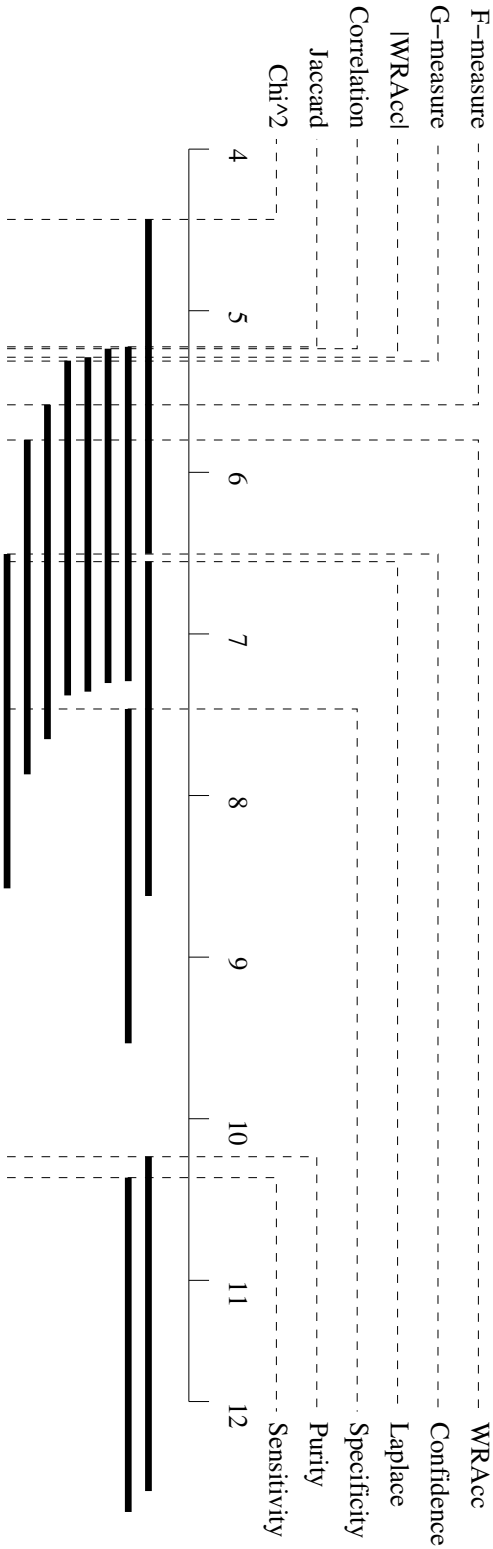


Figure 8.2: CD chart for $q = 1$ (CD $= 2.069$).

Since the Friedman test is passed, we can now perform Nemenyi tests to see which quality measures outperform others. For the $q = 1$ setting, the critical difference CD equals 2.069 with significance level $\alpha = 1\%$. For each pair of measures we compute from Table 8.3 whether their difference is larger than CD, and if so, the one with the smaller average rank is better than the other. The corresponding CD chart [19] can be found in Figure 8.2. Such a chart features a horizontal bar of length CD for each quality measure φ_i, starting at its average rank. Hence φ_i is significantly better than each quality measure whose bar starts to the right of the bar of φ_i. For instance, in Figure 8.2 we see that χ^2 is significantly better than Laplace, Specificity, Purity, and Sensitivity. Figure 8.3 displays the CD chart for the $q = 100$ setting.

When we have a dataset with many distinct target values, we repeatedly let one of the target values correspond to positive examples and the rest to negative examples. Hence the more distinct target values we have, the lower the average fraction of positive examples in the dataset. To see whether certain quality measures suffer from this effect, we also computed the average ranks considering only the 9 datasets with a binary target. The results can be found in the last two columns of Table 8.3. Again, the average ranks easily pass the Friedman test. Now that we have only 18 test cases, the critical difference for the Nemenyi test becomes $CD = 4.496$ with significance level $\alpha = 1\%$.

8.3.3 Validating EMM Results

So far we have illustrated our method with measures for Subgroup Discovery over a single discrete target. We now turn to the variant of EMM with the correlation between two targets as model class, as introduced in Chapter 4. In that chapter, we introduced three quality measures for the problem: φ_{abs}, φ_{ent}, and φ_{scd}. We validate these measures, together with some simpler measures that do not explicitly compare multiple models, but rather simply search for descriptions maximizing one particular quantity. For such quantities we consider a high positive correlation, by maximizing $\hat{r}$, a high negative correlation, by maximizing $-\hat{r}$, a high positive or negative correlation, by maximizing $\hat{r}^2$, and a near-zero correlation, by maximizing $-\hat{r}^2$. Notice that these four measures do not take the correlation on the

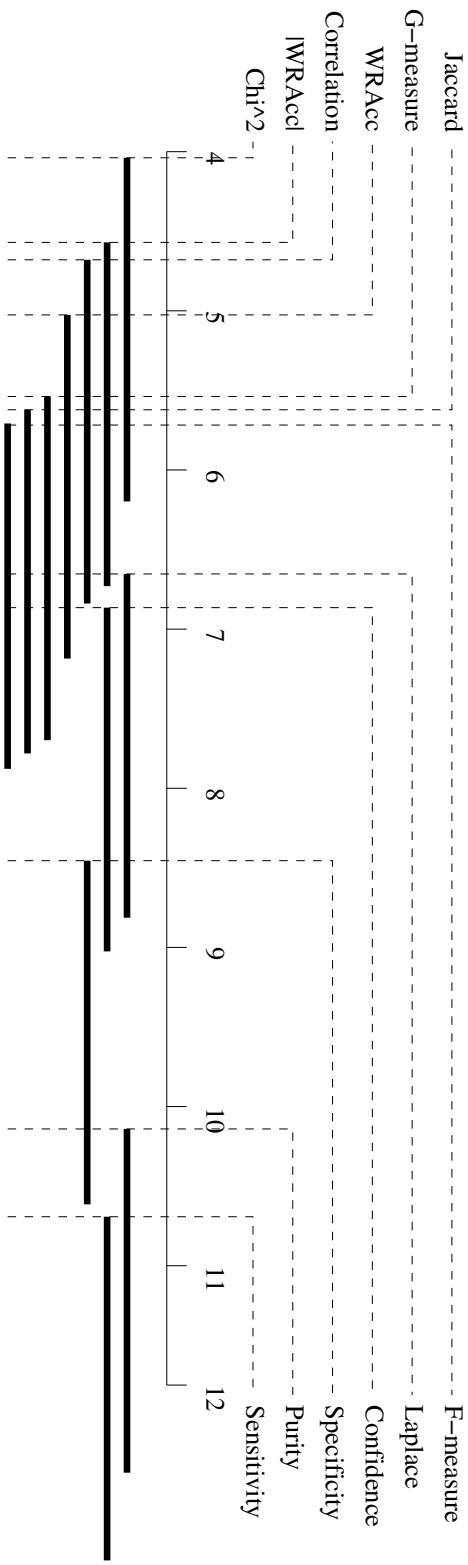


Figure 8.3: CD chart for $q = 100$ (CD $= 2.160$).

Table 8.4: Average ranks for correlation model measures.

Measure	Average rank
φ_{ent}	1.75
$\hat{r}$	3.00
$\hat{r}^2$	3.75
φ_{abs}	4.25
$-\hat{r}$	4.75
$-\hat{r}^2$	5.25
φ_{scd}	5.25
χ^2_F	21.96
CD $(\alpha = 1\%)$	4.114

whole dataset or the correlation on the complement into account. Hence they do not comply with the EMM quality measure design guidelines we outlined in Section 3.2.

We validate the 7 measures for this setting on the datasets and target concepts introduced in Table 4.1. The resulting average ranks over the 2 datasets — *Windsor Housing* and *Affymetrix* — can be found in Table 8.4. The Friedman test value for these ranks is $\chi^2_F = 21.96$, where 16.81 would be enough with 7 measures, so we can proceed with the Nemenyi test. The critical difference is CD $= 4.114$ with significance level $\alpha = 10\%$ when testing 7 measures on 4 test cases (aggregating over the results for $q = 1$ and $q = 100$). In these modest experiments, we find that no significant conclusions can be drawn.

8.4 Discussion

The previous section summarized the results experimentally obtained with our new method; in this section we will interpret them. We start with the results obtained by the technique for validating descriptions in a set $\mathcal{S}$ found through SD/EMM.

8.4.1 Validating Descriptions

From Table 8.2 we find that we cannot refute any descriptions from $\mathcal{S}$ in several datasets: *Adult*, *Credit-a*, *Ionosphere*, *Pima-indians*, and *Wisconsin*. To explain this result, we craft a metalearning dataset from Tables 8.1 and 8.2. We select the columns from Table 8.1 as descriptors of our metalearning dataset, and add three new columns, representing the total number of descriptors in the dataset, a boolean column representing whether the dataset has discrete descriptors, and a boolean column representing whether the dataset has numeric descriptors. As target column we add the last column of Table 8.2: the fraction of descriptions retained when insignificant descriptions are removed, with significance level $\alpha = 1\%$. On this metalearning dataset we perform a shallow (using search depth $d = 1$) but exhaustive Subgroup Discovery run, using Klösgen's mean test [55] as quality measure. The resulting metadescriptions should consist of those datasets with a relatively high fraction of kept descriptions.

The best metadescription is defined by the condition that the datasets have more than five numeric descriptors. The eight datasets covered by this metadescription are *Adult*, *Credit-a*, *Glass*, *Ionosphere*, *Labor*, *Pima-indians*, *Wisconsin*, and *Yeast*. This set encompasses all datasets for which we cannot refute any of the descriptions from $\mathcal{S}$. This makes sense, since for each dataset we have only considered the top-1000 descriptions, a fixed number independent of dataset characteristics. Numeric descriptors usually have many different values, resulting in a hypothesis space that is much larger than it would have been if the descriptors were discrete. Hence in datasets with relatively many numeric descriptors, it is more likely that the 1000 best descriptions represent relatively rare spikes in a quality distribution consisting mainly of low values. Therefore it is less likely that the random baseline incorporates some of these spikes, and thus the baseline is more likely to be relatively weak.

8.4.2 Validating Quality Measures

The results we obtained by the technique for validating quality measures show that χ^2 achieves the best performance of all quality measures in distinguishing the top-q descriptions from false discoveries. Many of the relations

between quality measures, however, are not significant. For $q = 1$, all other quality measures perform significantly better than Purity and Sensitivity. Additionally, Specificity performs significantly worse than Jaccard, Correlation, |WRAcc|, and the G-measure, and χ^2 significantly outperforms Laplace.

For $q = 100$, we see some slight changes: χ^2 and |WRAcc| now also perform significantly better than Confidence, and Specificity is now additionally outperformed by the F-measure and WRAcc while it no longer performs significantly better than Purity. Finally, Correlation significantly outperforms Confidence. Obviously, some measures might be better than others in distinguishing the top-q descriptions from false discoveries when $q = 1$, while others might be better when $q = 100$. The observed changes are not very dramatic, and we consider the selection of q a user-derived parameter in the method.

One of the significant relations that seems somewhat peculiar, is the result that for both $q = 1$ and $q = 100$, Confidence performs significantly better than Purity, while the latter is defined to be $\max\{\text{Confidence}, 1 - \text{Confidence}\}$. While there may be a good theoretical reason to consider the Purity of a description, we can see from the definition that Purity has a lower bound of 0.5, hence the random baseline will generate higher values with Purity than with Confidence. Apparently the quality of the descriptions found with Purity does not increase enough compared to those found with Confidence to compensate for this effect.

By comparing the second and third columns of Table 8.3 with the last two columns, we can see that |WRAcc|, WRAcc, and particularly Purity perform better when we restrict the tests to datasets with a binary target. These measures benefit from the fact that in these test cases we have a better balance between positive and negative examples in the data, compared to test cases on other datasets. We can also read from the table that we have fewer measures that are significantly better than others on datasets with a binary target. This is mainly because significance is hard to achieve in an experiment with only 18 test cases as opposed to 85 or 78 on all datasets. With 18 test cases, the critical difference for the Nemenyi test with significance level $\alpha = 1\%$ is $CD = 4.496$, rather than $CD = 2.069$ with 85 test cases. Since the average ranks range from 1 to 12, a critical

difference of 4.496 is substantial. More significant difference relations between the quality measures can be expected when experiments would be performed on more datasets with a binary target.

8.4.3 Validating EMM Results

The results for the EMM variant were generated on a modest number of test cases. As a result, the critical difference for the Nemenyi test is quite high, and one could not expect to find many significant results. Extensive experimentation may give a significant reason to prefer one measure over another in this setting. For now, what matters is that this illustrates that our method is applicable in more general settings than just traditional Subgroup Discovery.

Another noteworthy result, albeit non-significant, is the empirical illustration that a difference quantification is not always enough to make a good quality measure for EMM, as announced in Section 3.2.2. We had stated that deviations from the norm are easily achieved in small subsets of the data. Hence, quality measures that evaluate merely on the difference in model characteristics, such as φ_{abs}, will favor relatively small descriptions. It is relatively easy to find a small description with a high value for such a quality measure *on swap-randomized data as well*. This effect can be mitigated by incorporating an entropy term in the quality measure, as we have done with φ_{ent}. Recall that $\varphi_{\text{ent}} = \varphi_{\text{abs}} \cdot \varphi_{\text{ef}}$. As can be seen in Table 8.4, the incorporation of the entropy term can lift a quality measure with mediocre performance (φ_{abs}) in distinguishing real from false discoveries, to top-level performance (φ_{ent}).

8.5 Related Work

Statistical validation specifically tailored for Subgroup Discovery barely exists. Fortunately, many techniques for statistical validation in local pattern mining settings, which have been developed ever since association rules were invented, are applicable in Subgroup Discovery. Most of the recent approaches employ empirical p-values, as shortly introduced in Sec-

tion 8.3. This method has been applied in papers concerning significant query results on multi-relational databases [85] and swap randomization on high-dimensional 0/1 datasets [43]. In many circumstances, the use of empirical p-values is very appropriate. However, we attempt to validate descriptions with a high quality by comparing them to random descriptions that are expected to have a more moderate quality. Since we are trying to validate outliers in the quality measure distribution, in many cases we will find empirical p-values to be zero for many measures, hence they are not very useful for comparing the measures with each other.

A method that assigns nonempirical p-values to single association rules has been proposed by Megiddo and Srikant [79]. They use random approximation techniques to assign significance to single association rules and sets of associations. Unfortunately, their choice of underlying distribution is not motivated in any way.

Quality measures exist for Subgroup Discovery that directly implement a statistical significance test. For instance, one can show that Klösgen's mean test $\left(\sqrt{n}\,(p - p_0)\right)$ [55] is order-equivalent to a t-test. Also well suited for Subgroup Discovery is the chi-squared $\left(\chi^2\right)$ measure [98], originally defined for association rules. While such quality measures automatically statistically validate single descriptions, their application in Subgroup Discovery and hence use in a vast search space will invariably suffer from the multiple comparison problem, and hence the results will fall prey to the problem we attempt to solve in this chapter.

Tan et al. have developed a method [101] to compare quality measures on contingency tables by intrinsic properties. The results this method delivers are somewhat inconclusive, hence the method relies on experts to decide which measure is to be preferred. Also, the method seems not to be extendable beyond k-way contingency tables.

Finally, Webb devised a procedure to assign significance to individual descriptions [112]. He gives two different ways to perform a Bonferroni-style adjustment to the significance level: direct adjustment, and an approach that is very similar to the train-and-test-set procedure known from the determination of the predictive accuracy of a classifier. As is typical for Bonferroni correction, the adjustments may be a bit too strict. This especially holds when the search space becomes very large, for instance when

dealing with numeric descriptors. When applying a Bonferroni correction one assumes that the different hypotheses are independent, which in a Subgroup Discovery setting is not the case, leading to too strict adjustments to the significance level. Also, rather than being a method that assigns significance to descriptions, Webb's work is more a framework that can be used with any statistical hypothesis test.

8.6 Conclusions

We propose a method that deals with the multiple comparisons problem in SD/EMM, i.e. the problem that when exploring a vast search space one basically considers many candidates for a statistical hypothesis, hence one will inevitably incorrectly label some candidates as passing the test. Our method tackles this problem by building a statistical model for the false discoveries: the *Distribution of False Discoveries* (DFD). This distribution is generated by, given a dataset and quality measure, repeatedly running an SD/EMM algorithm on a swap-randomized version of the data. In this swap-randomized version, while the distribution of each target is maintained, the correlations with the descriptors and the correlations between targets are destroyed. Hence the best description discovered on this dataset represents a false discovery. The DFD is then determined by applying the central limit theorem to the qualities of these false discoveries.

Having determined the DFD, one can solve many practical problems prevalent in SD/EMM. For any discovered description, one can determine a p-value corresponding to the null hypothesis that it is generated by the DFD; refuting this null hypothesis implies that the description is not a false discovery. Given a set of quality measures, one can use the DFD to determine which quality measures are better than others in distinguishing the top-q descriptions from false discoveries. This gives an objective criterion for selecting a quality measure that is more likely to produce exceptional results. Finally, given some desired significance level α, one could extract from the DFD a minimum threshold for the quality measure at hand.

When validating single descriptions, we see that our method removes insignificant descriptions found on datasets that have few numeric descriptors. From metalearning we find that on large datasets, for instance with

more than five numeric descriptors, the random baseline is more likely to accept many descriptions. This is reasonable because of the associated larger hypothesis space. Table 8.2 shows that our method can remove insignificant descriptions on some of the datasets with more than five numeric descriptors, but not on all of them.

When we validate quality measures, we have outlined that the method we described determines the extent to which a quality measure is also an exceptionality measure. We have seen that of the twelve measures for Subgroup Discovery we tested, χ^2 is the best exceptionality measure, and Purity and Sensitivity are by far the worst. For the EMM correlation model variant no significant conclusions can be drawn from the modest experiments.

In this chapter we have presented a technique making extensive use of swap randomization. Notice that we do not by any means claim to have invented this particular randomization method. Also, its use in step I of the method we introduced in this chapter is not the only option available. We have extensively explained why using swap-randomized data leads to a good model for false discoveries, but it comes at a price: for every result of an SD/EMM run one wishes to validate, one has to run the same SD/EMM algorithm an additional x times, where x needs to be large enough to satisfy the constraints of the Central Limit Theorem. In the more traditional Subgroup Discovery setting, one can usually afford this extra computation time. For more complex settings, for instance the EMM variant using Bayesian networks introduced in Chapter 6, this becomes problematic. When computation time becomes an issue, one might consider different randomization techniques to generate $B_1, \ldots, B_x$, for instance by simply drawing a random sample from Ω of a certain size for each B_i. Before such a technique can be employed, its theoretical ramifications need to be explored. In future work, we also plan to empirically investigate the effect of certain parameters on the outcome of the method.

Another randomization-related point that is worthy of further investigation, is induced by the general applicability of the validation method beyond traditional Subgroup Discovery. Most of the experiments run in this chapter concern SD. In this setting, the exact implementation of swap-randomizing the target is clear-cut, as well as its philosophical implica-

tions: there is only one target to be permuted, and doing this breaks all connections between target and descriptors while keeping the target distribution intact. By contrast, in Exceptional Model Mining, there are multiple targets. Swap-randomizing these targets can be done in two straightforward ways, whose philosophical implications are unclear. In Section 8.2.1, we outlined how each target column is permuted *independently* from the other target columns. This ensures that connections between targets and descriptors are broken, while keeping the *marginal* distribution of each target intact. However, the *joint* distribution over the targets is broken. This last effect can be prevented by the design choice to swap-randomize the targets not independently, but jointly: we generate only one permutation, and apply that same permutation to every target column.

The goal of EMM is to measure unusual interactions between several targets. Hence, on first glance, it seems preferable to maintain the joint distribution over the targets when swap-randomizing. However, more specifically, EMM strives to find *subgroups* coinciding with unusual target interactions, where these interactions are gauged in terms of some kind of *modeling* over the targets. The subgroups are based on coherent descriptions: conditions on a few descriptive attributes of the dataset. Depending on the model class under consideration, the descriptive attributes may be representing latent variables that actually should have been present (for instance as a dummy variable) in our model. In fact, the subsequent chapter concerns an application of representing found subgroups in a global model. In this light, it becomes unclear to assess whether it still makes sense to simultaneously *maintain* the internal connections between the targets, and *break* the connections between targets and descriptors. Though breaking the joint target distribution, swap-randomizing targets independently at least has clear philosophical implications. Further study is needed regarding both the theoretical foundations and empirical consequences of choosing one of these swap randomization variants.

Chapter 9

Multi-label LeGo – Enhancing Multi-label Classifiers with Local Patterns

Contrary to ordinary classification, in multi-label classification (MLC) one can assign more than one class label to each record [106, 107]. For instance, when we have the earth's continents as classes, a news article about the March 2013 election of Pope Francis, who was born in Argentina, could be labeled with the *Europe* and *South America* classes. Originally, the main motivation for the multi-label approach came from the fields of medical diagnosis and text categorization, but nowadays multi-label methods are required by applications as diverse as semantic scene classification [6], protein function classification [28], and music categorization [103].

Many approaches to MLC take a decompositive approach, i.e. they decompose the MLC problem into a series of ordinary classification problems. The formulation of these problems often ignores interdependencies between labels, suggesting that the predictive performance may improve if label dependencies are taken into account. When, for instance, one considers a dataset where each label details the presence or absence of one kind of species in a certain region, the food chains between the species cause a plethora of strong correlations between labels. But interplay between species is more subtle than just correlations between pairs of species, as we have for instance seen in the *Pisaster* example of Chapter 2, where the dependence between *Haliclona* and *Anisodoris* is conditional on the

presence of *Pisaster*. The ability to consider such interplay is an essential element of good multi-label classifiers.

In this chapter we investigate incorporating locally exceptional interactions between labels in MLC, as an instance of the *LeGo framework* [37, 57]. In this framework, the KDD process is split up into several phases. First, local models are found, each representing only part of the data. Then, a subset of these models is selected. Finally, this subset is employed in constructing a global model. The crux is that straightforward classification methods can be used for building a global classifier, if the locally exceptional interactions between labels are represented by attributes constructed from descriptions found in the local modeling phase.

The descriptions representing these locally exceptional interactions are found with the EMM instance from Chapter 6: modeling the conditional dependencies between the labels by a Bayesian network, and striving to find descriptions for which the learned network has a substantially different structure than the network learned on the whole dataset. These descriptions can each be represented by a binary attribute of the data. At the end of this chapter we demonstrate that the integration of these description-based attributes into the classification process improves classifier performance. We also investigate whether the newly generated binary attributes are expressive enough to replace the original descriptive attributes, while maintaining classifier performance and increasing efficiency.

9.1 The LeGo Framework

As mentioned, the work in this chapter relies heavily on the LeGo framework [37, 57]. This framework assumes that the induction process is not executed by running a single learning algorithm, but rather consists of consecutive phases, as illustrated in Figure 9.1. In the first phase, a Local Pattern Mining algorithm is employed in order to obtain a number of informative descriptions, which can serve as attributes to be used in the subsequent phases. These descriptions can be considered partial solutions to local complexities in the data. In the second and third phase, the descriptions are filtered to reduce redundancy, and the selected descriptions are combined in a final global model, which is the outcome of the process.

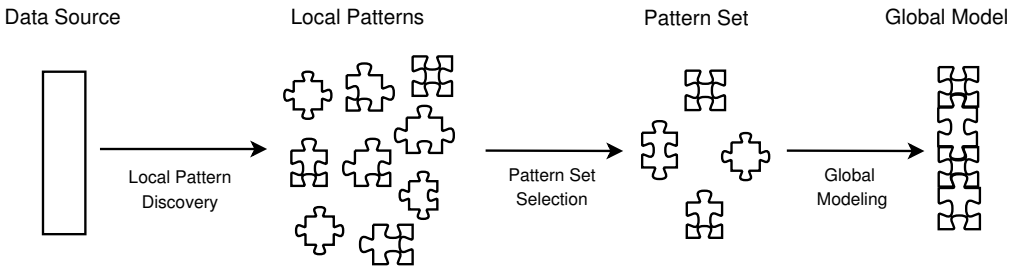


Figure 9.1: The LeGo framework.

The idea of the LeGo framework is that, instead of attacking the induction task in the original representation, we transform it by an automated process to a representation that already resolves a number of complexities in the original task. Then, the new representation can be approached with a standard Global Modeling technique, such as Support Vector Machines (SVMs) with linear kernels, since the potentially hard aspects of the original representation have been accounted for in the new representation by the descriptions. Generally, for this automated process, any of the existing Local Pattern Mining algorithms can be employed, thus benefiting from the wealth of LPM algorithms that has grown over the last decade.

The main reason to invest the additional computational cost of a LeGo approach over a single-step algorithm, is the expected increase in accuracy of the final model, caused by the higher level of exploration involved in the initial Local Pattern Mining phase. Typically, Global Modeling techniques employ some form of greedy search, and in complex tasks, subtle interactions between attributes may be overlooked as a result of this. By contrast, in most Local Pattern Mining methods, extensive consideration of combinations of attributes is quite common. When employing such exploratory algorithms as a form of preprocessing, one can think of the result (the descriptions) as partial solutions to local complexities in the data. The descriptions, which can be interpreted as new virtual attributes, still need to be combined into a global model, but potentially hard aspects of the original representation will have been accounted for. As a result, straightforward methods such as Support Vector Machines with linear kernels can be used in the Global Modeling phase.

The LeGo approach has shown its value in a range of settings [37], particularly regular binary classification [59, 100], but we have reasons for choosing

this approach in the context of multi-label classification (MLC). It is often mentioned that in MLC, one needs to consider potential interactions between the labels, and that simultaneous classification of the labels may benefit from knowledge about such interactions [11, 87, 92, 118].

In the remainder of this chapter, we will identify the targets in EMM with the labels in MLC, hence the terms *the labels* and *the targets* refer to the exact same thing. Similarly, to adhere to the usual terminology in classification, we will let *the attributes* refer to any attribute of the dataset which can be used by the classifier to define its decision boundary on. Hence, an attribute can both be a descriptive attribute from the original dataset, or a constructed attribute built from a description found in the Local Pattern Mining phase. Finally, since we employ commonly known feature selection methods in the Pattern Subset Discovery phase, we will occasionally refer to attributes as *features*. Neither the term "attribute" nor the term "feature" can ever refer to a target/label.

9.2 Multi-label Classification

The task of Multi-Label Classification (MLC) is, given a training set $\mathcal{E} \subseteq \Omega$, to learn a function $f : (a_1, \ldots, a_k) \rightarrow (\ell_1, \ldots, \ell_m)$ which predicts the labels for a given record. Many multi-label learning techniques reduce this problem to ordinary classification, where for a given record exactly one *class* is predicted, rather than multiple labels. We will use each of these techniques for decomposing a multi-label problem into an ordinary classification problem in the third LeGo phase (cf. Section 9.3.3).

The widely used *binary relevance* (BR) [106, 107] approach tackles a multi-label problem by learning a separate classifier $f_i : (a_1, \ldots, a_k) \rightarrow \ell_i$ for each label ℓ_i, as illustrated in Figure 9.2b. At query time, each binary classifier predicts whether its class is relevant for the query record or not, producing a set of relevant labels. Obviously, BR ignores interdependencies between classes since it learns the relevance of each class independently.

One could address this problem with *classifier chains* (CC) [92], which can model label dependencies since they stack the model outputs: the prediction of the model for label ℓ_i depends on the predictions for labels

Attributes			Labels $\in \{0,1\}^m$		
a_1^1,	$\ldots$,	a_k^1	ℓ_1^1,	$\ldots$,	ℓ_m^1
$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_1^N,	$\ldots$,	a_k^N	ℓ_1^N,	$\ldots$,	ℓ_m^N

(a) Input training set.

Attributes			$\text{Class}_1 \in \{0,1\}$
a_1^1,	$\ldots$,	a_k^1	ℓ_1^1
$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_1^N,	$\ldots$,	a_k^N	ℓ_1^N

$\cdots$

Attributes			$\text{Class}_m \in \{0,1\}$
a_1^1,	$\ldots$,	a_k^1	ℓ_m^1
$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_1^N,	$\ldots$,	a_k^N	ℓ_m^N

(b) *Binary Relevance* (BR) decomposition.

Attributes			Class $\in \mathcal{L}$
a_1^1,	$\ldots$,	a_k^1	y_1^1
$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_1^1,	$\ldots$,	a_k^1	$y_{\lvert\mathcal{Y}^1\rvert}^1$
$\vdots$	$\ddots$	$\vdots$	$\vdots$

(c) *Multiclass* (MC) decomposition (only for feature selection).

Attributes			Class $\in 2^{\mathcal{L}}$
a_1^1,	$\ldots$,	a_k^1	$\mathcal{Y}^1$
$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_1^N,	$\ldots$,	a_k^N	$\mathcal{Y}^N$

(d) *Label Powerset* (LP) decomposition.

Figure 9.2: Decomposition of multi-label training sets into binary (BR) or multiclass problems (MC, LP). Here, $\mathcal{Y}^i = \{y_1^i, \ldots, y_{|\mathcal{Y}^i|}^i \mid y_j^i \in \mathcal{L}\}$ denotes the assigned labels $\{\ell_j \mid \ell_j^i = 1\}$ to record r^i. MC replicates each record $|\mathcal{Y}^i|$ times, once with each assigned label. In LP, the predicted label set is from the set $\{\mathcal{Y}^i \mid i = 1, \ldots, m\} \subseteq 2^{\mathcal{L}}$ of label sets seen in the training data.

$\ell_1, \ldots, \ell_{i-1}$. Hence, CC captures dependencies of labels on multiple other labels, but the dependencies are one-directional: if label ℓ_i depends on the prediction for label ℓ_j, then ℓ_j does not depend on the prediction for ℓ_i.

An alternative approach is *calibrated label ranking* (CLR) [36], where the key idea is to learn one classifier for each binary comparison of labels. CLR learns binary classifiers $f_{ij} : (a_1, \ldots, a_k) \to (\ell_i \succ \ell_j)$, which predict for each label pair (ℓ_i, ℓ_j) whether ℓ_i is more likely to be relevant than ℓ_j. Thus, CLR (implicitly) takes correlations between pairs of labels into account. The decomposition into pairs of classes has the advantage of simpler subproblems and hence commonly more accurately performing models [74]. CLR ignores dependencies shared between larger sets of labels.

Finally, a simple way to take label dependencies into account is the *label powerset* (LP) approach [107], treating each combination of labels occurring in the training data as a separate value of a multi-class single-label classification problem (Figure 9.2d). Hence, LP caters for dependencies between larger sets of labels as they appear in the dataset. However, LP disregards the inclusion lattice that exists between label sets in MLC. If record r^1 has label set $\{\ell_1, \ell_2\}$, and record r^2 has label set $\{\ell_1, \ell_2, \ell_3\}$, then the label set for r^1 is a subset of the label set for r^2. However, LP will represent these label sets as unrelated values of a single class. So while LP captures subtle label dependencies, this inclusion information is not preserved.

9.3 LeGo Components

As Figure 9.1 illustrates, there are three main components in the LeGo framework. In the following three subsections we will outline what we do in each of these steps.

9.3.1 Local Pattern Mining Phase

In the first phase of the LeGo framework, Local Pattern Mining, we use the Exceptional Model Mining instance defined in Chapter 6. With quality measure φ_{weed}, we find a set P of descriptions for which a Bayesian network, modeling the conditional dependence relations between our labels $\ell_1, \ldots, \ell_m$, has an unusual structure.

9.3.2 Pattern Subset Discovery Phase

Having positioned Local Pattern Mining in a multi-label context, we now proceed to the second phase of the LeGo framework: Pattern Subset Discovery. A common approach for feature subset selection for classification problems is to measure some type of correlation between an attribute and the label. A subset of the attributes S from the whole set P is then determined either by selecting a number of best attributes or by selecting all attributes whose value exceeds a threshold.

Each description from the set P we found in the previous LeGo phase, is by definition a function (cf. Section 2.1), mapping the descriptive attributes of a record in the original dataset to either zero or one. Hence, a description can be trivially transformed into a binary attribute of the dataset, detailing for each record whether it is covered by the description or not. This representation as a binary attribute enables determining the correlation between an element from P and a single class label. However, in MLC, multiple class labels are available, leading to multiple correlation assessments for an element from P. Depending on the effect one strives to achieve, these assessments can be combined in a selection criterion in multiple ways. We experimented with the following approaches.

A simple way is to convert the multi-label problem into a *multiclass* (MC) classification problem, where each original record is converted into several new records, one for each label ℓ_i assigned to the record, using ℓ_i as the class value (see Figure 9.2c). However, this transformation does explicitly model label co-occurrence for a record, not taking the underlying label decomposition into account.

An alternative approach is to measure the correlations on the decomposed subproblems produced by the *binary relevance* (BR) decomposition (see Figure 9.2b). The m different correlation values for each attribute are then aggregated. In our experiments, we aggregated with the max operator, i.e., the overall relevancy of an attribute was determined by its maximum relevance in one of the training sets of the binary relevance classifiers. The main drawback of this approach is that it treats all labels independently and ignores that an attribute might only be relevant for a combination of class labels, but not for the individual labels.

The last approach employs the *label powerset* (LP) transformation (see Figure 9.2d) in order to also measure the correlation of an attribute to the simultaneous absence or occurrence of label sets. Hence, with the dataset transformed into a multiclass problem, common features selection techniques can be applied. The different decomposition approaches are depicted in Figure 9.2.

After the transformations, we can use common attribute correlation measures for evaluating the importance of an attribute in each of the three approaches. In particular, we used the information gain and the χ^2 statis-

tic of an attribute with respect to the class variable resulting from the decomposition, as shown in Figures 9.2b, 9.2c and 9.2d. Then we let each of the six feature selection methods select the best descriptions from P to form the subset S. The size |S| of the subset is fixed in our experiments (see Section 9.3.3).

The approach, adapted from multiclass classification, to measure the correlation between each attribute and the class variable has known weaknesses such as being susceptible to redundancies within the attributes. Hence, in order to evaluate the feature selection methods, we will compare them with the baseline method that simply draws S as a random sample from P.

9.3.3 Global Modeling Phase

For the learning of the global multi-label classification models in the Global Modeling phase, we experiment with several standard approaches including binary relevance (BR) and label powerset (LP) decompositions [106, 107], as well as a selection of effective recent state-of-the-art learners such as calibrated label ranking (CLR) [36, 105], and classifier chains (CC) [92]. The chosen algorithms cover a wide range of approaches and techniques used for learning multi-label problems (see Section 9.2), and are all included in Mulan, a library for multi-label classification algorithms [107, 108].

We combine the multi-label decomposition methods mentioned in Section 9.3.3 with several base learners: J48 with default settings [113], standard LibSVM [10], and LibSVM with a grid search on the parameters. In this last approach, multiple values for the SVM kernel parameters are tried, and the one with the best 3-fold cross-validation accuracy is selected for learning on the training set (as suggested by [10]). Both SVM methods are run once with the Gaussian Radial Basis Function as kernel, and once with a linear kernel using the efficient LibLinear implementation [29]. We will refer to LibSVM with the parameter grid search as MetaLibSVM, and denote the used kernel by a superscript *rbf* or *lin*.

For each classifier configuration, we learn three classifiers based on different attribute sets. The first classifier uses only the k attributes that make up the original dataset, and is denoted C_O (cf. Figure 9.3a). The second classifier, denoted C_S, uses only attributes constructed from our description

Attributes			Labels		
a_1^1,	...,	a_k^1	ℓ_1^1,	...,	ℓ_m^1
a_1^2,	...,	a_k^2	ℓ_1^2,	...,	ℓ_m^2
⋮	⋱	⋮	⋮	⋱	⋮
a_1^N,	...,	a_k^N	ℓ_1^N,	...,	ℓ_m^N

(a) Input training set C_O.

Attributes			Labels		
$D_1(a_1^1,\ldots,a_k^1)$,	...,	$D_{\lvert S\rvert}(a_1^1,\ldots,a_k^1)$	ℓ_1^1,	...,	ℓ_m^1
$D_1(a_1^2,\ldots,a_k^2)$,	...,	$D_{\lvert S\rvert}(a_1^2,\ldots,a_k^2)$	ℓ_1^2,	...,	ℓ_m^2
⋮	⋱	⋮	⋮	⋱	⋮
$D_1(a_1^N,\ldots,a_k^N)$,	...,	$D_{\lvert S\rvert}(a_1^N,\ldots,a_k^N)$	ℓ_1^N,	...,	ℓ_m^N

(b) Transformation into description space C_S.

Attributes						Labels		
a_1^1,	...,	a_k^1,	$D_1(a_1^1,\ldots,a_k^1)$,	...,	$D_{\lvert S\rvert}(a_1^1,\ldots,a_k^1)$	ℓ_1^1,	...,	ℓ_m^1
a_1^2,	...,	a_k^2,	$D_1(a_1^2,\ldots,a_k^2)$,	...,	$D_{\lvert S\rvert}(a_1^2,\ldots,a_k^2)$	ℓ_1^2,	...,	ℓ_m^2
⋮	⋱	⋮	⋮	⋱	⋮	⋮	⋱	⋮
a_1^N,	...,	a_k^N,	$D_1(a_1^N,\ldots,a_k^N)$,	...,	$D_{\lvert S\rvert}(a_1^N,\ldots,a_k^N)$	ℓ_1^N,	...,	ℓ_m^N

(c) Combined attributes in the LeGo classifier C_L.

Figure 9.3: A multi-label classification problem (a), its representation in description space (b) given the set of descriptions $D_1,\ldots,D_{|S|}$, and the LeGo combination (c).

set S (cf. Figure 9.3b). The third classifier employs both the k original and $|S|$ constructed attributes, in the spirit of LeGo, and is hence denoted C_L (cf. Figure 9.3c). Its attribute space consists of the k original attributes, and $|S|$ attributes constructed from the description set S for a grand total of $k + |S|$ attributes.

9.4 Experimental Setup

To experimentally validate the outlined LeGo method, we will compare the performance of the three classifiers based on different attribute sets C_O, C_S, and C_L. We will also investigate the relative performance of the feature selection methods, and of the classification approaches. All experiments

are performed on the *Emotions*, *Scene*, and *Yeast* datasets introduced in Chapter 6; see Table 6.1 for statistics regarding the datasets.

All statistics on the classification processes are estimated via 10-fold cross-validation. To enable a fair comparison of the LeGo classifier with the other classifiers, we let the entire learning process consider only the training set for each fold. This means that we have to run the Local Pattern Mining and Pattern Subset Discovery phase separately for each fold.

For every fold on every dataset, we determine the best $10,000$ descriptions, (if no $10,000$ descriptions can be found, we report them all), measuring the exceptionality with φ_{weed} as described in Chapter 6. We configure the beam search algorithm for EMM with a width of $w = 10$ and a depth of $d = 2$. The modest search depth is selected deliberately, to prevent producing an abundance of highly similar descriptions. We further bound the search by setting the minimal coverage of a description at 10% of the dataset.

For each dataset for each fold, we train classifiers from the three training sets C_O, C_S, and C_L for each combination of a decomposition approach and base learner. We select $|S| = k$ descriptions (cf. Section 9.3.3) from the generated set P, i.e. exactly as many description-based attributes for C_S and C_L as there are original descriptive attributes in C_O.

9.4.1 Evaluation Measures

We evaluate the effectiveness of the three classifiers for each combination on the respective test sets for each fold with five measures: Micro-Averaged Precision and Recall, Subset Accuracy, Ranking Loss, and Average Precision (for details cf. [36] and [107]). We define $\mathcal{Y}^i = \{\ell_j \mid \ell_j^i = 1\}$ as the set of assigned labels and $\widehat{\mathcal{Y}}^i$ as the set of predicted labels for a test record r^i. We consider these a well-balanced selection from the vast set of multi-label measures, evaluating different aspects of multi-label predictions such as good ranking performance and correct bipartition.

From a confusion matrix aggregated over all labels and records, *Precision* (PREC) computes the percentage of predicted labels that are relevant, and *Recall* (REC) computes the percentage of relevant labels that are predicted. Recall and Precision allow a commensurate evaluation of an algorithm, in

contrast to Hamming loss, which is often used but unfortunately generally favors algorithms with high precision and low recall. We have

$$\textsc{Prec} = \frac{\sum_i \left|\widehat{\mathcal{Y}}^i \cap \mathcal{Y}^i\right|}{\sum_i \left|\widehat{\mathcal{Y}}^i\right|} \qquad \textsc{Rec} = \frac{\sum_i \left|\widehat{\mathcal{Y}}^i \cap \mathcal{Y}^i\right|}{\sum_i |\mathcal{Y}^i|}$$

Subset Accuracy (Acc) denotes the percentage of perfectly predicted label sets, forming a multi-label version of traditional accuracy, i.e.

$$\textsc{Acc} = \frac{\sum_i I\left[\widehat{\mathcal{Y}}^i = \mathcal{Y}^i\right]}{\sum_i 1} \qquad \text{where} \quad I[x] = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Since the classifiers we consider are able to return rankings on the labels, we also compute the following rank-based loss measures, in which $\text{rank}(\ell)$ returns the position of label ℓ. *Ranking Loss* (Rank) returns the number of pairs of labels which are not correctly ordered, normalized by the total number of pairs, i.e.

$$\textsc{Rank} = \frac{|\{(\ell \in \mathcal{Y}, \ell' \notin \mathcal{Y}) \mid \text{rank}(\ell) < \text{rank}(\ell')\}|}{|\mathcal{Y}| \cdot (m - |\mathcal{Y}|)}$$

Average Precision (AvgP) computes the precision at each relevant label in the ranking, and averages these over all relevant labels, i.e.

$$\textsc{AvgP} = \frac{1}{|\mathcal{Y}|} \sum_{\ell \in \mathcal{Y}} \frac{|\{\ell' \in \mathcal{Y} \mid \text{rank}(\ell') \leq \text{rank}(\ell)\}|}{\text{rank}(\ell)}$$

These two ranking measures are computed for each record and then averaged over all records.

All values for all settings are averaged over the folds of the cross-validation. Thus we obtain 300 test cases (5 evaluation measures $\times$ 5 base learners $\times$ 4 decomposition approaches $\times$ 3 datasets).

9.4.2 Statistical Testing

To draw conclusions from the long list of raw results we obtained, we use again the methodology for the comparison of multiple algorithms described

by Demšar [19], as introduced in Section 8.2.3. Instead of comparing g quality measures, here we compare the three classifiers C_O, C_S, and C_L. Again, if the difference between the average ranks of two classifiers surpasses the computed critical difference, the better-ranked classifier performs significantly better.

9.5 Experimental Evaluation

The following subsections are dedicated to different aspects such as the evaluation of the different Pattern Subset Discovery approaches, the employment of the different attribute sets, the impact of the decomposition approaches, and efficiency.

9.5.1 Feature Selection Methods

Before comparing the three classifiers, we take a look at the relative performance of the different feature selection methods. When comparing the performance of the classifier C_L with different feature selection methods over all $T = 300$ test cases, we find the average ranks in Table 9.1. We compared the Binary Relevance, Label Powerset and MultiClass approach, each with evaluation measures χ^2 and information gain, and the random baseline approach.

The results show that no classifier employing a sophisticated feature selection method significantly[1] outperforms the classifier with random feature selection. Conversely, the classifier with random feature selection *does* significantly outperform several classifiers employing sophisticated feature selection. For the binary relevance and multiclass approaches this is reasonable, since the descriptions are explicitly designed to consider interdependencies between labels, while the BR and MC approaches select attributes

[1] Since we are comparing $g = 7$ different methods here, the critical value with significance level $\alpha = 5\%$ for the chi-squared distribution with $g - 1 = 6$ degrees of freedom equals 12.592. The Friedman statistic for these ranks equals $\chi_F^2 = 61.678$, hence the Friedman test is passed. For the Nemenyi test, when comparing 7 methods, the critical value is $q_{0.05} = 2.948$. Hence the critical difference between average ranks becomes $CD = 0.520$.

Table 9.1: Average ranks of the feature selection methods (cf. Section 9.3.2), with critical difference. The results for random feature selection are not split out over the two attribute correlation measures, since none is used.

	BR		LP		MC		Random	CD
	χ^2	gain	χ^2	gain	χ^2	gain		
Rank	4.445	3.932	3.507	4.263	3.707	4.490	3.657	0.520

based on their correlation with single labels only and hence ignore interdependencies. The Label Powerset approach should do better in this respect. In fact, the best average rank featured in Table 9.1 belongs to LP with the χ^2 evaluation measure. Since its improvement over the naive method is not significant, we did not further explore its performance, but that does not mean it is without merit.

Another reason for the bad performance of the feature selection methods is that they evaluate each attribute individually. One extreme use case will illustrate the problem: if we replicate each attribute x times and we select the x best attributes according to the presented methods, we will get x times the same attribute. In the Local Pattern Mining phase, we produce a high number of candidate attributes, hence we can expect to obtain groups of similar candidates. The random feature selection does not suffer from this problem. Hence, for the subsequent experiments, we decided not to use any sophisticated feature selection in the remaining experiments, and focus on the results for random feature selection.

9.5.2 Evaluation of the LeGo Approach

The first row in Table 9.2 compares the three different representations C_O, C_S, and C_L over the grand total of 300 test cases in terms of average ranks. We see that both C_O and C_L perform significantly ($\alpha = 5\%$)[2] better than

[2]Since we are comparing three classifiers, the Friedman statistic equals $\chi_F^2 = 52.687$. With significance level $\alpha = 5\%$, the critical value for the chi-squared distribution with 2 degrees of freedom equals 5.991, hence the null hypothesis of the Friedman test is comfortably rejected. For the post-hoc Nemenyi test, when comparing three classifiers the critical value is $q_{0.05} = 2.344$. Hence, the critical difference between average ranks becomes $CD = 0.191$, with significance level $\alpha = 5\%$.

Table 9.2: Comparison of different attribute sets. Average ranks of the three classifiers C_O, C_S, C_L, with critical difference, over all 300 test cases, over all 240 test cases barring J48, over all 60 test cases with a particular base learner, and over all 75 test cases with a particular decomposition method. Bold numbers indicates the top rank in the row, $>$ or $<$ indicate a significant difference to the direct neighbor classifier.

	C_O		C_L		C_S	CD
Overall	1.863	=	**1.797**	>	2.340	0.191
Without J48	1.971	<	**1.733**	>	2.296	0.214
MetaLibSVM^{rbf}	**1.483**	=	1.683	>	2.833	0.428
MetaLibSVM^{lin}	1.900	=	**1.800**	>	2.300	"
LibSVM^{rbf}	2.633	<	**1.683**	=	**1.683**	"
LibSVM^{lin}	1.867	=	**1.767**	>	2.367	"
J48	**1.433**	>	2.050	>	2.517	"
ACC	1.850	=	**1.800**	>	2.350	0.428
PREC	**1.700**	=	1.883	>	2.417	"
REC	1.983	=	**1.700**	>	2.317	"
AVGP	1.850	=	**1.833**	>	2.317	"
RANK	1.933	=	**1.767**	>	2.300	"
CLR	1.813	=	**1.760**	>	2.427	0.383
LP	**1.773**	=	1.827	>	2.400	"
CC	1.947	=	**1.720**	>	2.333	"
BR	1.920	=	**1.880**	=	2.200	"
Emotions	2.510	<	1.860	=	**1.630**	0.331
Scene	**1.480**	=	1.640	>	2.880	"
Yeast	**1.600**	=	1.890	>	2.510	"

C_S, i.e. the description-only classifier cannot compete with the original attributes or the combined classifier. The difference in performance between C_O and C_L is not significant. Although the average rank for the LeGo-based classifier is somewhat higher, we cannot claim that adding local patterns leads to a significant improvement. The remainder of Table 9.2 is concerned with stratified results.

When stratifying the results by base learner (the second block in Table 9.2), we notice a striking difference in average ranks between J48 and the rest.

Table 9.3: Average ranks of the base learners, with critical difference CD.

Approach	Rank
MetaLibSVM^{rbf}	1.489
MetaLibSVM^{lin}	2.972
LibSVM^{lin}	3.228
LibSVM^{rbf}	3.417
J48	3.894
CD	0.455

Restricted to the J48 results, we find that $r_O = 1.433$, $r_S = 2.517$, and $r_L = 2.050$, with CD $= 0.428$. Here, the classifier built from original attributes significantly ($\alpha = 5\%$) outperforms the LeGo classifier.

One reason for the performance gap between J48 and the SVM approach lies in the way these approaches construct their decision boundary. The SVM approaches draw one hyperplane through the attribute space, whereas J48 constructs a decision tree, which corresponds to a decision boundary consisting of axis-parallel fragments. The descriptions the EMM algorithm finds in the Local Pattern Mining phase are constructed by several conditions on single attributes. Hence the domain of each description has a shape similar to a J48 decision boundary, unlike a (non-degenerate) SVM decision boundary. Hence, the expected performance gain when adding such descriptions to the attribute space is much higher for the SVM approaches than for the J48 approach.

Using only the original attributes seems to be enough for the highly optimized non-linear MetaLibSVM^{rbf} method, though the difference with the combined attributes is not statistically significant. The remaining base learners benefit from the added descriptions. Notably, when using LibSVM^{rbf}, it is possible to rely only on the description-based attributes in order to outperform the classifiers trained on the original attributes.

Because the J48 approach results in such deviating ranks, we investigate the relative performance of the base learners. We compare their performance on the three classifiers C_O, C_S, and C_L, with decomposition methods BR, CC, CLR, and LP, on the three datasets *Emotions*, *Scene*, and *Yeast*, evaluated with the measures introduced in Section 9.4. The average ranks of the base

learners over these 180 test cases can be found in Table 9.3; again, the Friedman test is easily passed. The Nemenyi test shows that J48 performs significantly worse than all SVM methods and that MetaLibSVM^{rbf} clearly dominates the performance of the SVMs. This last point is not surprising, since the three datasets are known to be difficult and hence not linearly separable [36], which means that an advantage of the RBF-kernel over the linear kernel can be expected. Moreover, the non-extensively optimized LibSVM^{rbf} can be considered to be subsumed by the meta variant since the grid search includes the default settings.

Having just established that J48 is the worst-performing base learner and, additionally, that the similarity in form of the descriptions and the J48 decision boundary particularly damages the performance of the LeGo classifier, we repeat our overall comparison considering only the SVM variants. Moreover, SVMs are conceptually different from decision tree learners, which additionally justifies the separate comparison. The average ranks of the three classifiers C_O, C_S, and C_L on the remaining 240 test cases can be found in the second row of Table 9.2. This time, the Nemenyi test yields that on the SVM methods the LeGo classifier is generally significantly better than the classifier built from original attributes, even though for MetaLibSVM^{rbf} by itself this is not the case.

When stratifying the results by quality measure (the third block in Table 9.2) we find that the results are consistent over the chosen measures. For all measures we find that C_L significantly outperforms C_S, and C_O always outperforms C_S though not always significantly. Additionally, for all measures except precision, C_L outranks C_O, albeit non-significantly. This consistency provides evidence for robustness of the LeGo method.

The fourth block in Table 9.2 concerns the results stratified by transformation technique. With the exception of the Label Powerset approach, which by itself respects relatively complex label dependencies, all approaches benefit from the combination with the constructed LeGo attributes, though the differences are not statistically significant. Of peculiar interest is the benefit for the binary relevance approach, which in its original form considers each label independently. Though the Friedman test is not passed, the trend is confirmed by the results of CC, which additionally include attributes from the preceding base classifiers' predictions.

As stated in Section 9.2, to predict label ℓ_i the CC decomposition approach allows using the predictions made for labels $\ell_1, \ldots, \ell_{i-1}$. Hence we can view CC as an attribute enriching approach, adding an attribute set C. The result comparing the performance of the different attribute sets peak at O∪S∪C (rank 2.84) followed by O∪S (3.34), O∪C (3.53), O (3.6), S (3.89) and S∪C (3.97) (significant difference only between the first and each of the last combinations). Hence, adding C has an effect on performance similar to the effect of adding S, and BR particularly benefits if both are added, demonstrating that the locally exceptional descriptions provide additional information on the label dependencies, not covered by C.

In the last block of Table 9.2 we see that results vary wildly when stratified by dataset. We see no immediate reason why this should be the case; perhaps a study involving more datasets could be fruitful in this respect.

9.5.3 Evaluation of the Decompositive Approaches

We can learn more from Table 9.2 when more blocks are additionally stratified by evaluation measure. Indeed, the decision of the attribute base does not seem to have an impact on the metrics (for the SVM learners, not shown in the table). The only exception appears to be micro-averaged precision, for which C_O yields a small advantage over C_L. But as Table 9.4 demonstrates, the situation varies with respect to the decompositive approach used. As we can see in the upper block, there are clear tendencies regarding the preference for a particular metric.

For instance, LP has a clear advantage in terms of subset accuracy, which only CC is able to approximate. This stands to reason, since both approaches are dedicated to the prediction of a correct labelset. In fact, LP only predicts label sets previously seen in the training data. CC behaves similarly: if we consider only the additional attributes from the previous predictions (i.e. attribute set C), then we find that CC behaves similar to a sequence tagger. That is to say, for a particular sequence of labels $\ell_1, \ldots, \ell_{i-1}$ the i-th classifier in the chain will tend to predict $\ell_i = 1$ (or $\ell_i = 0$ respectively) only if $\ell_1, \ldots, \ell_i$ existed in the training data. The advantage of LP and CC is confirmed in the bottom block, which restricts the comparison to the usage of the most accurate base learner MetaLibSVM^{rbf}.

Table 9.4: Comparison of the decomposition approaches. The first block compares the approaches for all base learner combinations, the second one restricts on the usage of MetaLibSVM^{rbf}. The first row in each block indicates the average ranks with respect to all evaluation metrics, whereas the following rows distinguish between the individual measures.

Measure	CLR		LP		CC		BR	CD
all & all BC	**1.909**	>	2.462	=	2.700	=	2.929	0.313
Acc	3.400	<	**1.489**	=	1.722	>	3.389	0.700
Prec	1.989	>	3.467	=	3.111	<	**1.433**	0.700
Rec	2.156	=	**1.956**	=	2.422	>	3.467	0.700
AvgP	**1.000**	>	2.778	=	3.111	=	3.111	0.700
Rank	**1.000**	>	2.622	=	3.133	=	3.244	0.700
all & MetaLibSVM^{rbf}	2.111	=	**1.911**	>	2.911	=	3.067	0.700
Acc	3.667	<	**1.000**	=	2.000	=	3.333	1.563
Prec	2.111	=	3.444	=	3.444	<	**1.000**	1.563
Rec	2.778	<	**1.000**	=	2.333	=	3.889	1.563
AvgP	**1.000**	=	2.111	=	3.444	=	3.444	1.563
Rank	**1.000**	=	2.000	=	3.333	=	3.667	1.563

Precision is dominated by BR, followed by CLR. This result is obtained by being very cautious at prediction, as the values for recall show. Especially the highly optimized SVM is apparently fitted towards predicting a label only if it is very confident that the estimation is correct. It is not clear whether this is due to the high imbalance of the binary subproblems, e.g. compared to pairwise decomposition. CLR shows to be more robust, though a bias towards underestimating the size of the label sets is visible. Especially in this case the bias may originate from the conservative BR classifiers, which are included in the calibrated ensemble, since the difference between precision and recall is clearly higher for MetaLibSVM^{rbf}.

Contrasting behavior to BR is shown by LP, which dominates recall, especially for MetaLibSVM^{rbf}, but completely neglects precision. This indicates a preference for predicting the more rare large label sets. The best balance between precision and recall is shown by CLR, even for MetaLibSVM^{rbf}, for which the underestimation leads to low recall, but for which the competing classifier chains obtain the worst precision values together with LP.

The good balancing properties of CLR are confirmed by the results for ranking loss, which are clearly dominated by CLR's ability to produce a high density of relevant labels at the top of the the rankings. The high recall of LP corresponds to good ranking losses, but the low ranks of BR show that its high precision is not due to a good ranking ability. This behavior was already observed, e.g. in [36] and [74], where BR often correctly pushed a relevant class to the top, but obtained poor ranking losses. Similarly, CC's base classifiers are trained independently without a common basis for confidence scores and hence achieve a low ranking quality.

If we give equal weight to the five selected measures, we observe that CLR significantly outperforms the second-placed LP if all base learners are considered, and slightly loses against LP if MetaLibSVM^{rbf} is used (top row in both blocks in Table 9.4).

9.5.4 Efficiency

Apart from the unsatisfactory performance of J48 compared to SVM approaches, Table 9.3 also indicates that compared to the standard LibSVM approach, the extra computation time invested in the MetaLibSVM parameter grid search is rewarded with a significant increase in classifier performance. For both the linear and the RBF kernel, we see that the MetaLibSVM approach outperforms the LibSVM approach, although this difference is only significant for the RBF kernel. A more exhaustive parameter-optimizing search will probably be beneficial, since the grid search considers arbitrary parameter values. Whether the performance increase is worth the invested time is a question of preference. In the case where time and computing power are not limited resources, the increased performance is clearly worthwhile.

From a practical point of view, it is also interesting to analyze the efficiency of using the original attributes in comparison to using the constructed attributes. We expected an improvement in complexity just from the fact that the description-based attributes are binary in contrast to the more complex nature of the original attributes in the used datasets. In addition, it is well known that the possibly resulting sparseness of the binary attributes may also boost algorithms like SVMs.

For the comparison between training of C_O and C_S we focus on the Binary Relevance decomposition setting in order to allow a balanced comparison over the three datasets, since the complexity of BR scales linearly with respect to the number of classes. For J48 as base learner, we observed a reduction of training costs from 28% (*Emotions*) over 31% (*Scene*) to 47% for *Yeast*. For LibSVM^{rbf}, the difference was more pronounced, with a reduction of 60%, 52% and 50%, respectively. We obtained a similar picture for LibSVM^{lin} on two datasets, with a reduction of even 82% (*Emotions*) and 65% (*Scene*). On the *Yeast* dataset, however, training C_S surprisingly takes almost 7 times longer than training C_O. This case is very likely an exception since comparing LibSVM^{lin} and hence using different parameters shows again a clear reduction. The numbers for MetaLibSVM^{lin} and MetaLibSVM^{rbf} are omitted since they show a similar picture but are more difficult to compare directly since they always also include testing time.

Note that training C_L of course takes more computation time since we employ both attribute sets, but the overhead of using the constructed attributes from the descriptions is relatively small. Also, note that the overhead needs to be invested only once for training the classifier, possibly off-line, and that the resulting trained classifier can then be used again and again for classifying data; if one wants to classify more than once, the added complexity diminishes.

9.6 Discussion and Related Work

As we discussed in Section 3.3, exhaustive Local Pattern Mining methods exist. In this chapter, we have selected the Exceptional Model Mining instance with the structure of a Bayesian network model as target concept, to fulfill the Local Pattern Mining phase in the LeGo framework. For this method, no exhaustive approach exists. We expect little disadvantage from using heuristic rather than exhaustive search in the Local Pattern Mining phase. The found descriptions are afterwards put through the Pattern Subset Discovery phase, where they are subjected to feature selection which is heuristic by definition, hence there is really no point in enforcing an exhaustive search in the Local Pattern Mining phase.

The applied Local Pattern Mining algorithm was created to find descriptions that are interesting by themselves. The output of the algorithm is therefore not specifically tailored to be useful in a classification setting; this is not a guiding principle in the Exceptional Model Mining process. To the best of our knowledge, this work is a first attempt at testing the utility for classification of the result of such a stand-alone multi-label description discovery process. Some recent sophisticated classifiers, for instance the multi-label lazy associative classifiers [110], are also based on local patterns. However, these patterns serve only the classifier: interpretation is not considered. Hence the different phases, as present in the LeGo framework, are not as separated as they are in our work. Similarly, Cheng and Hüllermeier [11] incorporate additional attributes that encode the label distribution in the direct neighborhood by, in effect, stacking the output of a k-Nearest Neighbor classifier. However, this has to be done at (training and testing) runtime and cannot be done separately and beforehand.

Other known stacking approaches include the outcome of global classifiers. Godbole and Sarawagi [44] use the outputs of a BR-SVM classifier as additional input attributes for second-level SVMs. Similarly, Tsoumakas et al. [104] replace all original attributes by the predicted scores of a BR. The scores are additionally filtered according to their correlation to each other. The employed classifier chains [92] rely on stacking the outcomes of the predetermined sequence of previous binary relevance classifiers, which permits modeling conditional dependencies, but it does not rely on locality. Zhang and Zhang [118] also try to model label dependencies and start from the premise of eliminating the conditional dependency between the input attributes $a_1, \ldots, a_k$ and the individual labels by computing the errors e_i as difference between true label ℓ_i and the prediction. The isolated dependencies between labels are then approximated by the result of building a Bayesian network on these errors. A new BR classifier is then learned for each class with the set of parents as additional attributes. The very recent LIFT algorithm selects particularly representative centroids in the positive and negative records of a class by k-means clustering and then replaces the original attributes of a record by the distances to these representatives [117]. One may also interpret this approach as a different, pragmatic way of computing new suitable principal components and hence dimensionality reduction, which apparently works quite well.

9.7 Conclusions

We have proposed enhancing multi-label classification methods with local patterns in a LeGo setting. These descriptions are found through an instance of Exceptional Model Mining, a generalization of Subgroup Discovery striving to find subsets of the data with aberrant conditional dependence relations between targets. Hence each delivered description represents a local anomaly in conditional dependence relations between targets. Each description corresponds to a binary attribute which we add to the dataset, to tentatively improve classifier performance.

Experiments on three datasets show that for multi-label SVM classifiers the performance of the LeGo approach is significantly better than the traditional classification performance: investing extra time in running the EMM algorithm pays off when the resulting descriptions are used as constructed attributes. The J48 classifier does not benefit from the local pattern addition, which can be attributed to the similarity of the local decision boundaries produced by the EMM algorithm to those produced by the decision tree learner. Hence the expected performance gain when adding local patterns is lower for J48 than for approaches that learn different types of decision boundaries, such as SVM approaches.

The Friedman-Nemenyi analysis also shows that the constructed attributes generally cannot *replace* the original attributes without significant loss in classification performance. We find this reasonable, since these attributes are constructed from descriptions found by a search process that is not at all concerned with the potential of the descriptions for classification, but is focused on exceptionality. In fact, the description set may be highly redundant. Additionally, it is likely that the less exceptional part of the data, which by definition is the majority of the dataset, is underrepresented by the constructed attributes.

To the best of our knowledge, this is a first attempt at discovering multi-label descriptions and testing their utility for classification in a LeGo setting. Therefore this work can be extended in various ways. It might be interesting to develop more efficient techniques without losing performance. One could also explore other quality measures, such as the plain edit distance measure φ_{ed} from Section 6.3, or other search strategies. In partic-

ular, optimizing the beam search in order to properly balance its levels of exploration and exploitation, could fruitfully produce a more diverse set of attributes [70] in the Local Pattern Mining phase. Alternatively, description diversity could be addressed in the Pattern Subset Discovery phase, ensuring diversity within the subset S rather than enforcing diversity over the whole description set P.

As future work, we would like to expand our evaluation of these methods. Recently, it has been suggested that for multi-label classification, it is better to use stratified sampling than random sampling when cross-validating [97]. Also, experimentation on more datasets seems prudent. In this chapter, we have experimented on merely three datasets, selected for having a relatively low number of labels. As stated in Chapter 6, we have to fit a Bayesian network on the labels for each subgroup under consideration, which is a computationally expensive operation. The availability of more datasets with not too many labels (say, $m < 50$) would allow for more thorough empirical evaluation, especially since it would allow us to draw potentially significant conclusions from Friedman and Nemenyi tests per evaluation measure per base classifier per decomposition scheme. With three datasets this would be impossible, so we decided to aggregate all these test cases in one big test. The observed consistent results over all evaluation measures provide evidence that this aggregation is not completely wrong, but theoretically this violates the assumption of the tests that all test cases are independent. Therefore, the empirically drawn conclusions in this chapter should not be taken as irrefutable proof, but more as evidence contributing to our beliefs.

Acknowledgments

The research in this chapter is financially supported by the German Science Foundation (DFG). We also gratefully acknowledge support by the Frankfurt Center for Scientific Computing.

Chapter 10

Conclusions

We have introduced Exceptional Model Mining (EMM), a general framework to find subgroups of the data where something exceptional, something interesting is going on. These subgroups are not just any subset of the data: they must be coherent records in the dataset, covered by a succinct description in terms of conditions on attributes within the dataset. The attributes that can be used for such a description are strictly separated from the target attributes, which are used to evaluate the subgroups on. Hence, EMM can be seen as an extension of Subgroup Discovery (SD), incorporating a more complex target concept.

In traditional Subgroup Discovery the distribution of a single attribute is the target concept. In Exceptional Model Mining the target concept is a model over multiple attributes. We have discussed several model classes: correlation (Chapter 4), classification (Chapter 5), Bayesian network (Chapter 6), and linear regression models (Chapter 7). For each such model class we have developed quality measures: functions that extract relevant model characteristics, and from those characteristics compute a number quantifying how exceptional a description is. A description is considered exceptional when the model learned from the data covered by the description differs substantially, either from the model learned from the data belonging to its complement, or from the model learned from the overall dataset (for more on this choice, see Section 3.2.2). An Exceptional Model Mining run results in a succinct description of a subgroup, where for instance two targets are unusually correlated, or where a classifier performs

exceptionally good or bad, or where the conditional dependence relations between several targets deviate from the norm.

We have discussed experimental results for each of the introduced model classes. Among the most striking results are the coherent regions within Europe found on the *Mammals* data (see Section 6.2.2) with the Bayesian Network model, where animals depend on each other in a substantially different way, and the strong real-life evidence for the Giffen effect (see Section 7.2.2) found with the General Linear Regression model, where poor households in the Chinese provice Hunan display a positive price elasticity of demand for rice.

Since Exceptional Model Mining strives to find interesting subsets of the dataset, the search space is potentially exponentially large in the number of records in the dataset at hand. This leaves us exposed to the multiple comparisons problem: we are considering a large number of candidates for what essentially amounts to a statistical hypothesis, hence it is likely that by pure random effects, we will unjustly designate some of these candidates as passing the test. Such candidates are called false discoveries. We have demonstrated in Chapter 8 how we can turn the problematic existence of false discoveries into a valuable tool that allows us to solve multiple practical problems in Subgroup Discovery and Exceptional Model Mining. We employ a swap randomization technique to create a search space that is identical to the original search space, but with all connections with and between the targets severed. Running the original SD/EMM algorithm on this search space results in descriptions that can be seen as false discoveries. We build a global model, the Distribution of False Discoveries (DFD), over the qualities of these false discoveries. This enables us to compute a p-value, corresponding to the null hypothesis that a found description is generated by the DFD. Refuting this null hypothesis for a description we found through SD/EMM, implies that this description is unlikely to be a false discovery. Beyond assessing the significance of descriptions, DFD modeling can deliver a quantitative assessment which quality measures are better than others in distinguishing real from false discoveries, and allows us to compute a minimum threshold for each quality measure, that a description must exceed to be considered reliably exceptional.

Having introduced all these instances with their model classes and quality measures, a natural question arising is why Exceptional Model Mining is desirable. We have three answers to that question. For starters, the trivial reason to perform EMM is that we learn things about our data. Extracting pieces of information from a raw dataset is the core business of data mining, and it should not be thought of lightly if a method does merely that. As we have seen in the experimental sections of Chapters 4–7, each description one can find with EMM is such a coherent nugget of information. Those real-life nuggets are far more actionable for a domain expert than the raw data could ever be. Given that EMM is able to capture a richer concept of "interestingness" than conventional Subgroup Discovery, EMM can retrieve descriptions containing more information out of the data than was possible beforehand, as long as the domain expert and the data miner together can formulate a model for the particular concept of interestingness that they strive to find.

Beyond the trivial reason, EMM is a great tool for metalearning. For example, in Chapter 5 we introduced an EMM instance with a classification model as target concept. Hence this instance finds descriptions for which the classification is performed in a substantially different manner than overall, which could be interesting to the researcher. Additionally, one could mine explicitly on a metadataset crafted from the results of a classification run. Suppose one is interested in predicting a numeric variable, for instance the number of days a court case will take to resolve. Having trained and tested a classifier, we end up with a metadataset of court cases, each with the real number of days and the predicted number of days. We can now use these real and predicted numbers as the two targets in an EMM run, for instance using the correlation model from Chapter 4. This EMM run will result in coherent subsets of the data for which the predictions of our classifier are particularly good or bad, which is potentially very useful information for further development or finetuning of the classification algorithm.

Lastly, the descriptions found though EMM may be directly applicable in a setting that is less exploratory and more oriented towards a concrete goal. The EMM instance with a Bayesian network model as target concept, which we discussed in Chapter 6, is a good example. While the original goal of the EMM instance is simply to find descriptions for which the conditional

dependence relations between the targets are unusual, the descriptions have demonstrated their capability to improve multi-label SVM classifiers in Chapter 9, though it does not work as well for decision trees. The main idea is that every description can be seen as a binary attribute of the dataset, indicating whether the record is covered by the description. These binary attributes highlight regions in the dataset where the labels interact in an unusual manner, so employing them in the learning phase may improve a multi-label classifier. Even though predictiveness was not considered at all when the descriptions were found, the classifier performance of SVM methods improved when these additional attributes were available.

As was shortly indicated in Section 3.2.2, efficiency can be an issue when running Exceptional Model Mining. Even with relatively modest parameter settings of the beam search and a reasonably-sized dataset, it is not uncommon to consider a number of descriptions that runs in the hundreds of thousands. For each of these descriptions, a model most be learned from data, and the dissimilarity of two models must be assessed to assign a quality to the subgroup. If either learning the model or assessing the dissimilarity is computationally too expensive, we end up with an intractable algorithm.

When the chosen model class is not too complex (e.g. correlation, the alternative simple linear regression model from Section 7.4, classification), the problem is scarcely more serious than for traditional Subgroup Discovery. For the general linear regression model efficient fitting algorithms exist, and based on upper bounds on eigenvalues and error terms, there is a scheme to prune descriptions on which it is relatively difficult to learn the model [23]. For the Bayesian network model however, the outlook is much bleaker. Without assumptions or heuristics, learning a Bayesian network from data is exponential in the number of vertices in the network [47]. Even with strong restrictions on the network structure, the problem remains superlinear [34]. Hence, for each of the hundreds of thousands of descriptions we learn a model at a high computational cost. We think that the Bayesian network model complexity is on the borderline of what can reasonably be incorporated into the Exceptional Model Mining framework.

To alleviate the efficiency issue, there are a few straightforward steps a researcher can take. When a parallel single-pass algorithm with sublinear memory requirements exists to learn the model from data, we can use the GP-Growth algorithm [72] to prune the search space. Also, choosing to compare the model for a description to the model for the whole dataset, rather than the model for the complement of the description, divides the number of models to be learned by two, as discussed in Section 3.2.2. If all else fails, since we usually resort to heuristic search in EMM, we can set the parameters bounding the search (such as the beam width w discussed in Section 3.1) tighter to reduce the number of descriptions to be evaluated, at the cost of an increased chance that exceptional descriptions remain undiscovered.

Exceptional Model Mining is in many respects a white box system. When employing an EMM instance on a particular domain, it is fairly simple to convey to a domain expert what kind of exceptionality is being sought after (by means of agreeing on the model class). The resulting descriptions are conjunctions of a few conditions on single attributes, which should be simple to interpret for the expert. Depending on the model class, a domain expert may also be able to properly investigate the discrepancies in fitted models. For instance, in the case of a correlation or regression model, this may enrich the expert's understanding of the result, but in the case of a Bayesian network fitted on a hundred animals it probably will not. We expect that deploying existing EMM instances in, or developing new EMM instances for, other fields, could lead to many fruitful collaborations between data miners and experts in those fields.

References

[1] T. Aidt and Z. Tzannatos, Unions and Collective Bargaining, The World Bank, 2002.

[2] P. M. Anglin, R. Gençay, Semiparametric Estimation of a Hedonic Price Function, Journal of Applied Econometrics 11 (6), pp. 633–648, 1996.

[3] K. Bache, M. Lichman, UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml`, Irvine, CA, University of California, School of Information and Computer Science, 2013.

[4] S. D. Bay, M. J. Pazzani, Detecting Group Differences: Mining Contrast Sets, Data Mining and Knowledge Discovery 5 (3), pp. 213–246, 2001.

[5] H. Blockeel, L. De Raedt, J. Ramon, Top-Down Induction of Clustering Trees, Proc. ICML, pp. 55–63, 1998.

[6] M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning Multi-Label Scene Classification, Pattern Recognition 37 (9), pp. 1757–1771, 2004.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA, 1984.

[8] W. L. Buntine, Theory Refinement on Bayesian Networks, Proc. UAI, pp. 52–60, 1991.

[9] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, Bayesian networks and information retrieval: an introduction to the special issue, Information Processing & Management 40 (5), pp. 727–733, 2004.

[10] C.-C. Chang, C.-J. Lin, LIBSVM: a Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology 2 (27), pp. 1–27, 2011.

[11] W. Cheng, E. Hüllermeier, Combining Instance-Based Learning and Logistic Regression for Multilabel Classification, Machine Learning 76 (2-3), pp. 211–225, 2009.

[12] D. M. Chickering, A Transformational Characterization of Equivalent Bayesian Network Structures, Proc. UAI, pp. 87–98, 1995.

[13] R. D. Cook, Detection of Influential Observation in Linear Regression, Technometrics 19 (1), pp. 15–18, 1977.

[14] R. D. Cook, S. Weisberg, Characterizations of an Empirical Influence Function for Detecting Influential Cases in Regression, Technometrics 22 (4), pp. 495–508, 1980.

[15] R. D. Cook, S. Weisberg, Residuals and Influence in Regression, Chapman & Hall, London, 1982.

[16] M. Costanigro, R. C. Mittelhammer, J. J. McCluskey, Estimating Class-Specific Parametric Models under Class Uncertainty: Local Polynomial Regression Clustering in an Hedonic Analysis of Wine Markets, Journal of Applied Econometrics 24, pp. 1117–1135, 2009.

[17] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, D. J. Spiegelhalter, Probabilistic Networks and Expert Systems, Springer-Verlag, New York, pp. 31–33, 1999.

[18] G. A. Davis, Bayesian Reconstruction of Traffic Accidents, Law, Probability and Risk 2, pp. 69–89, 2003.

[19] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research 7, pp. 1–30, 2006.

[20] F. J. Díez, J. Mira, E. Iturralde, S. Zubillaga, DIAVAL, a Bayesian Expert System for Echocardiography, Artificial Intelligence in Medicine 10, pp. 59–73, 1997.

[21] G. Dong, J. Li, Efficient Mining of Emerging Patterns: Discovering Trends and Differences, Proc. KDD, pp. 43–52, 1999.

[22] C. Dougherty, Introduction to Econometrics (4th edition), Oxford University Press, Oxford, 2011.

[23] W. Duivesteijn, A. Feelders, A. Knobbe, Different Slopes for Different Folks – Mining for Exceptional Regression Models with Cook's Distance, Proc. KDD, pp. 868–876, 2012.

[24] W. Duivesteijn, A. Knobbe, Exploiting False Discoveries – Statistical Validation of Patterns and Quality Measures in Subgroup Discovery, Proc. ICDM, pp. 151–160, 2011.

[25] W. Duivesteijn, A. Knobbe, A. Feelders, M. van Leeuwen, Subgroup Discovery meets Bayesian Networks – An Exceptional Model Mining Approach, Proc. ICDM, pp. 158–167, 2010.

[26] W. Duivesteijn, E. Loza Mencía, J. Fürnkranz, A. Knobbe, Multi-label LeGo – Enhancing Multi-label Classifiers with Local Patterns, Proc. IDA, pp. 114-125, 2012.

[27] W. Duivesteijn, E. Loza Mencía, J. Fürnkranz, A. Knobbe, Multi-label LeGo – Enhancing Multi-label Classifiers with Local Patterns, Technical Report, Knowledge Engineering Group, Technische Universität Darmstadt, TUD-KE-2012-02, 2012.

[28] A. Elisseeff, J. Weston, A Kernel Method for Multi-Labelled Classification, Advances in Neural Information Processing Systems 14, pp. 681–687, MIT Press, Cambridge, MA, 2002.

[29] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9, pp. 1871–1874, 2008.

[30] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine 17 (3), pp. 37-54, 1996.

[31] M. Friedman, The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance, Journal of the American Statistical Association 32, pp. 675–701, 1937.

[32] M. Friedman, A Comparison of Alternative Tests of Significance for the Problem of m Rankings, Annals of Mathematical Statistics 11, pp. 86–92, 1940.

[33] N. Friedman, M. Linial, I. Nachman, D. Peér, Using Bayesian Networks to Analyze Expression Data, Journal of Computational Biology 7 (3-4), pp. 601–620, 2000.

[34] N. Friedman, I. Nachman, D. Peér, Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm, Proc. UAI, pp. 196–205, 1999.

[35] J. Fürnkranz, P. A. Flach, ROC 'n' Rule Learning – Towards a Better Understanding of Covering Algorithms, Machine Learning 58 (1), pp. 39–77, 2005.

[36] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, K. Brinker, Multilabel Classification via Calibrated Label Ranking, Machine Learning 73 (2), pp. 133–153, 2008.

[37] J. Fürnkranz, A. Knobbe (eds.), Special Issue: Global Modeling Using Local Patterns, Data Mining and Knowledge Discovery journal 20 (1), 2010.

[38] E. Galbrun, P. Miettinen, From Black and White to Full Color: Extending Redescription Mining Outside the Boolean World, Statistical Analysis and Data Mining 5 (4), pp. 284–303, 2012.

[39] A. Gallo, P. Miettinen, H. Mannila, Finding Subgroups having Several Descriptions: Algorithms for Redescription Mining, Proc. SDM, pp. 334–345, 2008.

[40] G. C. Garriga, H. Heikinheimo, J. K. Seppänen, Cross-Mining Binary and Numerical Attributes, Proc. ICDM, pp. 481-486, 2007.

[41] C. F. Gauss, Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium, Friedrich Perthes and I. H. Besser, Hamburg, 1809.

[42] J. F. Gentleman, M. B. Wilk, Detecting Outliers II: Supplementing the Direct Analysis of Residuals, Biometrics 31, pp. 387–410, 1975.

[43] A. Gionis, H. Mannila, T. Mielikäinen, P. Tsarapas, Assessing Data Mining Results via Swap Randomization, Proc. KDD, pp. 167–176, 2006.

[44] S. Godbole, S. Sarawagi, Discriminative Methods for Multi-Labeled Classification, Proc. PAKDD, pp. 22–30, 2004.

[45] H. Grosskreutz, S. Rüping, On Subgroup Discovery in Numerical Domains, Data Mining and Knowledge Discovery 19 (2), pp. 210–226, 2009.

[46] H. Grosskreutz, S. Rüping, S. Wrobel, Tight Optimistic Estimates for Fast Subgroup Discovery, Proc. ECML/PKDD (1), pp. 440–456, 2008.

[47] D. Heckerman, A Tutorial on Learning with Bayesian Networks, Proc. NATO Advanced Study Institute on Learning in Graphical Models, pp. 301–354, Kluwer Academic Publishers, Norwell, MA, 1998.

[48] D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian Networks: the Combination of Knowledge and Statistical Data, Machine Learning 20, pp. 197–243, 1995.

[49] H. Heikinheimo, M. Fortelius, J. Eronen, H. Manilla, Biogeography of European Land Mammals Shows Environmentally Distinct and Spatially Coherent Clusters, Journal of Biogeography 34 (6), pp. 1053–1064, 2007.

[50] F. Herrera, C. J. Carmona, P. González, M. J. del Jesus, An Overview on Subgroup Discovery: Foundations and Applications, Knowledge and Information Systems 29 (3), pp. 495–525, 2011.

[51] D. C. Hoaglin, R. Welsh, The Hat Matrix in Regression and ANOVA, American Statistician 32, pp. 17–22, 1978.

[52] Y. Hochberg, A. Tamhane, Multiple Comparison Procedures, Wiley, New York, 1987.

[53] R. T. Jensen, N. H. Miller, Giffen Behavior and Subsistence Consumption, American Economic Review 98 (4), pp. 1553–1577, 2008.

[54] A. M. Jorge, P. J. Azevedo, F. Pereira, Distribution Rules with Numeric Attributes of Interest, Proc. PKDD, pp. 247–258, 2006.

[55] W. Klösgen, Explora: A Multipattern and Multistrategy Discovery Assistant, Advances in Knowledge Discovery and Data Mining, pp. 249–271, 1996.

[56] W. Klösgen, Subgroup Discovery, in: W. Klösgen, J.M. Zytkow (eds.), Handbook of Data Mining and Knowledge Discovery, pp. 354–361, Oxford University Press, Oxford, 2002.

[57] A. Knobbe, B. Crémilleux, J. Fürnkranz, M. Scholz, From Local Patterns to Global Models: the LeGo Approach to Data Mining, Proc. ECML/PKDD Workshop: From Local Patterns to Global Models, pp. 1–16, 2008.

[58] A. Knobbe, E. Ho, Pattern Teams, Proc. PKDD, pp. 577–584, 2006.

[59] A. Knobbe, J. Valkonet, Building Classifiers from Pattern Teams, Proc. ECML PKDD Workshop: From Local Patterns to Global Models, pp. 77–93, 2009.

[60] D. E. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching, second edition, Addison–Wesley, Reading, MA, 1998.

[61] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Tree Ensembles for Predicting Structured Outputs, Pattern Recognition 46 (3), pp. 817–833, 2013.

[62] R. Kohavi, The Power of Decision Tables, Proc. ECML, pp. 174–189, 1995.

[63] R. M. Konijn, W. Kowalczyk, Hunting for Fraudsters in Random Forests, Proc. HAIS, pp. 174–185, 2012.

[64] E. van de Koppel, I. Slavkov, K. Astrahantseff, A. Schramm, J. Schulte, J. Vandesompele, E. de Jong, S. Dzeroski, A. Knobbe, Knowledge Discovery in Neuroblastoma-related Biological Data, Proc. PKDD Workshop: Data Mining in Functional Genomics and Proteomics, pp. 45-56, 2007.

[65] P. Kralj Novak, N. Lavrač, G. I. Webb, Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining, Journal of Machine Learning Research 10, pp. 377–403, 2009.

[66] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek, Outlier Detection in Arbitrarily Oriented Subspaces, Proc. ICDM, pp. 379–388, 2012.

[67] P. Larrañaga, M. Poza, Y. Yurramendi, R. H. Murga, C. M. H. Kuijpers, Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (9), pp. 912–926, 1996.

[68] N. Lavrač, B. Kavšek, P. Flach, L. Todorovski, Subgroup Discovery with CN2-SD, Journal of Machine Learning Research 5, pp. 153–188, 2004.

[69] M. van Leeuwen, Maximal Exceptions with Minimal Descriptions, Data Mining and Knowledge Discovery 21 (2), pp. 259–276, 2010.

[70] M. van Leeuwen, A. J. Knobbe, Non-redundant Subgroup Discovery in Large and Complex Data, Proc. ECML PKDD (3), pp. 459–474, 2011.

[71] D. Leman, A. Feelders, A. Knobbe, Exceptional Model Mining, Proc. ECML/PKDD (2), pp. 1–16, 2008.

[72] F. Lemmerich, M. Becker, M. Atzmüller, Generic Pattern Trees for Exhaustive Exceptional Model Mining, Proc. ECML-PKDD (2), pp. 277–292, 2012.

[73] E. Loza Mencía, Efficient Pairwise Multilabel Classification, PhD thesis, Technische Universität Darmstadt, 2012.

[74] E. Loza Mencía, J. Fürnkranz, Efficient Pairwise Multilabel Classification for Large-Scale Problems in the Legal Domain, Proc. ECML/PKDD (2), pp. 50–65, 2008.

[75] A. M. Lyapunov, Nouvelle Forme du Théorème sur la Limite de Probabilité, Mémoires de l'Académie Impériale des Sciences de St. Petersburg 12, pp. 1–24, 1901.

[76] M. Mampaey, S. Nijssen, A. Feelders, A. J. Knobbe, Efficient Algorithms for Finding Richer Subgroup Descriptions in Numeric and Nominal Data, Proc. ICDM, pp. 499–508, 2012.

[77] A. Marshall, Principles of Economics, MacMillan and Co., London, 1895.

[78] M. Meeng, A. J. Knobbe, Flexible Enrichment with Cortana – Software Demo, Proc. Benelearn, pp. 117–119, 2011.

[79] N. Megiddo, R. Srikant, Discovering Predictive Association Rules, Proc. KDD, pp. 274–278, 1998.

[80] A. J. Mitchell-Jones, W. Bogdanowicz, B. Krystufek, P. J. H. Reijnders, F. Spitzenberger, C. Stubbe, J. B. M. Thissen, V. Vohralík, J. Zima, The Atlas of European Mammals, Poyser Natural History, Academic Press, London, 1999.

[81] D. Moore, G. McCabe, Introduction to the Practice of Statistics, Freeman, New York, 1993.

[82] M. Neil, N. Fenton, M. Tailor, Using Bayesian Networks to Model Expected and Unexpected Operational Losses, Risk Analysis 25 (4), 2005.

[83] P. B. Nemenyi, Distribution-Free Multiple Comparisons, PhD thesis, Princeton University, 1963.

[84] J. Neter, M. Kutner, C. J. Nachtsheim, W. Wasserman, Applied Linear Statistical Models, WCB McGraw-Hill, New York, 1996.

[85] M. Ojala, G. C. Garriga, A. Gionis, H. Mannila, Evaluating Query Result Significance in Databases via Randomizations, Proc. SDM, pp. 906–917, 2010.

[86] R. T. Paine, Food Web Complexity and Species Diversity, The American Naturalist 100 (910), pp. 65–75, 1966.

[87] S.-H. Park, J. Fürnkranz, Multi-Label Classification with Label Constraints, Proc. ECML/PKDD Workshop: Preference Learning, pp. 157–171, 2008.

[88] K. Pearson, L. Filon, Mathematical Contributions to the Theory of Evolution, iv. on the Probable Errors of Frequency Constants and on the Influence of Random Selection on Variation and Correlation, Philosophical Transactions of the Royal Society of London, Series A, Containing Papers of a Mathematical or Physical Character, 191, pp. 229–311, 1898.

[89] B. F. I. Pieters, A. Knobbe, S. Džeroski, Subgroup Discovery in Ranked Data, with an Application to Gene Set Enrichment, Proc. ECML PKDD Workshop: Preference Learning, 2010.

[90] J. R. Quinlan, Learning with Continuous Classes, Proc. AJCAI, pp. 343–348, 1992.

[91] N. Ramakrishnan, D. Kumar, B. Mishra, M. Potts, R. F. Helm, Turning CARTwheels: an Alternating Algorithm for Mining Redescriptions, Proc. KDD, pp. 837–844, 2004.

[92] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier Chains for Multi-label Classification, Proc. ECML PKDD, pp. 254–269, 2009.

[93] L. Rezende, Econometrics of Auctions by Least Squares, Journal of Applied Econometrics 23, pp. 925–948, 2008.

[94] J. A. Rice, Mathematical Statistics and Data Analysis, second edition, Duxbury Press, Wadsworth Publishing Company, Belmont, CA, 1995.

[95] C. Riggelsen, Approximation Methods for Efficient Learning of Bayesian Networks, IOS Press, Amsterdam, 2008.

[96] E. Schubert, J. Wolfe, A. Tarnopolsky, Spectral Centroid and Timbre in Complex, Multiple Instrumental Textures, Proc. 8th International Conference on Music Perception & Cognition, pp. 654–657, 2004.

[97] K. Sechidis, G. Tsoumakas, I. P. Vlahavas, On the Stratification of Multi-label Data, Proc. ECML PKDD (3), pp. 145–158, 2011.

[98] C. Silverstein, S. Brin, R. Motwani, Beyond Market Baskets: Generalizing Association Rules to Dependence Rules, Data Mining and Knowledge Discovery 2 (1), pp. 39–68, 1998.

[99] T. Stengos, E. Zacharias, Intertemporal Pricing and Price Discrimination: A Semiparametric Hedonic Analysis of the Personal Computer Market, Journal of Applied Econometrics 21, pp. 371–386, 2006.

[100] J.-N. Sulzmann, J. Fürnkranz, A Comparison of Techniques for Selecting and Combining Class Association Rules, Proc. ECML/PKDD Workshop: From Local Patterns to Global Models, pp. 154-168, 2008.

[101] P.-N. Tan, V. Kumar, J. Srivastava, Selecting the right interestingness measure for association patterns, Proc. KDD, pp. 32–41, 2002.

[102] A. Tellegen, D. Watson, L. A. Clark, On the Dimensional and Hierarchical Structure of Affect, Psychological Science 10 (4), pp. 297–303, 1999.

[103] K. Trohidis, G. Tsoumakas, G. Kalliris, I. P. Vlahavas, Multi-Label Classification of Music into Emotions, Proc. 9th International Conference on Music Information Retrieval, pp. 325–330, 2008.

[104] G. Tsoumakas, A. Dimou, E. Spyromitros Xioufis, V. Mezaris, I. Kompatsiaris, I. Vlahavas, Correlation Based Pruning of Stacked Binary Relevance Models for Multi-Label Learning, Proc. 1st International Workshop on Learning from Multi-Label Data, pp. 101–116, 2009.

[105] G. Tsoumakas, E. Loza Mencía, I. Katakis, S.-H. Park, J. Fürnkranz, On the Combination of Two Decompositive Multi-Label Classification Methods, Proc. ECML PKDD Workshop: Preference Learning, pp. 114–129, 2009.

[106] G. Tsoumakas, I. Katakis, Multi-Label Classification: An Overview, International Journal of Data Warehousing and Mining 3 (3), pp. 1–13, 2007.

[107] G. Tsoumakas, I. Katakis, I. P. Vlahavas, Mining Multi-label Data, Data Mining and Knowledge Discovery Handbook, Springer, New York, pp. 667–685, 2010.

[108] G. Tsoumakas, J. Vilcek, E. Spyromitros Xioufis, I. P. Vlahavas, Mulan: A Java Library for Multi-Label Learning, Journal of Machine Learning Research 12, pp. 2411–2414, 2011.

[109] L. Umek, B. Zupan, Subgroup Discovery in Data Sets with Multi-Dimensional Responses, Intelligent Data Analysis 15 (4), pp. 533–549, 2011.

[110] A. Veloso, W. Meira Jr., M. A. Gonçalves, M. J. Zaki, Multi-label Lazy Associative Classification, Proc. PKDD, pp. 605–612, 2007.

[111] T. Verma, J. Pearl, Equivalence and Synthesis of Causal Models, Proc. UAI, pp. 255–270, 1990.

[112] G. I. Webb, Discovering Significant Patterns, Machine Learning 68 (1), pp. 1–33, 2007.

[113] I. H. Witten, E. Frank, M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco, CA, 2011.

[114] S. Wrobel, An Algorithm for Multi-relational Discovery of Subgroups, Proc. PKDD, pp. 78–87, 1997.

[115] G. Yang, L. Le Cam, Asymptotics in Statistics: Some Basic Concepts, Berlin, Springer-Verlag, 2000.

[116] B. Zhang, Regression Clustering, Proc. ICDM, pp. 451–458, 2003.

[117] M.-L. Zhang, Lift: Multi-Label Learning with Label-Specific Features, Proc. 22nd International Joint Conference on Artificial Intelligence, pp. 1609–1614, 2011.

[118] M.-L. Zhang, K. Zhang, Multi-Label Learning by Exploiting Label Dependency, Proc. KDD 2010, pp. 999–1008, 2010.

Nederlandse Samenvatting

Wanneer we een grote verzameling ruwe data hebben, poogt het deelgebied van Informatica genaamd *Datamining* er informatie uit te destilleren die kan worden geïnterpreteerd door de eindgebruiker, en bij voorkeur kan worden benut in het domein waar de gebruiker in geïnteresseerd is. Met de toegenomen rol van het internet in het dagelijks leven, en in het bijzonder de opkomst van smartphones, genereert iedere persoon en ieder bedrijf enorme hoeveelheden gegevens. Er ligt een taak voor de wetenschappelijke gemeenschap om methoden te ontwikkelen die een onoverzichtelijke hoeveelheid data nemen en er klompjes zinnige informatie uitpersen.

Elementen aanwijzen die afwijken van de norm is een belangrijke taak. Het meeste dataminingonderzoek in deze richting concentreert zich op afwijkingen *detecteren*. In Lokale Patroonmining zijn we echter niet tevreden met het aanwijzen van afwijkende elementen in de data. In plaats daarvan zoeken we naar *subgroepen*: coherente deelverzamelingen die kunnen worden *beschreven* door een klein aantal voorwaarden op attributen van de data. Het bestaan van zulke beschrijvingen maakt de resulterende afwijkende subgroepen meer actiegericht: als we bijvoorbeeld een farmaceutisch bedrijf vertellen dat vijf bepaalde personen slecht reageren op een medicijn, dan kan het bedrijf daar minder mee dan ze zouden kunnen als we ze kunnen vertellen dat de groep rokers slecht reageert op het medicijn.

"Afwijken van de norm" is multi-interpretabel. Traditioneel is zo'n uitzonderlijkheid gedefinieerd op basis van veelvoorkomendheid, of op basis van een afwijkende verdeling van één doelattribuut. Deze concepten omvatten niet alle potentieel interessante afwijkingen. Om deze algemene interesse vorm te geven, hebben we Exceptional Model Mining (het graven naar uitzonderlijke modellen) ontwikkeld.

De eerste stap van het EMM-raamwerk is het verdelen van de attributen in twee delen: een deel om de subgroepen op te *definiëren* (de *beschrijvers*), en een deel om de subgroepen op te *evalueren* (de *doelwitten*). Dan selecteren we een *modelklasse* over de doelwitten, en ontwerpen we een *kwaliteitsmaat* over de modelklasse. Ten slotte gebruiken we de al bestaande Subgroup Discovery methode om de beschrijver-ruimte te doorzoeken naar subgroepen die goed presteren volgens de kwaliteitsmaat. De modelklasse vertegenwoordigt het samenspel tussen de doelwitten, en de kwaliteitsmaat meet de uitzonderlijkheid van modelparameters. Bijvoorbeeld kunnen we zo subgroepen vinden waarvoor twee doelwitten ongebruikelijk correleren, waarvoor een Bayesiaans netwerk een afwijkende structuur heeft, of waarvoor een regressiemodel een uitzonderlijke parametervector heeft.

Het doorzoeken van de beschrijver-ruimte kost veel rekenkracht: we zoeken interessante deelverzamelingen van de data, en je kunt je voorstellen dat er van een grote dataverzameling veel kandidaat-deelverzamelingen zijn. Voor de EMM-instantie met regressie als modelklasse hebben we bovengrenzen afgeleid op de kwaliteit van een kandidaat, die we kunnen uitrekenen zonder de parametervector zelf uit te rekenen. Met behulp van deze bovengrenzen kunnen we deze dure laatste rekenstap vermijden voor maximaal 40% van de kandidaten, waardoor het hele proces sneller verloopt.

Door EMM-instanties hebben we subgroepen gevonden die weersomstandigheden betreffen waaronder voedselketens ontsporen, subgroepen die de economische wet van vraag en aanbod tarten, subgroepen die het dempende effect van vakbonden op de salarisverdeling illustreren, et cetera. Daarnaast hebben we een test ontwikkeld of zulke subgroepen valse ontdekkingen zijn: wanneer we een toets vaak uitvoeren zullen we uiteindelijk iets schijnbaar significants vinden, puur door toevalseffecten. Door een model te bouwen van kunstmatig gegenereerde valse ontdekkingen, en gevonden subgroepen hiermee te vergelijken, kunnen we inschatten of het waarschijnlijk is dat onze subgroepen ook valse ontdekkingen zijn.

We hebben bepaald of de resultaten van één EMM-instantie (met een Bayesiaans netwerk als modelklasse) ook kunnen worden gebruikt om de doelwitten beter te *voorspellen* wanneer we nieuwe data binnenkrijgen. We laten zien dat we door uitzonderlijk samenspel tussen de doelwitten met EMM te vatten, soms in staat zijn deze voorspellingen te verbeteren.

English Summary

When given a large volume of raw data, the Computer Science subfield called *Data Mining* strives to extract information from the data; information that can be interpreted by whoever is using the data mining method at hand, and preferably used within the domain that person is interested in. With the advance of the internet in everyday life, and particularly the ubiquity of smartphones nowadays, gargantuan amounts of data are being generated by every person and company in the world. Hence the scientific community needs to develop methods that take on a seemingly uninspectable amount of data and squeeze out nuggets of information.

Identifying elements that behave differently from the norm is a task of paramount importance. Most data mining research in this direction focuses on *detecting* outliers. In Local Pattern Mining, however, we are not just looking for any deviating record or set of records in the data. Instead, we are looking for *subgroups*: coherent subsets that can be *described* in terms of a few conditions on attributes of the data. The existence of such descriptions makes the resulting deviating subgroups more *actionable*: for instance, if we tell a pharmaceutical company that five given persons react badly to a certain type of medication, it is more difficult for them to act on the information than it would be if we could tell them that the group of smokers react badly to the medication.

"Behaving differently from the norm" can be defined in many ways. Traditionally such exceptionality is measured in terms of frequency (Frequent Itemset Mining), or in terms of a deviating distribution of one target attribute (Subgroup Discovery). These concepts do not encompass all forms of deviation we may be interested in. To accomodate a more general form of interestingness, we developed *Exceptional Model Mining*.

The first step of the EMM framework is partitioning the attributes in two: one set to *define* subgroups on (the *descriptors*), and one set to *evaluate* the subgroups on (the *targets*). Then a *model class* is selected over the targets, and a *quality measure* over this model class is designed. Finally, the already existing Subgroup Discovery methodology is used to scan the descriptor space for subgroups that perform well according to the quality measure. The model class represents interplay between the targets, and the quality measure gauges the exceptionality of model parameters. For instance, we can find subgroups for which two targets are unusually correlated, for which a classifier performs unusually, for which a Bayesian network on several nominal targets has a deviating structure, or for which a regression model has an exceptional parameter vector.

Scanning the descriptor space is computationally very intensive: we search for interesting subsets of a dataset, and one can imagine that for a large datasets, there are many candidate subsets. For the EMM instance with the regression model class, we have derived some upper bounds on the quality of a candidate subgroup, that can be computed without computing the parameter vector. Using the bounds, this last relatively expensive computation step can be omitted for up to 40% of the candidate subgroups, thus speeding up the whole process.

Using EMM instances, we have found subgroups concerning meteorological conditions coinciding with food chain displacement, subgroups defying the economical law of demand, subgroups showcasing the dampening effect of collective bargaining on the distribution of salaries, etcetera. Additionally, we have developed a method to test whether such subgroups are false discoveries: solving a statistical problem roughly stating that, when we run many tests, we will eventually find something seemingly significant, purely by random effects. By generating a baseline of artificial false discoveries, and comparing the subgroups we find with the baseline, we can assess whether it is likely that our found subgroups are false discoveries too.

We have determined whether the results of one EMM instance (with the Bayesian network model) can additionally be used to improve the *prediction* of the targets when we are given new records of our dataset. Capturing the exceptional target interplay through EMM is shown capable of improving such a prediction in certain cases.

Acknowledgments

Many thanks to the following people.

My coworkers at LIACS: Joost Kok, Arno Knobbe, Marvin Meeng, Rob Konijn, Arne Koopman, Xin Li, Ugo Vespier, Ricardo Cachucho and Irene Martorelli, Shengfa Miao, Joaquin Vanschoren, Siegfried Nijssen, Michael Mampaey, Jan van Rijn, Wojtek Kowalczyk, Peter van der Putten, Hendrik Blockeel, Carla Silva, Geraldine Ribeiro, Ana Loureiro, Claudio Sá, Jefrey Lijffijt and Lotte van den Berg, Jouke Witteveen, and Peter Grünwald.

The LaDiDa group at Utrecht University: Arno Siebes, Ad Feelders, Matthijs van Leeuwen, Nicola Barile, Diyah Puspitaningrum, Roel Bertens, Jilles Vreeken, Hans Philippi, and Lennart Herlaar.

The TU Darmstadt Knowledge Engineering group: Johannes Fürnkranz, Eneldo Loza Mencía, Jan-Nikolas Sulzmann, Frederik Janssen, Sang-Hyeun Park, Lorenz Weizsäcker, and Dirk große Osterhues.

My friends at the VARIENG Research Unit: Terttu Nevalainen, Tanja Säily, and Mikko Hakala.

My family: Teun van Oudheusden and Lena van Oudheusden-van Seters, Cees Duivesteijn and Sjane Duivesteijn-van Oudheusden, and Hugo Duivesteijn and Ines van Drie.

My friends: Thomas van Dijk, Chris van Dijk and Gerri van Dijk-Engelen, Joeri Noort and Karin Lemmers, Remco Okhuijsen, Erik van Ommeren, Jan-Pieter van den Heuvel and Gwyneth van den Heuvel-Ouwehand, Wouter Slob and Frances de Kok, and Bas den Heijer.

Thank you for being part of the journey (so far).

Curriculum Vitae

Wouter Duivesteijn was born in Rotterdam, the Netherlands, on Sunday, the 9th of December in 1984. From 1996 until 2002 he was a student at Penta College C.S.G. Blaise Pascal in Spijkenisse, at the Gymnasium level. He then studied a double major in Mathematics and Computer Science at Utrecht University, obtaining B.Sc. degrees in both disciplines in 2005. In 2005 he began studying a double Master's programme at Utrecht University, graduating with a M.Sc. degree in Fundamental Mathematics in October 2007, and a M.Sc. degree in Applied Computing Science in October 2008. During his studies, from September 2007 until September 2008, he was the president of Study Association A–Eskwadraat, organizing activities for approximately 1700 members studying Mathematics, Physics, Computer Science, or Information Science at Utrecht University. In October 2008, he began studying for yet another M.Sc. degree, in History & Philosophy of Science. He abandoned this educational path when he was given the opportunity to become a Ph.D. candidate.

In July 2009, Wouter started his Ph.D. studies in the Data Mining group of the Algorithms cluster at LIACS, the Leiden Institute of Advanced Computer Science, at Leiden University. This Ph.D. trajectory was performed under the guidance of Joost N. Kok, and with direct supervision of Arno J. Knobbe. Since the project involved researchers not only from Leiden University, but also from Utrecht University, Wouter spent part of his time as a guest at the Algorithmic Data Analysis group at Utrecht University, under the guidance of Arno P.J.M. Siebes, and with direct supervision of Ad J. Feelders.

Wouter is driven by his incurable curiosity. Outside of his job as a researcher, this is expressed by his insatiable hunger for new music to sing along or listen to (a hunger once instilled by listening to The Beatles, and yearly reinforced by visiting the North Sea Jazz festival), and his enjoyment in traveling the world. He strives to have visited every continent on this planet before his 30^{th} birthday, and will continue his academic career as a postdoctoral researcher at Sonderforschungsbereich 876: Verfügbarkeit von Information durch Analyse unter Ressourcenbeschränkung, a collaborative research center which is part of the Technische Universität Dortmund. Wouter is looking forward to exploring that corner of the (scientific) world.

Dit is het begin
Maar het lijkt wel een einde
— Spinvis, De Zevende Nacht

