SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of

STATISTICS AND QUANTITATIVE METHODS

PhD program in **STATISTICS AND MATHEMATICAL FINANCE** Cycle **XXXII**

Curriculum in **STATISTICS**

ROBUST MODEL-BASED CLASSIFICATION AND CLUSTERING: ADVANCES IN LEARNING FROM CONTAMINATED DATASETS

Surname: **CAPPOZZO** Name: **ANDREA**

Registration number: **814541**

Tutor: Prof. FRANCESCA **GRESELIN**

Supervisor: Prof. FRANCESCA **GRESELIN**

Co-Supervisor: Prof. THOMAS BRENDAN **MURPHY**

Coordinator: Prof. GIORGIO **VITTADINI**

ACADEMIC YEAR 2018/2019

UNIVERSITY OF MILANO BICOCCA

DOCTORAL THESIS

Robust model-based classification and clustering: advances in learning from contaminated datasets

Author:
Andrea CAPPOZZO

Supervisor:
Prof. Francesca GRESELIN
Prof. Brendan MURPHY

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Statistics and Quantitative Methods

28 October 2019

Abstract

At the time of writing, an ever-increasing amount of data is collected every day, with the volume of such generated records estimated to be doubling every two years. Thanks to the technological advancements, datasets are becoming massive in terms of size and substantially more complex in nature. Nevertheless, this abundance of "raw information" does come at a price: wrong measurements, data-entry errors, breakdowns of automatic collection systems and several other causes may ultimately undermine the overall data quality. The percentage of encoding errors in real-world databases, all fields taken together, is estimated to be approximately five percent. In particular, unreliable observations are ubiquitously encountered in applications that require involved recording operations, complex preprocessing procedures and whenever human supervision and subjective judgment are unavoidable. Medical studies, for instance, are often based on data self-recorded by patients, who tend to underestimate and/or hide sensitive behaviors; furthermore, also on the physicians side biased and wrong diagnoses may occur. Image analysis is also prone to errors, since data quality is highly influenced by the level of details in each sample and, when a labeling task is involved, to the variability associated to it. Food authenticity studies are also critical: no observation is to be trusted in a context wherein the final purpose is exactly to detect potential adulterated units. Metagenomics analyses too require careful sample preparations and delicate bioinformatics procedures, during which many initial units are discarded due to their unreliability. Lastly, in chemometrics, calibration errors in the machinery and interexpert variability in samples preparation can ultimately generate untrustworthy recordings. The real applications considered in this manuscript stem precisely from the last three mentioned fields. To this extent, robust methods have a central role in properly converting contaminated "raw information" to trustworthy knowledge: a primary goal of any statistical analysis.

The present manuscript presents novel methodologies for performing reliable inference, within the model-based classification and clustering framework, in presence of contaminated data. First, we propose a robust modification to a family of semi-supervised patterned models, for accomplishing classification when dealing with both class and attribute noise. Second, we develop a discriminant analysis method for anomaly and novelty detection, with the final aim of discovering label noise, outliers and unobserved classes in an unlabeled dataset. Third, we introduce two robust variable selection methods, that effectively perform high-dimensional discrimination within an adulterated scenario.

The thesis is organized as follows. Chapter 1 reviews the statistical concepts of model-based classification and robustness, with focus on tools and methodologies that will be considered in the later sections. In addition, we list here the main real-data complexities that are encountered in this context: they are the motivations that led to the novel contributions present in the remaining parts of the manuscript. In Chapter 2, we introduce a family of robust semi-supervised patterned models, resistant to the harmful effects produced by wrongly labeled units and observations with corrupted attributes. Making use of impartial trimming and eigenvalue-ratio constraints, robust parameter estimates are obtained and a robust classification rule is defined. Chapter 3 describes a model-based discriminant analysis method for anomaly and novelty detection. We show that the methodology effectively performs classification in presence of label noise, outliers and unobserved classes in the test set. Parameters of known and hidden groups

are robustly estimated via two flexible EM-based approaches, one considering the union of training and test sets, and the other made of two phases, performing sequential inference for known and hidden classes. Chapter 4 deals with adulterated high-dimensional data, and how to develop a robust model-based classifier in this context, where noise assumes the form of outliers, label noise and irrelevant features. To this aim, we introduce two wrapper variable selection methods for identifying the relevant features, that is those bringing significant information about class separation. The first method embeds the fully-supervised version of the methodology developed in Chapter 2 within a greedy-forward algorithm, validating stepwise inclusion and exclusion of variables from the relevant subset via a robust information criterion. The second one resorts to the theory of maximum likelihood and irrelevance, defining an objective function in which the subset of relevant variables is regarded as a parameter to be estimated. Chapter 5 concludes the manuscript, summarizing the main contributions and emphasizing future research developments.

Contents

List of Figures

List of Tables

"I want to grow. I want to be better. You grow. We all grow. We're made to grow. You either evolve or you disappear."
— **Tupac Shakur**

Chapter 1

Introduction

Classification is an instinctive task performed by human beings. Hundreds of times per day, our brain unconsciously classifies objects, creatures and actions based on our past experience. The learning process we undertook as children was mostly driven by explicit involvement and direct evidence, from which memories were produced and, ultimately, knowledge created. As a confirmation of our empirical-based knowledge, one may think of the, seemingly naive, task of distinguishing whether the animal in Figure 1.1a is a cat or a dog. Our brain effortlessly and instantly classifies it as a cat, and we are not even aware of the involvement of discriminating features, such as pointy ears and whiskers, that made the identification so easy. Simply, we have seen so many examples of cats and dogs in the past that as soon as we are faced with a new "standard" sample we are immediately able to recognize its breed. The same task is not accomplished as smoothly when focusing on the cat [1] depicted in Figure 1.1b: our stereotypical

(A) Clearly, a cat

(B) Another cat

FIGURE 1.1: Two pictures of cats, one (left panel) displays stereotypical traits, the other (right panel) possesses less common features.

idea of "catness" does not properly match with the kitten's features, and therefore correctly recognizing it as a cat results more difficult.
Moreover, imagine we were to teach someone how to distinguish between the two pets by only showing him/her a (limited) number of labelled pictures of cats and dogs, respectively. That is, a discriminating rule needs to be developed by direct exposition of samples from the two groups, with no explicit mention to group-distinctive traits. Probably, considering this scenario, we would not include Atchoum in the set of photos to be displayed, as its characteristics are likely to weaken rather than improving the empirical understanding on how to distinguish a cat from a dog. Likewise, when the learning entity is a computer system, the input data quality highly influences the output results: a concept that is known in the literature as "Garbage In, Garbage Out" principle.
The present manuscript investigates and proposes innovative solutions to situations where Atchoum-like samples are present in both input and output of a classification task, with the

[1] Its name is Atchoum, and it has an Instagram account

final aim of developing a set of *robust* classification rules. A statistical methodology is robust when it performs reliable inference even if model assumptions are not entirely met by the analysed dataset (Hampel et al., 2005). The approach adopted throughout this monograph is the so called *model-based framework for classification*, whose definition involves the following three elements (Bouveyron et al., 2019):

1. the usage of a finite mixture, that is the natural probabilistic framework for heterogeneous data modeling,
2. the estimation of model parameters, which is achieved via a well-defined statistical method,
3. the adoption of a probability-based rule to perform a-posteriori classification, conditionally on the estimated model.

The main methodological notions for model-based classification and robustness are reviewed in the upcoming Sections. The remainder of the chapter is organized as follows: in Section 1.1 the finite mixture model is introduced, covering its definition and estimation, specifically for the Gaussian case. Model-Based classification is formally characterized in Section 1.2, by presenting the notions of discriminant analysis and semi-supervised classification. Section 1.3 comprises a (non-exhaustive) list of difficulties that are encountered when performing model-based classification. Section 1.4 examines the, increasingly current, topic of performing classification with high-dimensional data, focusing on the variable selection framework. Section 1.5 concludes this introductory chapter describing the main contributions of the present monograph to the model-based classification literature, with the aim of addressing some of the open problems that were highlighted during Section 1.3.

1.1 Finite Mixture Models

Finite mixtures of distributions are a mathematical-based approach to account for heterogeneity in a population. They provide a flexible and convenient semi-parametric framework for modeling unknown distributional shapes, with a possibly different goal; being it classification, density estimation or many more (McLachlan and Peel, 2004). By appropriately choosing its components, a finite mixture is able to model quite complex phenomena, effectively handling situations in which a single parametric family fails in providing a satisfactory fit. Their first appearance goes back all the way to Pearson, 1894, and since then countless research papers on the topic have been produced.
Denote with $\mathbf{y}_1, \dots, \mathbf{y}_M$ a set of M multivariate data. The p-dimensional observation $\mathbf{y}_m$, $m = 1, \dots, M$, is supposed to be a realization of a continuous random vector $\mathcal{Y} \in \mathbb{R}^p$, with probability density function $f(\mathbf{y}_m)$. When $f(\mathbf{y}_m)$ can be written in the form

$$f(\mathbf{y}_m) = \sum_{g=1}^{G} \tau_g f_g(\mathbf{y}_m; \boldsymbol{\theta}_g) \tag{1.1}$$

it is called a *G-component finite mixture* density. In Equation (1.1), $f_g(\cdot; \boldsymbol{\theta}_g)$ is the density of the gth mixture component parametrized by $\boldsymbol{\theta}_g$. The mixing proportion, or weight, τ_g represents the probability that an observation was generated by the gth component, with $\tau_g \geq 0$ for $g = 1, \dots, G$ and $\sum_{g=1}^{G} \tau_g = 1$. Altogether then, the probability distribution of $\mathbf{y}_m$ is a weighted average of G component densities $f_g(\cdot; \boldsymbol{\theta}_g)$, $g = 1, \dots, G$. Commonly, the mixture components are specified to belong to the same parametric family, so that $f_g(\cdot; \boldsymbol{\theta}_g) = f(\cdot; \boldsymbol{\theta}_g)$ $\forall g$, $g = 1, \dots, G$. In the upcoming Subsections, as well as throughout the rest of the manuscript, we will focus on the most commonly and widely employed normal mixtures.

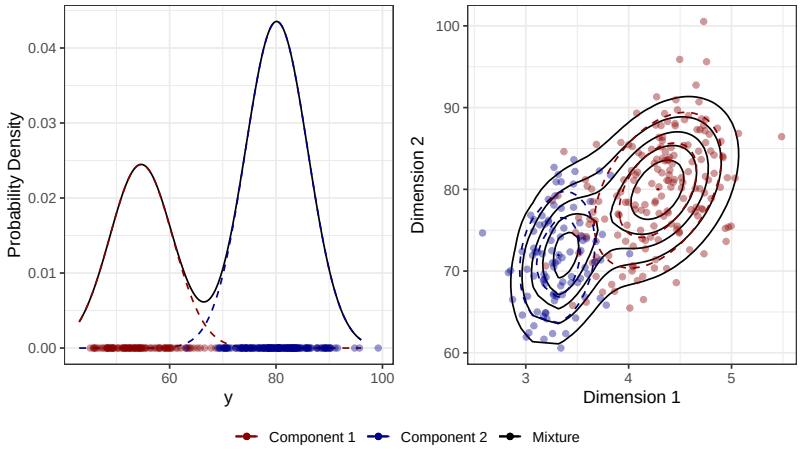

FIGURE 1.2: Probability density function for a one-dimensional univariate finite normal mixture (left panel) and contours of the density function for a two-dimensional finite bivariate normal mixture (right panel) with two mixture components. The dots show two samples simulated from the respective density, with the colors indicating the mixture component from which they were generated.

1.1.1 Gaussian Mixture Models

In the previous Section we have defined the general form of a mixture model. In practice, the components $f(\cdot;\boldsymbol{\theta}_g)$ are often considered to belong to the normal family, leading to the so-called *Gaussian Mixture Model* (GMM). Specifically, when $p = 1$, $\mathbf{y}_m$ is one dimensional and the gth component is a $N(\mu_g, \sigma_g^2)$ density function with $\boldsymbol{\theta}_g = (\mu_g, \sigma_g)$ component-wise mean and standard deviation. When $p \geq 2$, $f(\cdot;\boldsymbol{\theta}_g)$ is the multivariate normal distribution parametrized by its mean vector $\boldsymbol{\mu}_g$ and by its covariance matrix $\boldsymbol{\Sigma}_g$ and has the form

$$\phi\left(\mathbf{y}_m;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\right) = \left|2\pi\boldsymbol{\Sigma}_g\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)^T \boldsymbol{\Sigma}_g^{-1}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)\right\}. \tag{1.2}$$

The density of a Gaussian mixture model for observation $\mathbf{y}_m$ is therefore denoted as follows:

$$f(\mathbf{y}_m) = \sum_{g=1}^{G} \tau_g \phi(\mathbf{y}_m;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g). \tag{1.3}$$

Figure 1.2 shows an example of the density function for a one-dimensional normal mixture (left panel) and density contours for a two-dimensional bivariate Gaussian mixture (right panel) with two mixture components.

1.1.2 Maximum Likelihood estimation

Given an *i.i.d.* sample $\mathbf{y}_1,\ldots,\mathbf{y}_M$ from (1.3), a natural way for estimating the set of model parameters $\boldsymbol{\Theta} = \{\tau_1,\ldots,\tau_G,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_G,\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_G\}$ is via maximum likelihood (ML), with the log-likelihood being

$$\ell_O(\boldsymbol{\tau},\boldsymbol{\mu},\boldsymbol{\Sigma}|\mathbf{Y}) = \sum_{m=1}^{M} \log\left[\sum_{g=1}^{G} \tau_g \phi\left(\mathbf{y}_m;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g\right)\right]. \tag{1.4}$$

In mixture models however, the direct maximization of the objective function in (1.4) is intractable, and the Expectation-Maximization or EM algorithm (Dempster et al., 1977) is the standard approach used for tackling the problem. The EM algorithm is a general framework for maximum likelihood estimation, specifically tailored for incomplete-data scenarios, that is when units are (or can be thought of being) only partially observed (McLachlan and Krishnan, 2008). This situation naturally arises in mixture models, since we observe realizations $\mathbf{y}_m$, $m = 1, \ldots, M$, but we do not know from which component density they were originated. In this context then, we call the quantity in (1.4) the *observed data log-likelihood* (explicitly indicated by the subscript O), as it includes in the specification only observed quantities. The estimation procedure is carried out considering instead the "complete data" $(\mathbf{y}_m, \mathbf{z}_m)$, $m = 1, \ldots, M$, where $\mathbf{z}_m = (z_{m1}, \ldots, z_{mG})$ is the unobserved portion of the data, with

$$z_{mg} = \begin{cases} 1 & \text{if } \mathbf{y}_m \text{ belongs to group } g \\ 0 & \text{otherwise.} \end{cases} \tag{1.5}$$

$\mathbf{z}_m$ are taken to be independent and identically distributed according to a multinomial distribution of one draw with G categories, each with probability τ_g, $g = 1, \ldots, G$ and $\sum_{g=1}^{G} \tau_g = 1$. With this extra piece of information, we can define the *complete data log-likelihood*:

$$\ell_C(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{Y}, \mathbf{z}) = \sum_{m=1}^{M} \sum_{g=1}^{G} z_{mg} \log \left[\tau_g \phi \left(\mathbf{y}_m ; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g \right) \right]. \tag{1.6}$$

Notice that the log-likelihood specification in (1.6) includes not only the realizations $\mathbf{y}_m$, but, through $\mathbf{z}_m$, also the associated component density from which each unit was generated. A typical situation in which the complete data log-likelihood is directly available is model-based classification (see Section 1.2).

In maximizing (1.4), the EM algorithm alternates between two steps. The expectation step (E-step) computes the conditional expectation of the unobserved data given the observed data and the current parameter estimates: $\hat{z}_{mg} = E\left[z_{mg} | \mathbf{y}_m ; \boldsymbol{\Theta} \right]$. The E-step at the $(k+1)$th iteration of the EM algorithm for mixture models reads:

$$\hat{z}_{mg}^{(k+1)} = \frac{\hat{\tau}_g^{(k)} \phi \left(\mathbf{y}_m ; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)} \right)}{\sum_{j=1}^{G} \hat{\tau}_j^{(k)} \phi \left(\mathbf{y}_m ; \hat{\boldsymbol{\mu}}_j^{(k)}, \hat{\boldsymbol{\Sigma}}_j^{(k)} \right)} \tag{1.7}$$

where $\hat{\tau}_g^{(k)}$, $\hat{\boldsymbol{\mu}}_g^{(k)}$ and $\hat{\boldsymbol{\Sigma}}_g^{(k)}$ are the estimated values respectively for τ_g, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ after the kth EM iteration.

The maximization step (M-step) determines the parameters value that maximizes the expected log-likelihood obtained from the E-step. Closed-form solutions are obtained in the M-step for the mixing proportion and the mean vector of each Gaussian component:

$$\begin{aligned} \hat{\tau}_g^{(k+1)} &= \frac{\hat{m}_g^{(k+1)}}{M}; \\ \hat{\boldsymbol{\mu}}_g^{(k+1)} &= \frac{\sum_{m=1}^{M} \hat{z}_{mg}^{(k+1)} \mathbf{y}_m}{\hat{m}_g^{(k+1)}} \end{aligned} \tag{1.8}$$

where $\hat{m}_g^{(k+1)} = \sum_{m=1}^{M} \hat{z}_{mg}^{(k+1)}$, $g = 1, \ldots, G$. Estimation for the covariance matrices may depend

on their parametrization (see Section 1.2.1); when, for instance, heteroscedastic covariance matrices are considered, the M-step update is obtained as follows:

$$\hat{\boldsymbol{\Sigma}}_g^{(k+1)} = \frac{\sum_{m=1}^{M} \hat{z}_{mg}^{(k+1)} \left(\mathbf{y}_m - \hat{\boldsymbol{\mu}}_g^{(k+1)}\right)\left(\mathbf{y}_m - \hat{\boldsymbol{\mu}}_g^{(k+1)}\right)^T}{\hat{m}_g^{(k+1)}}. \tag{1.9}$$

Dempster et al., 1977 have proved the convergence of the EM algorithm to a local maximum of the log-likelihood function, with the objective function not decreasing after each EM iteration. A standard approach for validating whether the procedure has converged is to monitor the increase in the log-likelihood between the last two iterations, typically an increase of less than 10^{-5} is deemed sufficient to stop the algorithm, retaining the estimates from the last iteration as to be the final ML estimates.
Being the likelihood not convex in general, the converged estimate may depend on the initial chosen values. The problem of properly initializing the EM algorithm is particularly critical in the clustering context, where no prior information regarding the group structure is available. Since the focus of the present monograph is supervised and semi-supervised learning rather than clustering, we will overlook the EM initialization issue at the moment, postponing its treatment in the dedicated Section of adaptive learning (see Chapter 3). Nonetheless, the interested reader is referred to Biernacki et al., 2003, Maitra, 2009, Scrucca and Raftery, 2015 and references therein for a thorough treatment of the problem.

1.2 Model-Based Classification

In this Section we review the main concepts of supervised classification based on mixture models, with particular focus on Eigenvalue Decomposition Discriminant Analysis and its semi-supervised formulation, as introduced in Dean et al., 2006, since they are the basis of the novel robust semi-supervised classifier introduced in Chapter 2. Other popular model-based classification methods, not reviewed here for sake of brevity, include Regularized Discriminant Analysis (Friedman, 1989) and Mixture Discriminant Analysis (Hastie and Tibshirani, 1996; Fraley and Raftery, 2002).

1.2.1 Eigenvalue Decomposition Discriminant Analysis

Model-based discriminant analysis (McLachlan, 1992; Fraley and Raftery, 2002) is a probabilistic approach for supervised classification, in which a classifier is built from a complete set of learning observations $\{(\mathbf{x}_1, \mathbf{l}_1), \ldots, (\mathbf{x}_N, \mathbf{l}_N)\}$; where $\mathbf{x}_n$ and $\mathbf{l}_n$, $n = 1, \ldots, N$, are independent realizations of random vectors $\mathcal{X} \in \mathbb{R}^p$ and $\mathcal{G} \in \{1, \ldots, G\}$, respectively. That is, $\mathbf{x}_n$ denotes a p-variate observation and $\mathbf{l}_n$ its associated class label, such that $l_{ng} = 1$ if observation n belongs to group g and 0 otherwise, $g = 1, \ldots, G$. Considering a Gaussian framework, and following the notation introduced in Section 1.1, the probabilistic mechanism that is assumed to have generated the data is as follows:

$$\begin{aligned} \mathcal{G} &\sim Mult_G(1; \tau_1, \ldots, \tau_G) \\ \mathcal{X}|\mathcal{G} = g &\sim \mathcal{N}_p(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \end{aligned} \tag{1.10}$$

Therefore, the joint density of $(\mathbf{x}_n, \mathbf{l}_n)$ is given by:

$$f(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\Theta}) = \prod_{g=1}^{G} \left[\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]^{l_{ng}}. \tag{1.11}$$

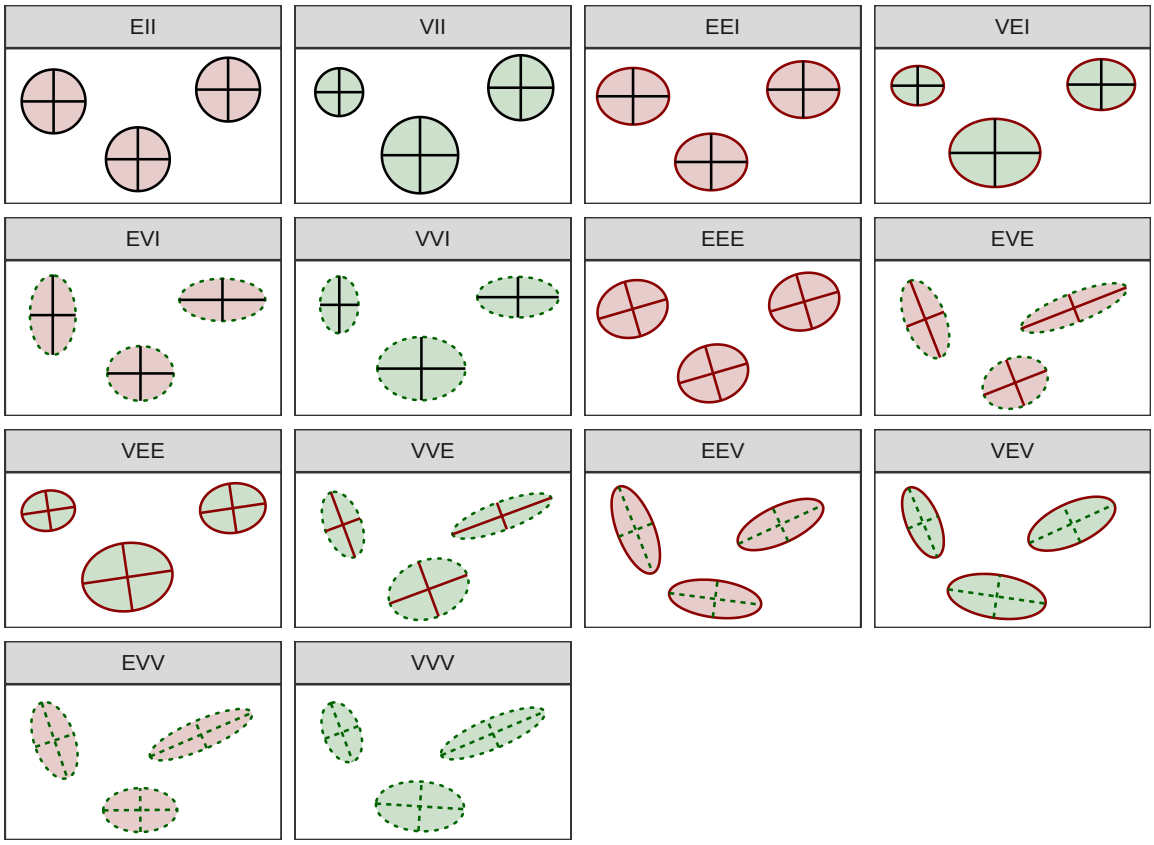

FIGURE 1.3: Ellipses of isodensity for each of the 14 Gaussian models obtained by eigen-decomposition in case of three groups in two dimensions. Green (red) area denotes variable (equal) volume across components. Dashed green (solid red) perimeter denotes variable (equal) shape across components. Dashed green (solid red) axes denote variable (equal) orientation across components. Solid black perimeter denotes spherical shape. Solid black axes denote axis-aligned orientation.

Notice that (1.11) is closely tied with the complete-data log-likelihood introduced in (1.6): clearly, there is no incomplete-data in the learning set since we observe both the realization $\mathbf{x}_n$ and its associated component via $\mathbf{l}_n$, $\forall n, n = 1, \dots, N$. The marginal density for $\mathbf{x}_n$ can be obtained by integrating the class labels out of the joint density in (1.11), in doing so we retrieve the mixture model defined in (1.3).
Discriminant analysis makes use of data with known labels to estimate model parameters for creating a classification rule. The trained classifier is subsequently employed for assigning a set of unlabelled observations $\mathbf{y}_m$, $m = 1, \dots, M$ to the class g with the associated highest posterior probability:

$$z_{mg} = \mathbb{P}(\mathcal{G} = g | \mathcal{X} = \mathbf{y}_m) = \frac{\tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)}{\sum_{j=1}^{G} \tau_j \phi(\mathbf{y}_m; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{1.12}$$

using the maximum a posteriori (MAP) rule. The afore-described framework is widely employed in classification tasks, thanks to its probabilistic formulation and well-established efficacy.
Observing that the number of parameters in the component covariance matrices in the joint density (1.11) grows quadratically with the dimension p, Bensmail and Celeux, 1996 introduced a parsimonious parametrization. They proposed to enforce additional assumptions on the matrices structure, based on the eigen-decomposition of Banfield and Raftery, 1993 and Celeux and Govaert, 1995:

$$\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{D}_g \boldsymbol{A}_g \boldsymbol{D}_g^{'} \tag{1.13}$$

where $\boldsymbol{D}_g$ is an orthogonal matrix of eigenvectors, $\boldsymbol{A}_g$ is a diagonal matrix such that $|\boldsymbol{A}_g| = 1$ and $\lambda_g = |\boldsymbol{\Sigma}_g|^{1/p}$. These elements correspond respectively to the orientation, shape and volume (alternatively called scale) of the different Gaussian components. Allowing each parameter in (1.13) to be equal or different across groups, Bensmail and Celeux, 1996 define a family of 14 patterned models, listed in Table 1.1 and graphically represented in Figure 1.3. Such class of models is particularly flexible, as it includes very popular classification methods like Linear Discriminant Analysis and Quadratic Discriminant Analysis as special cases for the EEE and VVV models, respectively (Hastie and Tibshirani, 1996).
The computation needed to get ML estimates in discriminant analysis is equivalent to a single M-step of the EM algorithm described in Section 1.1.2. The covariance estimate depends on its parametrization, details of the M-step for each of the 14 patterned model is given in Celeux and Govaert, 1995. Eigenvalue Decomposition Discriminant Analysis (EDDA) is implemented in the `mclust` `R` package (Scrucca et al., 2016).

1.2.2 Semi-supervised classification

Exploiting the assumption that the data generating process outlined in (1.10) is the same for both labelled and unlabelled observations, Dean et al., 2006 propose to include also the data whose memberships are unknown in the parameter estimation. That is, information about group structure that may be contained in both labelled and unlabelled samples is combined in order to improve the classifier performance, in a semi-supervised manner.

TABLE 1.1: Nomenclature, characteristics and parametrization of the covariance matrices $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, G$ in the EDDA family of Gaussian parsimonious models.

Model	Volume	Shape	Orientation	$\boldsymbol{\Sigma}_g$
EII	Equal	Spherical	-	$\lambda \boldsymbol{I}$
VII	Variable	Spherical	-	$\lambda_g \boldsymbol{I}$
EEI	Equal	Equal	Axis-aligned	$\lambda \boldsymbol{A}$
VEI	Variable	Equal	Axis-aligned	$\lambda_g \boldsymbol{A}$
EVI	Equal	Variable	Axis-aligned	$\lambda \boldsymbol{A}_g$
VVI	Variable	Variable	Axis-aligned	$\lambda_g \boldsymbol{A}_g$
EEE	Equal	Equal	Equal	$\lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}'$
VEE	Variable	Equal	Equal	$\lambda_g \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}'$
EVE	Equal	Variable	Equal	$\lambda \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}'$
EEV	Equal	Equal	Variable	$\lambda \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g'$
VVE	Variable	Variable	Equal	$\lambda_g \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}'$
VEV	Variable	Equal	Variable	$\lambda_g \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g'$
EVV	Equal	Variable	Variable	$\lambda \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g'$
VVV	Variable	Variable	Variable	$\lambda_g \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g'$

Under the framework defined in Section 1.2.1, and given the set of available information $\{(\mathbf{x}_n, \mathbf{l}_n)|n = 1, \ldots, N\} \cup \{\mathbf{y}_m|m = 1, \ldots, M\}$, the *observed log-likelihood* is

$$\begin{aligned}\ell_O(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X}, \mathbf{Y}, \mathbf{l}) = &\sum_{n=1}^{N}\sum_{g=1}^{G} l_{ng} \log\left[\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right] + \\ &+ \sum_{m=1}^{M} \log\left[\sum_{g=1}^{G} \tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]\end{aligned} \tag{1.14}$$

in which both labelled and unlabelled samples are accounted for in the likelihood definition. Notice that, compare to the standard observed log-likelihood for mixture models defined in (1.4), here we include in addition the contribution given by the labelled set $\{(\mathbf{x}_1, \mathbf{l}_1), \ldots, (\mathbf{x}_N, \mathbf{l}_N)\}$. Treating the (unknown) labels z_{mg}, $m = 1, \ldots, M$, $g = 1, \ldots, G$ as missing data and including them in the likelihood specification defines the so called *complete-data log-likelihood* (see Section 1.1.2):

$$\begin{aligned}\ell_C(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X}, \mathbf{Y}, \mathbf{l}, \mathbf{z}) = &\sum_{n=1}^{N}\sum_{g=1}^{G} l_{ng} \log\left[\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right] + \\ &+ \sum_{m=1}^{M}\sum_{g=1}^{G} z_{mg} \log\left[\tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]\end{aligned} \tag{1.15}$$

Maximum likelihood estimates for (1.14) are obtained through a slight modification of the EM algorithm described in Section 1.1.2. The unlabelled data are then classified according to $\hat{z}_{mg}$, using the MAP. The updating classification rules was demonstrated to give improved classification performance over the classical model-based discriminant analysis in some food authenticity applications, particularly when the training size is small. An implementation of this can be found in the `upclass` `R` package (Russell et al., 2014).

1.2.3 Model selection

In most situations designing a classifier involves a selection step: particularly, for the models described in Section 1.2.1 and 1.2.2 this refers to the selection of the "best" parsimonious structure out of the 14 patterned models listed in Table 1.1. The definition of "best model" depends on the context: often, in classification, good overall prediction accuracy is of greatest interest, and, therefore, the final goal would be to minimize the misclassification error in the test set. The most widely data-driven approach used for estimating the misclassification error of a classifier is *cross-validation* (CV). Cross-validation involves partitioning the training set into complementary subsets, fitting the model on a subset and assessing its performance on the other subset, see, for instance, Section 7.10 in Hastie et al., 2009. Alternatively, in the model-based classification setting, penalized log-likelihood criteria, such as the Bayesian Information Criterion (BIC) (Schwarz, 1978) or the Akaike criterion (AIC) (Akaike, 1974) are at our disposal.
For the novel models developed in the remaining chapters of this thesis, robust penalized log-likelihood criteria will be used for performing model selection (see Sections 2.2.4 and 3.2.5). The rationale behind this choice is twofold: clearly, the computational cost needed for executing cross-validation is much higher than the one necessary for computing an information criterion, but there is also another, more crucial, argument to be made. As anticipated in the introductory section of this chapter, the main goal of the present monograph is to develop classifiers that perform well even when dealing with real-data complexities. Particularly, the problem of uncertain labels (see Subsection 1.3.3) has a yet unpredictable effect on cross-validation. Given a well-trained classifier, imagine that some mislabelled units fall in the validation set: they will be correctly assigned to their true group in contradiction to their (wrong) label, so misclassification error will be biased. Robust prediction loss functions, such as the root trimmed mean squared prediction error (RTMSPE), coupled with cross-validation have been successfully considered for model selection in robust regression (Alfons et al., 2013). Unfortunately, when there is label noise, how to modify CV for validating a classifier is yet to be understood. Further research is needed to develop a coherent and reliable data-driven method with adulterated data.
To conclude, we recall the general definition for the Bayesian Information Criterion (BIC):

$$BIC = 2\ell_O\left(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right) - v_{XXX}\log\left(T\right) \tag{1.16}$$

where $\ell_O\left(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right)$ denotes the maximized observed data log-likelihood, v_{XXX} is a penalty term equal to the number of parameters to be estimated according to the model chosen in Table 1.1 and T is the number of observations considered for model fitting. Following the notation introduced in the previous Sections, $T = N$ for the EDDA model and $T = N + M$ for its semi-supervised version. The covariance structure that leads to the highest BIC value is ultimately selected. From a theoretical viewpoint it is well known that, in a Bayesian framework, the BIC approximates the log evidence of the postulated model (Schwarz, 1978; Kass, 1993; Kass and Raftery, 1995). Even though no corresponding theory has yet been established for the "trimmed" counterpart of (1.16), a discussion and some initial attempts in justifying the usage of trimmed information criteria in the context of robust model selection is reported in Appendix D.

1.3 Learning in presence of real-data complexities

Having defined the general framework for model-based supervised and semi-supervised learning, the present Section lists a set of difficulties that are often encountered in performing real-data classification. Particularly, the need of overcoming some (or all) of the below-described

problems at once with a single methodology was the main motivation that led to the development of the novel models described in the remaining Chapters.

1.3.1 Outliers

In statistical analysis, an exact definition of an outlier often depends on hidden assumptions regarding the data structure and the applied detection method (Maimon and Rokach, 2005). Hawkins, 1980 defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Barnett and Lewis, 1974 indicate that an outlying observation is one that appears to deviate markedly from other members of the sample in which it occurs. Equivalently, Johnson et al., 2002 regard as outlier an observation in a dataset which appears to be inconsistent with the remainder of that set of data. In a classification context, outlying units (or units with attribute noise) present contamination in the exploratory variables, that is, they display unusual values on their predictors (Wu, 1995). Apart from these quite general specifications, the concept of outlier, unfortunately, does not possess an unanimously accepted, rigorous mathematical definition (Ritter, 2014).
Generally, three main approaches can be employed when building a classifier from a noisy dataset: cleaning the data, modeling the noise and using robust estimators of model parameters (Bouveyron and Girard, 2009). To this extent, trimming is probably the earliest safeguard against outliers, as it was employed already toward the end of the nineteenth century (Newcomb, 1886). Firstly introduced in the clustering analysis literature by Cuesta-Albertos et al., 1997, *impartial trimming* is a data-driven procedure that excludes observations identified as outliers when estimating the model parameters. The final result of a statistical method that employs impartial trimming is therefore a set of robustly estimated parameters, as well as a flag that identifies the "most outlying" observations according to the postulated model. Interestingly, to our best knowledge, impartial trimming had never been (directly) used in a classification context: the methodology described in Chapter 2 takes advantage of its flexibility for jointly dealing with outliers and uncertain labels. Given the importance played by impartial trimming in the upcoming Chapters, we hereafter briefly describe its usage in robustifying the EM algorithm for model-based clustering, along the lines of García-Escudero et al., 2008.
The considered methodological framework is the *spurious-outliers model* (Gallegos and Ritter, 2005), for which it is assumed that the data contains $\lfloor M\alpha \rfloor$ "spurious" observations, $\alpha \in (0,1)$, that should not be included in estimating the mixture model. Such probabilistic structure is also employed in providing theoretical justification for the robust variable selection method developed in Section 4.2.1, therefore, its detailed treatment is postponed to the fourth chapter of the present manuscript. Here, we limit to recall its definition and main properties. Following the notation introduced in Section 1.1, the spurious mixture model is defined through its likelihood:

$$L(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}|\mathbf{Y}) = \prod_{m=1}^{M} \left[\prod_{g=1}^{G} \tau_g \phi \left(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g \right) \right]^{z_{mg}\varphi(\mathbf{y}_m)} \times \left[w \left(\mathbf{y}_m; \boldsymbol{\psi}_m \right) \right]^{1-\varphi(\mathbf{y}_m)}, \tag{1.17}$$

where $\varphi(\cdot)$ is a 0–1 indicator function that expresses whether observation $\mathbf{y}_m$ is regarded as spurious or not, and $w(\cdot; \boldsymbol{\psi}_m)$ is a generic probability density function in $\mathbb{R}^p$, parametrized by $\boldsymbol{\psi}_m \in \boldsymbol{\Psi}_m$. That is, if the m^{th} observation is contaminated, it is generated from an almost arbitrary subject-specific distribution $w(\cdot; \boldsymbol{\psi}_m)$. Under weak assumptions on the contaminating distribution $w(\cdot; \boldsymbol{\psi}_m)$, Gallegos and Ritter, 2005 and García-Escudero et al., 2008 show that mixture parameters and group memberships can be inferred by just maximizing the *trimmed*

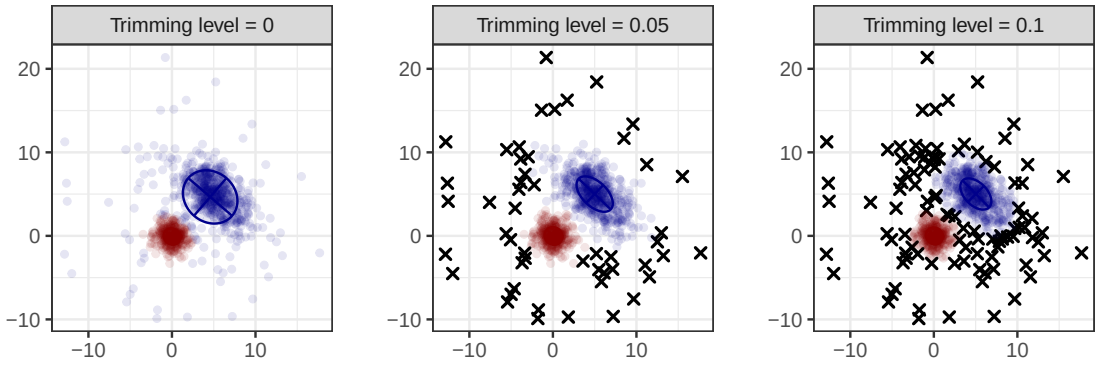

FIGURE 1.4: Scatter plot from fitting with the tclust algorithm a two-dimensional finite bivariate normal mixture with two components to a noisy dataset, varying trimming level. The colors indicate the assigned group, trimmed observations are denoted by $\times$.

log-likelihood (Neykov et al., 2007):

$$\ell_{trim}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{z}|\mathbf{Y}) = \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \sum_{g=1}^{G} z_{mg} \log \left[\tau_g \phi\left(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right)\right]. \tag{1.18}$$

The tclust algorithm (García-Escudero et al., 2008; Fritz et al., 2013) serves the purpose of maximizing (1.18): it can be seen as a CEM-like algorithm (Celeux and Govaert, 1992) with an additional Concentration Step (Rousseeuw and Driessen, 1999). The tclust algorithm operates as follows at iteration k:

- Concentration Step: The trimming procedure is implemented by discarding the $\lfloor M\alpha \rfloor$ observations $\mathbf{y}_m$ with smaller values of

$$D\left(\mathbf{y}_m; \hat{\boldsymbol{\Theta}}^{(k)}\right) = \sum_{g=1}^{G} \hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right) \quad m = 1, \ldots, M. \tag{1.19}$$

  That is, $\varphi(\mathbf{y}_m) = 0$ in (1.18) for such observations.

- Expectation Step: For each non-trimmed observation $\mathbf{y}_m$, compute the posterior probabilities as per the E-step described in Section 1.1.2, hard assigning each $\mathbf{y}_m$ to the group component for which $\hat{z}_{mg}^{(k+1)}$ is highest.

- Constrained Maximization Step: the mixture parameters are updated based only on the non-spurious observations, i.e., units for which $\varphi(\mathbf{y}_m) = 1$, $m = 1, \ldots, M$. Here, an additional restriction may be enforced (see Section 1.3.2 for details).

The positive impact impartial trimming has in robustifying parameter estimates and detecting outlying observations is presented in Figure 1.4.
Several other methods for performing model-based classification robust to outlying observations were proposed in the literature. Hawkins and McLachlan, 1997 developed a high-breakdown criterion to linear discriminant analysis, involving the identification of the data subset whose deletion leads to the smallest determinant for the within-group covariance matrix. S-estimators and the re-weighted MCD estimator for multivariate location and dispersion matrices were respectively employed in He and Fung, 2000 and Hubert and Van Driessen, 2004

for building a robust discriminant rule. Lastly, Vanden Branden and Hubert, 2005 provide a framework for high-dimensional classification by means of a Robust Soft Independent Modelling of Class Analogies (RSIMCA) method.

1.3.2 Degeneracies

Degenerate solutions generally arise when fitting mixture models with the EM algorithm, rather than when a model-based approach is employed for classification. Nevertheless, it becomes paramount to be protected against singular and spurious solutions whenever a semi-supervised or an adaptive learner, i.e., a model able to look for unobserved extra classes, is considered (see 1.3.4 and the method developed in Chapter 3).
Extensive literature has been devoted to studying the appearance of the so-called degenerate solutions that may be provided by the EM algorithm when fitting a finite mixture to a set of data (Peel and McLachlan, 2000; Biernacki, 2007; Ingrassia and Rocci, 2011). This is due to the likelihood function itself, rather than being a shortcoming of the EM procedure: it is easy to show that for elliptical mixture models with unrestricted covariance matrices the associated likelihood is unbounded (Day, 1969); simply take $\boldsymbol{\mu}_1 = \mathbf{y}_1$ and $|\boldsymbol{\Sigma}_1| \to 0$ in Equation (1.3). Therefore, the maximization of (1.4) without any constraint is an ill-posed mathematical problem. An even more subtle issue, at least from a practitioner perspective, is the appearance of solutions that are not exactly degenerate, but they can be regarded as spurious since they lie very close to the boundary of the parameter space. They occur when an estimated component has a very small generalized variance, fitting few data points almost lying on a lower-dimensional space (Peel and McLachlan, 2000). Spurious solutions often display a high likelihood value, however, little insight can be obtained in real-world applications as they are mostly a result of modeling a localized random pattern rather than a true underlying group. In general, methods for dealing with this issue can be grouped in constraint methods, Bayesian methods, penalty methods, and others. We will review the first approaches, as they are an essential part of the methodologies introduced in Chapters 2 and 3. A comprehensive list of references for the remaining ones can be found in García-Escudero et al., 2018b.
The seminal paper of this strand is Hathaway, 1985, where he proposed to maximize the likelihood subject to the constraints that the eigenvalues of $\boldsymbol{\Sigma}_g \boldsymbol{\Sigma}_h^{-1}$ be grater than or equal to some minimum value $c' > 0$, $1 \leq g \neq h \leq G$. Unfortunately, no dedicated algorithm was proposed for carrying out the associated constrained maximization. Another type of constraint, based on the Löwner matrix ordering ($\preceq$) was proposed by Gallegos and Ritter, 2009, requiring the scatter matrix to satisfy $\boldsymbol{\Sigma}_g \preceq c^{-1}\boldsymbol{\Sigma}_h$ for every g and h. Nonetheless, a specific algorithm for solving this type of problem in closed form for any fixed value of the constant c is still missing. A conceptually similar proposal of constraining the ratio of the maximum to the minimum of all the eigenvalues of all the mixture component covariance matrices was introduced in García-Escudero et al., 2015. Formally, the authors considered the problem of maximizing (1.4) with the additional *eigenvalues-ratio constraint*:

$$\Pi / \pi \leq c \tag{1.20}$$

where $\Pi = \max_{g=1\ldots G} \max_{l=1\ldots p} d_{lg}$ and $\pi = \min_{g=1\ldots G} \min_{l=1\ldots p} d_{lg}$, with d_{lg}, $l = 1, \ldots, p$, being the eigenvalues of the matrix $\boldsymbol{\Sigma}_g$ and $c \geq 1$ being a fixed constant. The constraint in (1.20) controls the relative variability within and between group populations. It also avoids singularities, providing a well-defined maximization problem for MLE of (1.4). Furthermore, Fritz et al., 2013 provide a computationally efficient and closed form solution for directly enforcing the constraint in (1.20) at each iteration of the EM algorithm. In detail, the authors recast the complex multivariate minimization problem of the original tclust algorithm (García-Escudero

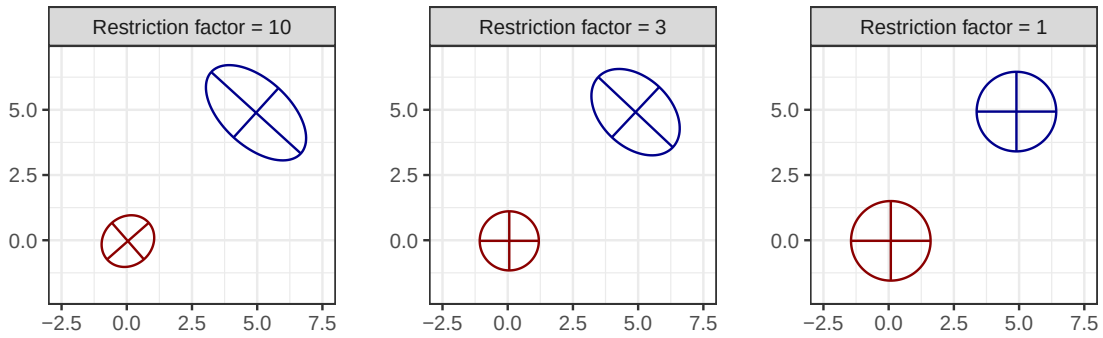

FIGURE 1.5: Bivariate Gaussian equidensity ellipsoids with different eigenvalues-ratio restriction, varying constant c

et al., 2008), whose dimension depends on the value of p and G, with the minimization of a one-dimensional function. A truncation operator is defined on the set of eigenvalues of the covariance matrices, and its minimum is attained by only evaluating a univariate objective function $2Gp + 1$ times.
More formally, in the case of unconstrained scatters (VVV model), a truncated eigenvalue d_{lg}^m is defined as follows:

$$d_{lg}^m = \begin{cases} d_{lg} & \text{if } d_{lg} \in [m, cm] \\ m & \text{if } d_{lg} < m \\ cm & \text{if } d_{lg} > cm \end{cases} \tag{1.21}$$

with m some threshold value. Fritz et al., 2013 proved that for obtaining the best set of truncated eigenvalues that satisfies (1.20) it is sufficient to find m_{opt} that minimizes

$$f(m) = \sum_{g=1}^{G} n_g \sum_{l=1}^{p} \left[\log d_{lg}^m + \frac{d_{lg}}{d_{lg}^m} \right] \tag{1.22}$$

where n_g denotes the sample size of the g-th group at that specific iteration of the EM algorithm. As mentioned above, the minimum of (1.22) is achieved by only evaluating the function in $2Gp + 1$ points. This is proven by firstly rewriting $f(m)$ as follows:

$$\begin{aligned} f(m) = \sum_{g=1}^{G} n_g \Bigg[& \sum_{l=1}^{p} \left(\log m + \frac{d_{lg}}{m} \right) \mathbb{I}(d_{lg} < m) + \sum_{l=1}^{p} \left(\log d_{lg} + 1 \right) \mathbb{I}(m \leq d_{lg} \leq cm) + \\ & + \sum_{l=1}^{p} \left(\log cm + \frac{d_{lg}}{cm} \right) \mathbb{I}(d_{lg} > cm) \Bigg]. \end{aligned} \tag{1.23}$$

Being (1.22) a continuous differentiable function, m_{opt} is one of those values m^* that set its first derivative to 0:

$$
\begin{aligned}
f'(m) = 0 \Leftrightarrow \sum_{g=1}^{G} n_g \Bigg[\sum_{l=1}^{p} \left(\frac{1}{m} - \frac{d_{lg}}{m^2} \right) \mathbb{I}(d_{lg} < m) + \\
+ \sum_{l=1}^{p} \left(\frac{1}{m} - \frac{d_{lg}}{cm^2} \right) \mathbb{I}(d_{lg} > cm) \Bigg] = 0 \\
\sum_{g=1}^{G} n_g \left[\sum_{l=1}^{p} (m - d_{lg}) \, \mathbb{I}(d_{lg} < m) + \sum_{l=1}^{p} (m - d_{lg}/c) \, \mathbb{I}(d_{lg} > cm) \right] = 0 \\
\sum_{g=1}^{G} n_g \left[\sum_{l=1}^{p} (m) \, \mathbb{I}(d_{lg} < m) - \sum_{l=1}^{p} (d_{lg}) \, \mathbb{I}(d_{lg} < m) + \right. \\
\left. + \sum_{l=1}^{p} (m) \, \mathbb{I}(d_{lg} > cm) - \sum_{l=1}^{p} (d_{lg}/c) \, \mathbb{I}(d_{lg} > cm) \right] = 0 \\
\sum_{g=1}^{G} n_g \left[\sum_{l=1}^{p} (m) \left(\mathbb{I}(d_{lg} < m) + \mathbb{I}(d_{lg} > cm) \right) + \right. \\
\left. - \sum_{l=1}^{p} (d_{lg}) \left(\mathbb{I}(d_{lg} < m) + \frac{1}{c} \mathbb{I}(d_{lg} > cm) \right) \right] = 0 \quad (1.24)
\end{aligned}
$$

Noticing that each term in (1.24) is greater than 0, equation (1.24) is satisfied for all $m^* > 0$ such that:

$$
m^* = \frac{\sum_{g=1}^{G} n_g \left(\sum_{l=1}^{p} d_{lg} (\mathbb{I}(d_{lg} < m) + \sum_{l=1}^{p} \frac{d_{lg}}{c} \mathbb{I}(d_{lg} > cm) \right)}{\sum_{g=1}^{G} n_g \sum_{l=1}^{p} \left(\mathbb{I}(d_{lg} < m) + \mathbb{I}(d_{lg} > cm) \right)}. \quad (1.25)
$$

Now consider the sequence $u_1 \leq u_2 \leq \cdots \leq u_{2pG}$ obtained by ordering

$$
d_{11}, d_{12}, \ldots, d_{lg}, d_{pG}, d_{11}/c, d_{12}/c, \ldots, d_{lg}/c, d_{pG}/c. \quad (1.26)
$$

Each addend on the right hand-side of equation (1.25) takes constant values in the intervals $(-\infty, u_1], (u_1, u_2], \ldots, (u_{2pG}, \infty)$. Therefore (1.25) is a step function that assumes a finite set of values: m_{opt} is chosen as to be the one in the $2Gp + 1$ distinct entries that minimizes (1.22). A graphical representation of the eigenvalues-ratio restriction, varying values of the constraint c in the multivariate Gaussian equidensity ellipsoids, is reported in Figure 1.5.
For the remaining part of this monograph, the eigenvalues-ratio constraint in (1.20) will be considered for protecting our models against degeneracies and to reduce the occurrence of spurious solutions. Particularly, novel algorithms for combining such constraint within the family of parsimonious Gaussian models listed in Table 1.1 are introduced: details are given in Section 2.8.

1.3.3 Uncertain labels

When building a classifier in a real-data context a second difficulty that is likely to be encountered is the so called *label noise problem*: learning samples with wrongly associated labels (Zhu and Wu, 2004). In this context, the learning scenario is denoted as *imperfectly supervised*: i.e., pattern recognition applications where the assumption of label correctness does not hold for all the elements of the training set (Barandela and Gasca, 2000). Often, a labeled dataset is the output

of human activity: classes are manually assigned by domain experts to a set of given measurements, an example being for instance biomedical applications. Clearly, this kind of heuristic supervision may be imprecise, difficult and/or expensive; and several reasons like subjectivity, data-entry errors or information inadequacy in the identification process may cause a labeling error (Brodley and Friedl, 1999). More specifically, four main potential sources of label noise can be identified (Frénay and Verleysen, 2014). First, lack of information may result in the impossibility of obtaining reliable labeling (Hickey, 1996). Second, the labeling process itself may be prone to errors, especially if automatically performed by a sensitive system (Lagacherie and Holmes, 1997). Third, as previously mentioned, when subjectivity is inherent in the labeling task, as in medical applications (Malossini et al., 2006) and, lastly, label noise can simply come from data encoding or communication problems (Angluin and Laird, 1988).
Frénay and Verleysen, 2014 provide a taxonomy of label noise, characterizing it in terms of probabilistic dependence with respect to feature variables and group memberships. According to the authors, labels are *noisy completely at random* when the occurrence of an error is independent of the other variables, including the true class itself; *noisy at random* when the appearance of label noise depends on the true class and *noisy not at random* when a more complex dependence structure is present in the data. Clearly, if the noise is to be modeled, ad-hoc procedures that depend on the underlying noise structure need to be considered.
Incorrect labels can strongly undermine the classifier performance, especially if the training size is small. Specifically, label noise may badly affect the predicting power of a model, lowering its classification accuracy, and unnecessarily increase model complexity, especially when it comes to detect the most discriminative variables (Zhang et al., 2006). Section 1.4 and Chapter 4 are entirely devoted to this topic. Therefore, robust methods capable of dealing with label noise are critically important in applications. Likewise in dealing with outliers, as seen in Section 1.3.1, this can be generally achieved by cleaning the data, modeling the noise and finally using robust estimations of model parameters. A novel approach of the latter type based on impartial trimming and eigenvalues-ratio constraints is developed in Chapter 2. For a state-of-the-art review and a thorough treatment on class noise, the reader is referred to the recent surveis of Frénay and Verleysen, 2014 and Prati et al., 2019, while the "Supervised Classification with Uncertain Labels" part in Section 5.5 of Bouveyron et al., 2019 provides a comprehensive list of approaches that deal with classification with uncertain labels.

1.3.4 Unobserved classes

The last real-data complexity with whom the present monograph deals with is unobserved classes. The usual framework of supervised classification does not consider the possibility of having test units belonging to a class not previously observed in the learning phase. A classic hypothesis is that the training set contains samples for each and every group within the population of interest. Nevertheless, this strong assumption may not hold true in fields like biology, where novel species may appear and their detection is an important issue, or in social network analysis where communities continuously expand and evolve. Therefore, a classifier suitable for these situations need to adapt to the detection of previously unobserved classes. Unfortunately, standard supervised methods will predict class labels only within the set of groups previously encountered in the learning phase.
A related topic to supervised classification with unobserved classes is called novelty detection, that is, the identification of new or unknown data or signal that a machine learning system is not aware of during training (Markou and Singh, 2003). A first attempt in dealing with this issue in the classification context is considered in Tax and Duin, 1998, where novelties are detected based on the instability of the outputs of different methods, namely mixture of Gaussians, Parzen estimator and nearest neighbor estimator. Another very popular approach is the

Support Vector Method for novelty detection (Schölkopf et al., 2000), in which known and novel objects are identified through the estimation of a separating function, whose functional form is given by a kernel expansion. Even though novelty detection methods are able to identify new or unobserved data points, they lack the ability of recognizing several homogeneous and previously hidden groups, and to adapt the classifier to new situations for classifying future observations.
Within the model-based framework, a classifier capable of detecting several unobserved classes in a set of unlabelled observations was proposed in Bouveyron, 2014: we briefly describe its main features here, as an extension for joint anomaly and novelty detection is proposed in Chapter 3. The Adaptive Mixture Discriminant Analysis (AMDA) is a model-based framework for supervised classification that accounts for the case where some of the test units might belong to a group not encountered in the training set. That is, the AMDA classifier is able to adapt for the detection of previously unseen classes in the unlabeled sample.
More formally, following and adapting the notation introduced in Section 1.2, let $\{(\mathbf{y}_1, \mathbf{z}_1), \ldots, (\mathbf{y}_M, \mathbf{z}_M)\}$ be the set of the unlabeled observations $\mathbf{y}_m$ with unknown classes $\mathbf{z}_m$, where $z_{mg} = 1$ if observation m belongs to group g and 0 otherwise, $g = 1, \ldots, E$, with $E \geq G$. Note that, contrarily to what previously assumed, now E classes, with E possibly bigger than G, are supposed to be present in the test set. That is, there might be a number H of "hidden" classes not previously encountered in the training set, such that $E = G + H$, with $H \geq 0$. Under this new framework, the *observed log-likelihood* for the semi-supervised classification defined in (1.14) modifies to:

$$\begin{aligned}\ell_O(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{X}, \mathbf{Y}, \mathbf{l}) = \sum_{n=1}^{N} \sum_{g=1}^{G} l_{ng} \log \left[\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right] + \\ + \sum_{m=1}^{M} \log \left[\sum_{g=1}^{G} \tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) + \sum_{h=G+1}^{E} \tau_h \phi(\mathbf{y}_m; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h) \right].\end{aligned} \tag{1.27}$$

In (1.27), the difference compared to the likelihood in (1.14) is made explicit: here the summation within the second logarithm goes from 1 to E. Clearly, when $E = G$, the likelihood in (1.27) equals the one in (1.14) since no extra classes are present in the test set, and the problem simplifies to semi-supervised model-based classification. Notice further that the first term in (1.27) accounts for the joint distribution of $(\mathbf{x}_n, \mathbf{l}_n)$, since both are observed; whereas in the second term only the marginal density of $\mathbf{y}_m$ contributes to the likelihood, given that its associated label $\mathbf{z}_m$ is unknown. Two alternative EM-based approaches for maximizing (1.27) with respect to $\tau_g, \boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, E$ are proposed in Bouveyron, 2014. The adapted classifier assigns a new observation $\mathbf{y}_m$ to a known or previously unseen class with the associated highest posterior probability:

$$\hat{z}_{mg} = \mathbb{P}(\mathcal{C} = g | \mathcal{X} = \mathbf{y}_m) = \frac{\hat{\tau}_g f(\mathbf{y}_m; \hat{\boldsymbol{\theta}}_g)}{\sum_{j=1}^{E} \hat{\tau}_j f(\mathbf{y}_m; \hat{\boldsymbol{\theta}}_j)}, \text{ for } g = 1, \ldots, G, G+1, \ldots, E.$$

Notice that the total number E of groups is not established in advance and needs to be estimated: classical tools for model selection in the mixture model framework serve to this purpose (Akaike, 1974; Schwarz, 1978).
Other connected methods include the initial approach of Miller and Browning, 2003, where a mixture model with observed and unobserved classes is considered for modeling jointly labelled and unlabelled observations. A conceptually related inferential procedure, named *transductive approach*, is developed in Section 3.2.2. More recently, Vatanen et al., 2012 proposed a semi-supervised classifier based on a mixture of Gaussians for detecting anomalies among a background of normal data, a situation that regularly arises in experimental high energy

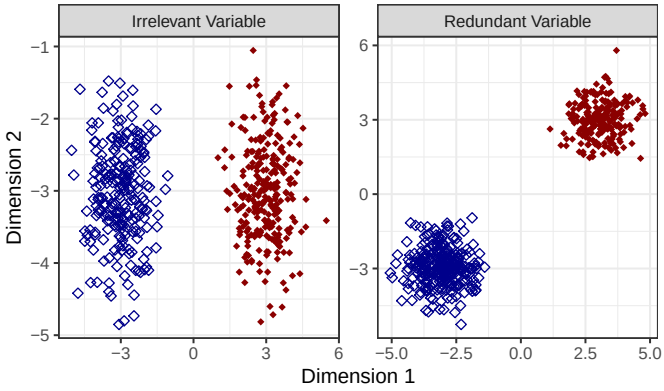

FIGURE 1.6: Examples of learning scenarios for which the second dimension is irrelevant (left panel) or redundant (right panel) in discriminating the two groups.

physics. An exhaustive bibliographic reference of model-based methods for anomaly detection is given in the "Novelty Detection" part of Section 5.5 in Bouveyron et al., 2019.

1.4 High dimensional data and variable selection

Nowadays, in many scientific domains such as chemometrics, computer vision, engineering and genetics among others, it is increasingly common to measure hundreds or thousands of variables on each sample. In principle, depending on the problem at hand, all the available features might be relevant and thus deemed to be included in a subsequent analysis. Most often, however, incorporating every piece of information at our disposal unnecessarily increases model complexity and, ultimately, it may undermine the entire output of a statistical procedure. Model-based methods,as the ones described in the previous sections, are particularly sensitive to the well-known *curse of dimensionality* (Bellman, 1957), as such models are overparametrized and suffer from identifiability problems in high dimensional spaces (Bouveyron and Brunet-Saumard, 2014). Therefore, in a discriminant analysis context, selecting the useful variables that better unveil the group structure is crucial to learn an efficient classifier. This has been known for a long time, as demonstrated by the specific literature reviews on the topic in the fields of machine learning (Blum and Langley, 1997; Yu and Liu, 2004; Liu and Motoda, 2007), data mining (Dash and Liu, 1997; Kohavi and John, 1997), bioinformatics (Saeys et al., 2007), genomic (Yu, 2008) and statistics (McLachlan, 1992; Guyon et al., 2007). Hereafter, we review the main theory of feature selection in discriminant analysis, as it is the groundwork needed for the treatment of the robust variable selection methods developed in Chapter 4.

1.4.1 Variables role in discriminant analysis

The detection of p relevant features (out of the whole collection of $P >> p$ available variables) on which to train the classifier is particularly desirable as (McLachlan, 1992):

- it simplifies parameters estimation and interpretation,
- it avoids loss on predictive power due to the inclusion of irrelevant and redundant information,
- it leads to cost reduction on future data collection and processing.

Therefore, with the aim of choosing the best predictors, it is crucial to define the concept of "relevant variable". The framework of model-based discriminant analysis allows to define "relevance" in terms of probabilistic dependence (or independence) with respect to the class membership (Ritter, 2014). The distribution of the *relevant variables*, i.e., features that bring significant information on class separation, directly depends on the class membership itself. In discriminating men and women of the same ethnicity for example, the height is naturally relevant. *Irrelevant or noisy variables*, on the contrary, do not contain any discriminating power, and hence their distribution is completely independent from the group structure. To continue with our previous example, hair and eye color do not convey any information on the gender of a person. Lastly, *redundant variables* essentially contain discriminant information that is already provided by the relevant ones: their distribution is conditionally independent of the grouping variable, given the relevant ones. If the height of a person is known, little extra information is gained by finding out his/her chest perimeter for determining his/her gender. In Figure 1.6, the first dimension is a relevant variable for discriminating the two groups, while the second dimension is respectively irrelevant in the left panel and redundant in the right one.

1.4.2 Methods for variable selection

Depending on how the variable selection process interacts with the model estimation, two general approaches for feature identification can be defined. Following the nomenclature introduced by John et al., 1994, *filter methods* are those in which the selection acts as a pre (or post) processing step, discarding variables whose distribution appears non-informative. Since the selection via filter methods is performed separately from the model estimation, i.e., without reference to the class membership, such techniques may miss important grouping information; a standard example being Principal Component Analysis (Chang, 1983). For a state-of-the-art benchmark study on the comparison of filter methods for feature selection in high dimensional classification, the reader is referred to Bommert et al., 2019.
For the second class of methods, the feature identification is "wrapped" around the classification procedure; hence they are denoted as *wrapper approaches*. Within this framework, variable selection and model estimation are simultaneously performed, aiming at identifying the predictors that better describe the underlying data partition. Focusing on the model-based methods for classification, Murphy et al., 2010 provide a wrapper approach for feature selection in semi-supervised discriminant analysis, recasting the feature identification as a model selection problem. The authors developed a greedy search and a headlong search algorithm for finding a local optimum in the model space, inspired by the seminal work on variable selection in model-based clustering of Dean et al., 2006, wherein for the first time the potential correlation between relevant and irrelevant variables was taken into account. Similarly, a general methodology for selecting predictors in model-based discriminant analysis was introduced in Maugis et al., 2011, where also theoretical results on model identifiability and consistency of the proposed criterion were validated. More recently, a regularization approach for feature selection in model-based clustering and classification was introduced in Celeux et al., 2019, where a lasso-like procedure is employed for overcoming the slowness yielded by stepwise algorithms when dealing with high-dimensional problems. The `SelvarMix R` package provides an efficient `C++` implementation of the afore-mentioned procedure.
Lastly, methods that lie in between the two approaches have also been developed in the literature. Such hybrid methods usually involve feature selection based on some measure of separability between groups, like the one introduced by Indahl and Næs, 2004, specifically tailored for spectroscopic data, and the one proposed by Andrews and McNicholas, 2014. Further, a series of techniques based on metaheuristic strategies for variable selection in discriminant

analysis can be found in Pacheco et al., 2006, while the method of Chiang and Pell, 2004 relies on a stochastic search based on genetic algorithms. In general, even though being more complex and computationally intensive, wrapper approaches provide better classification results and more accurate representation of the data generating process (Kohavi and John, 1997). For this reason, the present manuscript will focus on wrapper approaches: the novel methods introduced in Chapter 4 fall within this category.

1.5 Outline and main contributions

The present Chapter has reviewed the main methodological notions and concepts that will be adopted and developed in the remaining part of the monograph. As previously mentioned, in this thesis we investigate and propose novel solutions for dealing with adulteration in model-based classification. That is, when the learning framework is affected by real-data complexities, being them outliers, uncertain labels, unobserved classes and irrelevant variables. Particularly, Chapter 2 deals with scenarios in which the learning data are affected by the joint harmful effect of label noise and outliers. In Chapter 3 we introduce a novel methodology for anomaly and novelty detection, that extends the method developed in Chapter 2 accounting for the possible presence of hidden classes in the test set not previously encountered in the training set. Chapter 4 tackles the problem of robust high-dimensional supervised learning, where not only observations are possibly affected by attribute and class noise but also a subset of the recorded features is irrelevant to the classification task.
In details, Chapter 2 proposes a robust modification to a family of semi-supervised patterned models, for performing classification in presence of both class and attribute noise. We show that our methodology effectively addresses the issues generated by these two noise types, by identifying wrongly labeled units (noise in the response variable) and corrupted attributes in units (noise in the explanatory variables). Robust parameter estimates can therefore be obtained by excluding the noisy observations from the estimation procedure, both in the training set, and in the test set. Our proposal is based on incorporating impartial trimming and eigenvalue-ratio constraints in previous semi-supervised methods. We adapt the trimming procedure to the two different frameworks, i.e., for the labeled units and the unlabeled ones. After completing the robust estimation process, trimmed observations can be classified as well, by the usual Bayes rule. This final step allows the researcher to detect whether one observation is indeed extreme in terms of its attributes or it has been wrongly assigned to a different class. Such feature seems particularly desirable in food authenticity applications, where, due to imprecise readings and fraudulent units, it is likely to have label noise also within the labeled set. Some simulations, and a study on real data from pure and adulterated Honey samples, show the effectiveness of our proposal.
Chapter 3 proposes a model-based discriminant analysis method for anomaly and novelty detection. We show that the methodology effectively performs classification in presence of label noise, outliers and unobserved classes in the test set. By incorporating impartial trimming and eigenvalue-ratio constraints, our proposal robustly estimates model parameters of known and hidden classes, identifying as a by-product wrongly labeled and/or adulterated observations. Considering a parsimonious family of patterned models, two flexible EM-based approaches are proposed for parameter estimation: one based on the union of training and test sets, and the other made of two phases, performing sequential inference for known and hidden groups. Furthermore, we let the latter approach exploit the partial order structure of the parsimonious models, deriving fast and closed-form solutions for estimating the parameters of the extra classes. The resulting methodology includes several model-based classification methods as special cases. A robust data-driven criterion is adapted for selecting the number of unobserved

groups and constraint strength in covariances estimation. An extensive simulation study and applications on a grapevine microbiome dataset prove the effectiveness of our proposal. Particularly, the classifier capability in discriminating (known and previously unobserved) grape provenances, within an adulterated context, may foster promising developments in the food authenticity domain.
Chapter 4 introduces two wrapper variable selection methods, resistant to outliers and label noise. We show that by means of these approaches we can effectively perform high-dimensional discrimination in an adulterated scenario. The first wrapper method embeds a robust model-based classifier within a greedy-forward algorithm, validating stepwise inclusion and exclusion of variables from the relevant subset via a robust information criterion. Theoretical justification that corroborates the procedure is also discussed. The second wrapper method resorts to the theory of maximum likelihood and irrelevance, defining an objective function in which the subset of relevant variables is regarded as a parameter to be estimated. A dedicated algorithm for Maximum Likelihood Estimation within a Gaussian family of patterned models is developed, and practical implementation issues are considered. Further, pros and cons of the two novel procedures are discussed. A simulation study is developed for assessing the effectiveness of our proposals in recovering the true discriminative features in a contaminated scenario, comparing their performances against well-known variable selection criteria. The novel methods are then successfully applied in solving a high-dimensional classification problem of contaminated spectroscopic data. High discriminating power is exhibited by the final models, whence the identification of the wrongly labeled and/or adulterated observations is derived as a by-product of the estimation procedures.
Chapter 5 concludes the manuscript, summarizing the main contributions and emphasizing future research developments.
The last part of the thesis includes a somewhat less related topic: the employment of robust mixture of factor analyzers in a clustering context for detecting wine adulteration. Given that this piece of work has been produced during the early stage of the PhD program, we decided to include it in Appendix A. The remaining appendices supplement the main chapters with additional material related to computational and theoretical aspects. Appendix B provides the listings for the main `R` routines developed in the thesis, while Appendix C reports some details related to the computing time required by such novel methods. Lastly, in Appendix D we present a discussion and some initial attempts to justify the usage of trimmed information criteria in the context of robust model selection.

Chapter 2

Robust model-based classification for attribute and class noise

Based on:

Cappozzo, A., Greselin, F., Murphy, T. B.
"A robust approach to model-based classification based on trimming and constraints"
Advances in Data Analysis and Classification (2019)
https://doi.org/10.1007/s11634-019-00371-w

2.1 Introduction

In statistical learning, we define classification as the task of assigning group memberships to a set of unlabelled observations. Whenever a labelled sample (i.e., the training set) is available, the information contained in such dataset is exploited to classify the remaining unlabelled observations (i.e., the test set), either in a supervised or in a semi-supervised manner, depending whether the information contained in the test set is included in building the classifier (e.g. McNicholas, 2016). Either way, the presence of unreliable data points can be detrimental for the classification process, especially if the training size is small (Zhu and Wu, 2004).
Broadly speaking, noise is anything that obscures the relationship between the attributes and the class membership (Hickey, 1996). In a classification context, Wu, 1995 distinguishes between two types of noise: attribute noise and class noise. The former is related to contamination in the exploratory variables, that is when observations present unusual values on their predictors; whereas the latter refers to samples whose associated labels are wrong.
The approach presented in this chapter is based on a robust estimation of a Gaussian mixture model with parsimonious structure, to account for both attribute and label noise. Our conjecture is that the contaminated observations would be the least plausible units under the robustly estimated model: the corrupted subsample will be revealed by detecting those observations with the lowest contributions to the associated likelihood. Impartial trimming (Gordaliza, 1991b; Gordaliza, 1991a; Cuesta-Albertos et al., 1997) is employed for robustifying the parameter estimates, being a well established technique to treat mild and gross outliers in the clustering literature (García-Escudero et al., 2010b) and here used, for the first time, to additionally account for label noise in a classification framework. A semi-supervised approach is developed, where information contained in both labelled and unlabelled samples is combined for improving the classifier performance and for defining a data-driven method to identify outlying observations possibly present in the test set.

The rest of the Chapter is organized as follows. Section 2.2 introduces the robust updating classification rules, covering the model formulation, inference aspects and model selection. Simulation studies to compare the method introduced in Section 2.2 with other popular classification methods are reported in Section 2.3. Finally, in Section 2.4 our proposal is employed in performing classification and adulteration detection in a food authenticity context, dealing with contaminated samples of Irish honey. Concluding notes and further research directions are outlined in Section 2.5. The proof of Proposition 1, details on the parameter values for simulation study II and efficient algorithms for enforcing the eigen-ratio constraint for different patterned models are deferred respectively to appendices A, B and C (Sections 2.6, 2.7 and 2.8).

2.2 Robust Updating Classification Rules

We introduce here a robust modification to the updating classification rules described in Section 1.2.2, with the final aim of developing a classifier whose performance is not affected by contaminated data, either in the form of label noise and outlying observations.

2.2.1 Model formulation

The main idea of the proposed approach is to employ techniques originated in the branch of robust statistics to obtain a model-based classifier in which parameters are robustly estimated and outlying observations identified. We are interested in providing a method that jointly accounts for noise on response and exploratory variables, where the former might be present in the labelled set and the latter in both the labelled and unlabelled sets. We propose to modify the log-likelihood in (1.14) with a *trimmed mixture log-likelihood* (Neykov et al., 2007) and to employ impartial trimming and constraints on the covariance matrices for achieving both robust parameter estimation and identification of the unreliable sub-sample. Impartial trimming is enforced by considering the distinct structure of the likelihoods associated to the labelled and unlabelled sets, accounting for the possible label noise that might be present in the labelled sample (see Section 2.2.2 for details). Following the same notation introduced in Section 1.2.1, we aim at maximizing the *trimmed observed data log-likelihood*:

$$\begin{aligned}\ell_{trim}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{X}, \mathbf{Y}, \mathbf{l}) = \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \sum_{g=1}^{G} \mathbf{l}_{ng} \log \left[\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right] + \\ + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \log \left[\sum_{g=1}^{G} \tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right]\end{aligned} \tag{2.1}$$

where $\zeta(\cdot)$, $\varphi(\cdot)$ are 0-1 trimming indicator functions, that express whether observation $\mathbf{x}_n$ and $\mathbf{y}_m$ are trimmed off or not. A fixed fraction α_l and α_u of observations, belonging to the labelled and unlabelled set respectively, is unassigned by setting $\sum_{n=1}^{N} \zeta(\mathbf{x}_n) = \lceil N(1-\alpha_l) \rceil$ and $\sum_{m=1}^{M} \varphi(\mathbf{y}_m) = \lceil M(1-\alpha_u) \rceil$. In particular, we will see in Section 2.2.2 that the less plausible samples under the currently estimated model are tentatively trimmed out at each step of the iterations that leads to the final estimate. The *labelled trimming level* α_l and the *unlabelled trimming level* α_u account for possible adulteration in both sets. At the end of the iterations, a value of $\zeta(\mathbf{x}_n) = 0$ or $\varphi(\mathbf{y}_m) = 0$ corresponds to identify $\mathbf{x}_n$ or $\mathbf{y}_m$, respectively, as unreliable observations. Notice that impartial trimming automatically deals with both class noise and attribute noise, as observations that suffer from either noise structure will give low contribution to the associated likelihood.

Maximization of (2.1) is carried out via the EM algorithm, in which an appropriate Concentration Step (Rousseeuw and Driessen, 1999) is performed in both labelled and unlabelled sets at

TABLE 2.1: Nomenclature, covariance structure and number of free parameters in $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_G$: γ denotes the number of parameters related to the orthogonal rotation and δ the number of parameters related to the eigenvalues. The last column indicates whether the eigenvalue-ratio (ER) constraint is required.

Model	$\boldsymbol{\Sigma}_g$	γ	δ	ER
EII	$\lambda \boldsymbol{I}$	-	1	Not required
VII	$\lambda_g \boldsymbol{I}$	-	G	Required
EEI	$\lambda \boldsymbol{A}$	-	p	Not required
VEI	$\lambda_g \boldsymbol{A}$	-	$G+p-1$	Required
EVI	$\lambda \boldsymbol{A}_g$	-	$Gp-(G-1)$	Required
VVI	$\lambda_g \boldsymbol{A}_g$	-	Gp	Required
EEE	$\lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}'$	$p(p-1)/2$	p	Not required
VEE	$\lambda_g \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}'$	$p(p-1)/2$	$G+p-1$	Required
EVE	$\lambda \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}'$	$p(p-1)/2$	$Gp-(G-1)$	Required
EEV	$\lambda \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g'$	$Gp(p-1)/2$	p	Not required
VVE	$\lambda_g \boldsymbol{D}\boldsymbol{A}_g\boldsymbol{D}'$	$p(p-1)/2$	Gp	Required
VEV	$\lambda_g \boldsymbol{D}_g\boldsymbol{A}\boldsymbol{D}_g'$	$Gp(p-1)/2$	$G+p-1$	Required
EVV	$\lambda \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g'$	$Gp(p-1)/2$	$Gp-(G-1)$	Required
VVV	$\lambda_g \boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g'$	$Gp(p-1)/2$	Gp	Required

each iteration to enforce the impartial trimming. In addition, we protect the parameter estimation from spurious solutions, that may arise whenever one component of the mixture fits a random pattern in the data, considering the eigenvalues-ratio restriction introduced in Section 1.3.2. Notice that, when the family of patterned models detailed in Section 1.2.1 is considered, the constraint in (1.20) is still needed whenever either the shape or the volume is free to vary across components (García-Escudero et al., 2018b). That is for all models in Table 2.1 that present "Required" entry in the ER column. The considered approach is the (semi)-supervised version of the methodology proposed in Dotto and Farcomeni, 2019, which is framed in a completely unsupervised scenario. Feasible and computationally efficient algorithms for enforcing the eigen-ratio constraint for different patterned models are reported in the Appendix C (Section 2.8).

2.2.2 Estimation procedure

The EM algorithm for obtaining Maximum Trimmed Likelihood Estimates of the robust updating classification rules involves the following steps:

- *Robust Initialization:* Set $k = 0$. Employing only the labelled data, we obtain robust starting values for the mean vector $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$ of the multivariate normal density for each group g, $g = 1, \ldots, G$, by means of the following procedure:

  1. For each class g, draw a random $(p+1)$-subset J_g and compute its empirical mean $\hat{\boldsymbol{\mu}}_g^{(0)}$ and variance covariance matrix $\hat{\boldsymbol{\Sigma}}_g^{(0)}$ according to the considered parsimonious structure. This procedure yields better initial subsets than drawing random $\lceil N(1-\alpha_l)\rceil$-subsets directly, because the probability of drawing an outlier-free $(p+1)$-subset is much higher than that of drawing an outlier-free $\lceil N(1-\alpha_l)\rceil$-subset (Hubert et al., 2018).

2. Set

$$\hat{\boldsymbol{\theta}} = \{\hat{\tau}_1, \ldots, \hat{\tau}_G, \hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_G, \hat{\boldsymbol{\Sigma}}_1, \ldots, \hat{\boldsymbol{\Sigma}}_G\} = \\ = \{\hat{\tau}_1^{(0)}, \ldots, \hat{\tau}_G^{(0)}, \hat{\boldsymbol{\mu}}_1^{(0)}, \ldots, \hat{\boldsymbol{\mu}}_G^{(0)}, \hat{\boldsymbol{\Sigma}}_1^{(0)}, \ldots, \hat{\boldsymbol{\Sigma}}_G^{(0)}\}$$

where $\hat{\tau}_1^{(0)} = \ldots = \hat{\tau}_G^{(0)} = 1/G$.

3. For each $\mathbf{x}_n$, $n = 1, \ldots, N$, compute the conditional density

$$f(\mathbf{x}_n | l_{ng} = 1; \hat{\boldsymbol{\theta}}) = \phi\left(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g\right) \quad g = 1, \ldots, G. \tag{2.2}$$

$\lfloor N\alpha_l \rfloor$% of the samples with lowest value of (2.2) are temporarily discarded as possible outliers, namely label noise and/or attribute noise. That is, $\zeta(\mathbf{x}_n) = 0$ for such observations.

4. The parameter estimates are updated, based on the non-discarded observations:

$$\hat{\tau}_g = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng}}{\lceil N(1-\alpha_l) \rceil} \quad g = 1, \ldots, G \tag{2.3}$$

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} \mathbf{x}_n}{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng}} \quad g = 1, \ldots, G. \tag{2.4}$$

Estimation of $\boldsymbol{\Sigma}_g$ depends on the considered patterned model, details are given in Bensmail and Celeux, 1996.

5. Iterate $3-4$ until the $\lfloor N\alpha_l \rfloor$ discarded observations are exactly the same on two consecutive iterations, then stop (usually, ≤ 3 iterations are required).

The procedure described in steps $1-5$ is performed `nsamp` times, and the parameter estimates $\hat{\boldsymbol{\theta}}^R$ that lead to the highest value of the objective function $\ell_{trim}(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}} | \mathbf{X}, 1) = \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \sum_{g=1}^{G} l_{ng} \log\left[\hat{\tau}_g \phi(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)\right]$, out of `nsamp` repetitions, are retained. The aforedescribed procedure stems from the ideas of the FastMCD algorithm of Rousseeuw and Driessen, 1999, here adapted for dealing with parsimonious structures in the covariance matrices. Retaining $\hat{\boldsymbol{\theta}}^R$ as final estimate leads to a fully supervised robust model-based method, called REDDA hereafter (see Section 2.3.1). Then, if the selected patterned model allows for heteroscedastic $\boldsymbol{\Sigma}_g$ and (1.20) is not satisfied, constrained maximization is enforced, see Appendix C (Section 2.8) for details.

- *EM Iterations:* denote by $\hat{\boldsymbol{\theta}}^{(k)} = \{\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_G^{(k)}, \hat{\boldsymbol{\mu}}_1^{(k)}, \ldots, \hat{\boldsymbol{\mu}}_G^{(k)}, \hat{\boldsymbol{\Sigma}}_1^{(k)}, \ldots, \hat{\boldsymbol{\Sigma}}_G^{(k)}\}$ the parameter estimates at the k-th iteration of the algorithm.

  - *Step 1 - Concentration*: The trimming procedure is implemented by discarding the $\lfloor N\alpha_l \rfloor$ observations $\mathbf{x}_n$ with smaller values of

$$D\left(\mathbf{x}_n; \hat{\boldsymbol{\theta}}^{(k)}\right) = \prod_{g=1}^{G} \left[\phi\left(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right)\right]^{l_{ng}} \quad n = 1, \ldots, N \tag{2.5}$$

and discarding the $\lfloor M\alpha_u \rfloor$ observations $\mathbf{y}_m$ with smaller values of

$$D\left(\mathbf{y}_m; \hat{\boldsymbol{\theta}}^{(k)}\right) = \sum_{g=1}^{G} \hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right) \quad m = 1, \ldots, M. \tag{2.6}$$

– *Step 2 - Expectation*: For each non-trimmed observation $\mathbf{y}_m$ compute the posterior probabilities

$$\hat{z}_{mg}^{(k+1)} = \frac{\hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right)}{D\left(\mathbf{y}_m; \hat{\boldsymbol{\theta}}^{(k)}\right)} \quad g = 1, \ldots, G; \quad m = 1, \ldots, M. \tag{2.7}$$

– *Step 3 - Constrained Maximization*: The parameter estimates are updated, based on the non-discarded observations and the current estimates for the unknown labels:

$$\hat{\tau}_g^{(k+1)} = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \hat{z}_{mg}^{(k+1)}}{\lceil N(1-\alpha_l) \rceil + \lceil M(1-\alpha_u) \rceil} \quad g = 1, \ldots, G \tag{2.8}$$

$$\hat{\boldsymbol{\mu}}_g^{(k+1)} = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} \mathbf{x}_n + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \hat{z}_{mg}^{(k+1)} \mathbf{y}_m}{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \hat{z}_{mg}^{(k+1)}} \quad g = 1, \ldots, G. \tag{2.9}$$

Estimation of $\boldsymbol{\Sigma}_g$ depends on the considered patterned model and on the eigenvalues-ratio constraint. Details are given in Bensmail and Celeux, 1996 and, if (1.20) is not satisfied, in Appendix C (Section 2.8).

– *Step 4 - Convergence of the EM algorithm*: Check for algorithm convergence (see Section 2.2.3). If convergence has not been reached, set $k = k + 1$ and repeat steps 1-4.

Notice how the trimming step differs between the labelled and unlabelled observations. We implicitly assume that a label in the training set conveys a sound meaning about the presence of a class of objects. Therefore, in the labelled set, we opted for trimming the samples with lowest conditional density $f(\mathbf{x}_n | l_{ng} = 1; \hat{\boldsymbol{\theta}}^{(k)}) = \phi\left(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right)$. The alternative choice of considering the joint density $f(\mathbf{x}_n, l_{ng}; \hat{\boldsymbol{\theta}}^{(k)}) = \prod_{g=1}^{G} \left[\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]^{l_{ng}}$ is instead prone to completely trim off groups with small prior probability τ_g for large enough value of α_l, and should be discarded. Note that with (2.5) we are both discriminating label noise (i.e., observations that are likely to belong to the mixture model but whose associated label is wrong) and outliers. In the unlabelled set, on the other hand, trimming is based on the marginal density $f(\mathbf{y}_m; \hat{\boldsymbol{\theta}}^{(k)}) = \sum_{g=1}^{G} \hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right)$, having no prior information on the group membership of the samples.
Once convergence is reached, the estimated values $\hat{z}_{mg}$ provide a classification for the unlabelled observations $\mathbf{y}_m$, assigning observation m to group g if $\hat{z}_{mg} > \hat{z}_{mg'}$ for all $g' \neq g$. Final values of $\zeta(\mathbf{x}_n) = 0$, and $\varphi(\mathbf{y}_m) = 0$, classify $\mathbf{x}_n$ and $\mathbf{y}_m$ respectively, as outlying observations.
The routines for the robust updating classification rules have been written in R language (R Core Team, 2018): the source code is available at https://github.com/AndreaCappozzo/rupclass.
The estimation procedure detailed in this Section implies the monotonicity of the algorithm, according to:

Proposition 1: If the values $\zeta(\mathbf{x}_n)$, $\varphi(\mathbf{y}_m)$, $n = 1, \ldots, N$, $m = 1, \ldots, M$ are kept fixed, the EM algorithm described in Section 2.2.2 implies $\ell_{trim}(\hat{\boldsymbol{\theta}}^{(k+1)} | \mathbf{X}, \mathbf{Y}, 1) \geq \ell_{trim}(\hat{\boldsymbol{\theta}}^{(k)} | \mathbf{X}, \mathbf{Y}, 1)$ at any k.

The proof is reported in Appendix A (Section 2.6). Furthermore, our estimation procedure reduces possible incorrect modes of the optimization function (spurious maximizers) and offers a constructive way to obtain a maximizer $\hat{\theta}_n$ for the sample problem, that converges to

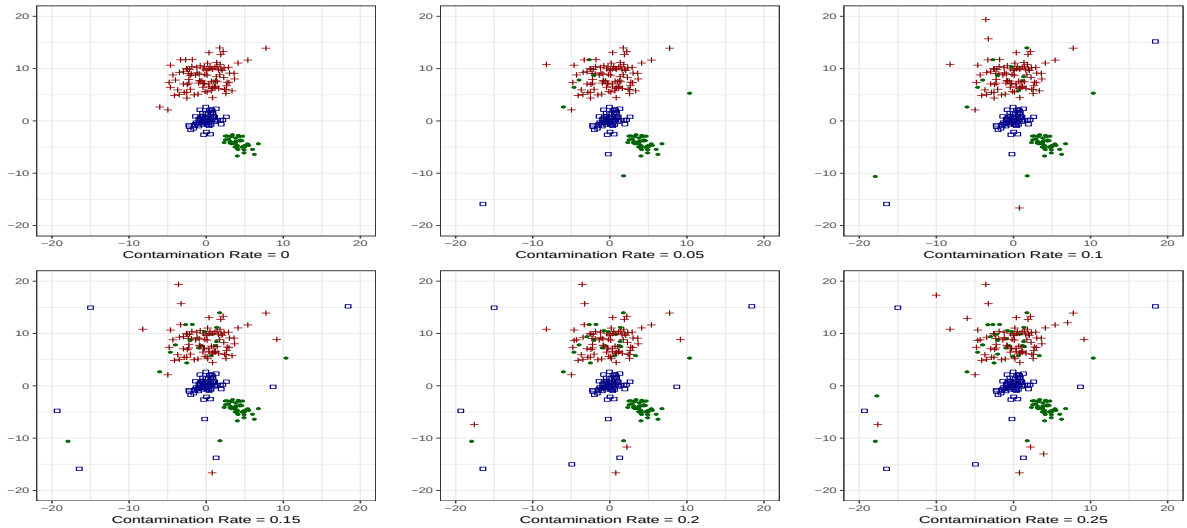

FIGURE 2.1: Simulated data considering the simulation setup described in Section 2.3.1, varying contamination rate η

the global maximizer for the population, see García-Escudero et al., 2008 and García-Escudero et al., 2015.

2.2.3 Convergence criterion

We assess whether the EM algorithm has reached convergence evaluating at each iteration how close the trimmed log-likelihood is to its estimated asymptotic value, using the Aitken acceleration (Aitken, 1926):

$$a^{(k)} = \frac{\ell_{trim}^{(k+1)} - \ell_{trim}^{(k)}}{\ell_{trim}^{(k)} - \ell_{trim}^{(k-1)}} \tag{2.10}$$

where $\ell_{trim}^{(k)}$ is the trimmed observed data log-likelihood from iteration k. The asymptotic estimate of the trimmed log-likelihood at iteration k is given by (Böhning et al., 1994):

$$\ell_{\infty_{trim}}^{(k)} = \ell_{trim}^{(k)} + \frac{1}{1 - a^{(k)}} \left(\ell_{trim}^{(k+1)} - \ell_{trim}^{(k)} \right). \tag{2.11}$$

The EM algorithm is considered to have converged when $|\ell_{\infty_{trim}}^{(k)} - \ell_{trim}^{(k)}| < \varepsilon$; a value of $\varepsilon = 10^{-5}$ has been chosen for the experiments reported in the next Sessions.

2.2.4 Model selection

A robust likelihood-based criterion is employed for choosing the best model among the 14 patterned covariance structures listed in Table 2.1 and a reasonable value for the constraint c in (1.20):

$$TBIC = 2\ell_{trim}(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) - v_{XXX}^{c} \log\left(\lceil N(1-\alpha_l) \rceil + \lceil M(1-\alpha_u) \rceil \right) \tag{2.12}$$

where $\ell_{trim}(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ denotes the maximized trimmed observed data log-likelihood and v_{XXX}^{c} a penalty term whose definition is:

$$v_{XXX}^{c} = Gp + G - 1 + \gamma + (\delta - 1)\left(1 - \frac{1}{c}\right) + 1. \tag{2.13}$$

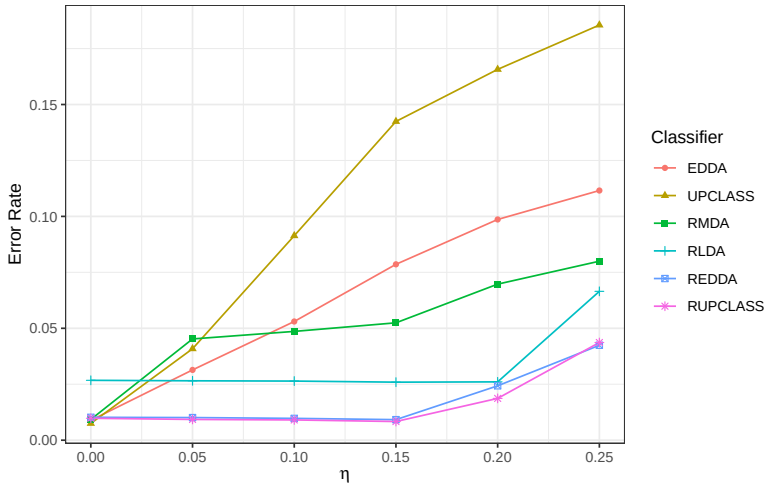

FIGURE 2.2: Average misclassification errors on $B = 1000$ runs for different classification methods, increasing contamination rate.

That is, v^c_{XXX} depends on the total number of parameters to be estimated: γ and δ for every XXX patterned model are given in Table 2.1. It also accounts for the trimming levels and for the eigen-ratio constraint c, according to Cerioli et al., 2018a. Note that, when $c \to +\infty$ and $\alpha_l = \alpha_u = 0$, (2.12) is the Bayesian Information Criterion (Schwarz, 1978). A general discussion on the rationale behind the usage of trimmed information criteria and future research directions for their theoretical development is reported in Appendix D.

2.3 Simulation studies

In this Section, we present two simulated data experiments: Simulation Study I compares the performances of several model-based classification methods in a low dimensional setting when dealing with noisy data at different contamination rates; Simulation Study II considers a higher dimensional scenario in which the accuracy performance of some popular classification methods is assessed, at a fixed contamination rate. In both scenarios we consider a joint noise structure on response and exploratory variables.

2.3.1 Simulation study I

Experimental setup

We consider a data generating process given by a mixture of $G = 3$ components of bivariate normal distributions, according to the following parameters:

$$\boldsymbol{\tau} = (0.3, 0.2, 0.5)', \quad \boldsymbol{\mu}_1 = (0,0)', \quad \boldsymbol{\mu}_2 = (4,-4)', \quad \boldsymbol{\mu}_3 = (0,8)'$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & -0.3 \\ -0.3 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 6.71 & 2.09 \\ 2.09 & 6.71 \end{bmatrix}.$$

600 observations were generated from the model, randomly assigning $N = 200$ to the labelled set and $M = 400$ to the unlabelled set. The labelled set was subsequently adulterated with contamination rate η (ranging from 0 to 0.25), wrongly assigning $\lceil \eta/2N \rceil$ of the third group units to the first class and adding $\lceil \eta/2N \rceil$ randomly labelled points generated from a Uniform

TABLE 2.2: Average misclassification errors on $B = 1000$ runs, varying method and contamination rate η. Standard errors are reported in parenthesis.

η	0	0.05	0.10	0.15	0.20	0.25
EDDA	0.009	0.031	0.053	0.079	0.099	0.112
	(0.005)	(0.026)	(0.043)	(0.051)	(0.054)	(0.05)
UPCLASS	0.008	0.041	0.091	0.142	0.166	0.186
	(0.004)	(0.056)	(0.088)	(0.088)	(0.08)	(0.067)
RMDA	0.009	0.045	0.049	0.052	0.07	0.08
	(0.005)	(0.072)	(0.063)	(0.057)	(0.068)	(0.073)
RLDA	0.027	0.027	0.026	0.026	0.026	0.067
	(0.009)	(0.009)	(0.009)	(0.008)	(0.009)	(0.037)
REDDA	0.01	0.01	0.01	0.009	0.024	0.042
	(0.005)	(0.005)	(0.005)	(0.005)	(0.014)	(0.014)
RUPCLASS	0.01	0.009	0.009	0.008	0.019	0.044
	(0.005)	(0.005)	(0.005)	(0.005)	(0.013)	(0.014)

distribution on the square with vertices $[(-20,-20),(-20,20),(20,-20),(20,20)]$. The contamination is therefore twofold, involving jointly label switching and outliers for a total of ηN adulterated labelled units. Examples of labelled datasets with different contamination rates are reported in Figure 2.1. Performances of 6 model-based classification methods are considered:

- EDDA: Eigenvalue Decomposition Discriminant Analysis (Bensmail and Celeux, 1996)
- UPCLASS: Updating Classification Rules (Dean et al., 2006)
- RMDA: Robust Mixture Discriminant Analysis (Bouveyron and Girard, 2009)
- RLDA: Robust Linear Discriminant Analysis (Hawkins and McLachlan, 1997)
- REDDA: Robust Eigenvalue Decomposition Discriminant Analysis. This is the supervised version of the model described in Section 2.2, where only the labelled observations are used for parameters estimation obtained via the robust initialization detailed in Section 2.2.2.
- RUPCLASS: Robust Updating Classification Rules. The semi-supervised method described in Section 2.2.

To make a fair performance comparison, a level of $\alpha_l = 0.15$ (REDDA and RUPCLASS) and $\alpha_u = 0.05$ (RUPCLASS) have been kept fixed throughout the simulation study. Nevertheless, exploratory tools such as Density-Based Silhouette plot (Menardi, 2011) and heuristic procedures as the ones introduced in García-Escudero et al., 2011 and García-Escudero et al., 2018c could be employed to validate and assess the choice of α_l and α_u. In particular, notice that the quantities $D\left(\mathbf{x}_n;\hat{\boldsymbol{\theta}}\right)$ and $D\left(\mathbf{y}_m;\hat{\boldsymbol{\theta}}\right)$ measure the evidence of the belonging of $\mathbf{x}_n$ and $\mathbf{y}_m$ to the training and test model terms, respectively. We can therefore make use of their ordered distribution to provide reasonable values for the labeled and unlabeled trimming levels, along the lines of García-Escudero et al., 2018c. An explicative example of how this heuristic works for the training set is provided in Figure 2.4, considering a realization of the afore-described data generating process with contamination level η equal to 0.15. In the left column of Figure 2.4 the points $\{\frac{n}{N}, D_{(n)}(x_n;\hat{\theta})\}$, where $D_{(n)}\left(\mathbf{x}_n;\hat{\boldsymbol{\theta}}\right)$ identifies the non-decreasing sequence of $D\left(\mathbf{x}_n;\hat{\boldsymbol{\theta}}\right)$, have been plotted for REDDA models estimated with increasing labeled trimming

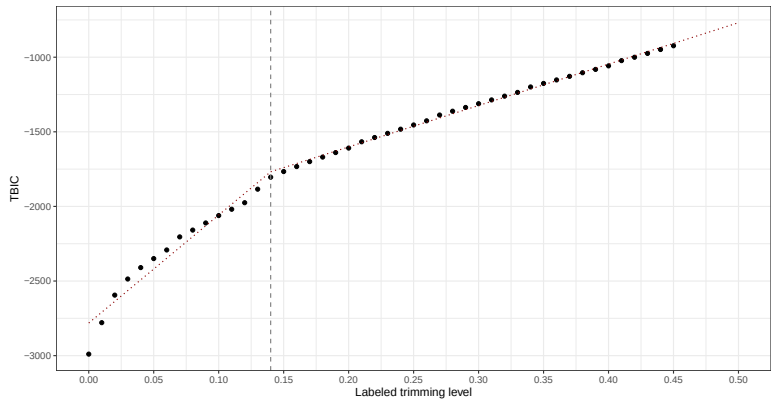

FIGURE 2.3: Labeled trimming levels α_l against TBIC values, with reference to REDDA models. The dotted red lines represent the segmented threshold model in (2.14), fitted via the `chngpt R` package. The dashed grey line highlights the estimated change point $\hat{e} = 0.14$.

levels $\alpha_l \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. Ideally, the appropriate choice of α_l is recognized by identifying the plot in which an elbow arises in the proximity of the considered trimming level, that is, a value α_l such that $D_{(n)}\left(\mathbf{x}_n; \hat{\boldsymbol{\theta}}\right)$ is steep for $n/N < \alpha_l$ and the slope of the curve decreases for $n/N > \alpha_l$. We see that, in this example, such heuristic favors the correct solution $\alpha_l = 0.15$, highlighted also by the associated partition depicted in the right column of Figure 2.4: both outliers and label noise originally present in the learning set are identified and correctly disposed of via impartial trimming. Alternatively, as already performed in Neykov et al., 2007 in the context of robust mixtures, a two-phase regression of the TBIC values against the trimming percentages may be employed to detect a change-point in the resulting pattern. In details, the considered regression equation reads:

$$TBIC = \beta_0 + \beta_1(\alpha_l - e)_+ + \gamma\alpha_l \tag{2.14}$$

where e is the threshold parameter and $(\alpha_l - e)_+$ denotes the hinge function, which equals $\alpha_l - e$ when $\alpha_l > e$ and 0 otherwise. The segmented threshold model in (2.14) is fitted via the `chngpt R` package (Fong et al., 2017), returning estimates $\hat{\beta}_0 = -2781.467$, $\hat{\beta}_1 = -4464.528$ and $\hat{\gamma} = 7239.799$. The change point is estimated to be at $\hat{e} = 0.14$ (see Figure 2.3), adjacent to the true contamination level η equal to 0.15. With respect to the standard mixture framework, a change-point detection approach could even be more appropriate in our classification context, as the true number of classes is known a priori and need not be sought, simplifying the tuning of the trimming levels. The very same rationales may be similarly applied to the test units for setting a reasonable value for the unlabeled trimming parameter α_u. Furthermore, one could also monitor, varying α_u, the stability of the group partition in the test set to identify possible changes due to the unexpected inclusion of outliers in the estimation procedure, along the lines of Cerioli et al., 2018a. A more automatic approach based on an iterative re-weighting procedure to estimate the percentage of contamination, like the one introduced in Dotto et al., 2018, could also be adapted to our framework. This, however, goes beyond the scope of the present manuscript, it will nonetheless be addressed in the future. A value of $c = 20$ was selected for the eigenvalue-ratio restriction in (1.20). Simulation study results are presented in the following subsections.

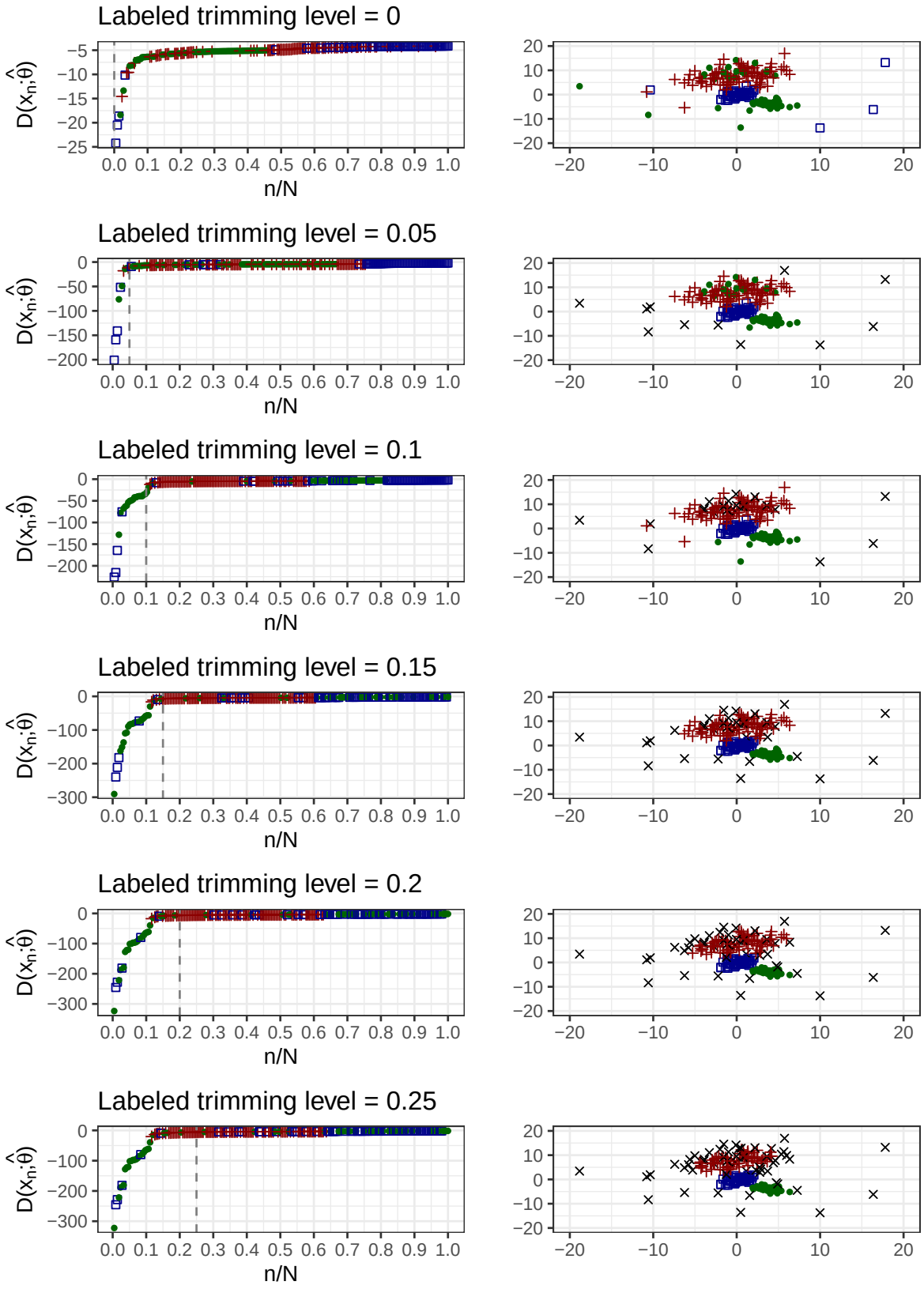

FIGURE 2.4: Considering the $D\left(\mathbf{x}_n;\hat{\boldsymbol{\theta}}\right)$ in non-decreasing order, giving raise to the ordered sequence $D_{(n)}\left(\mathbf{x}_n;\hat{\boldsymbol{\theta}}\right)$, points $\{\frac{n}{N}, D_{(n)}(x_n;\hat{\theta})\}$ are plotted (left column); the resulting trimming assignment (right column) with reference to the REDDA model, varying labeled trimming level α_l. Employed trimming levels are highlighted by vertical lines (left column), trimmed observations are denoted by "$\times$" (right column).

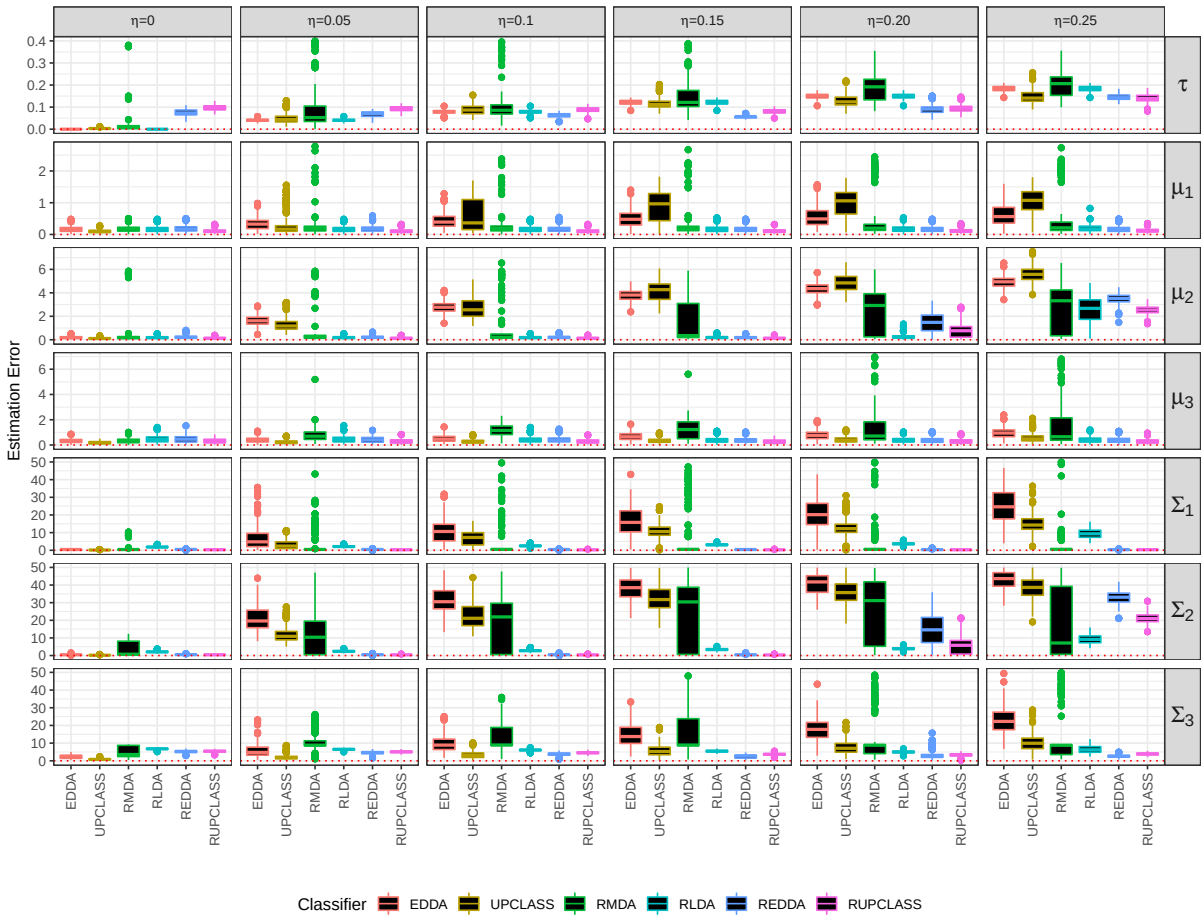

FIGURE 2.5: Box plots of the simulated estimation errors for the parameters of the mixture, computed via Euclidean norms for the proportion vector $\boldsymbol{\tau}$, the mean vectors $\boldsymbol{\mu}_g$ and covariance matrices $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, 3$ for the different models, varying contamination rate η from 0 to 0.25.

Classification performance

Average misclassification errors for the different methods and for varying contamination rates are reported in Table 2.2 and in Figure 2.2. The error rate is computed on the unlabelled dataset and averaged over the $B = 1000$ simulations. As expected, the misclassification error is fairly equal to all methods when there is no contamination rate, with the only exception being RLDA: this is due to the implicit model assumption that $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3$, which is not the case in our simulated scenario. As the contamination rate increases, so does the error rate for the non-robust methods (EDDA and UPCLASS), whereas for RLDA and RMDA it has a lower increment rate. Nevertheless, such methods fail to jointly cope with both sources of adulteration, namely class and attribute noise. Our proposals REDDA and RUPCLASS, thanks to the trimming step enforced in the estimation process, have always higher correct classification rates, on average, at any adulteration level. Notice that, to compare results of robust and non-robust methods, also the trimmed observations were classified a-posteriori according to the Bayes rule, assigning them to the component g having greater value of $\hat{\tau}_g \phi(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g)$.
On average, the robust semi-supervised approach performs better than the supervised counterpart, due to the information incorporated from genuine unlabelled data in the estimation process. Interestingly, the same behavior is not reflected in the non-robust counterparts, where the detrimental effect of contaminated labelled units magnifies the bias of the UPCLASS method. Therefore, robust solutions are even more paramount when a semi-supervised approach is considered.

Parameter estimation

Figure 2.5 reports the box plots of the simulated estimation error over $B = 1000$ Monte Carlo repetitions for the parameters of the mixture model, computing Euclidean norms for the proportion vector $\boldsymbol{\tau}$, the mean vectors $\boldsymbol{\mu}_g$ and covariance matrices $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, 3$. The estimated values for the mixing proportions are mildly affected when increasing contamination is considered; conversely, the estimation of $\boldsymbol{\mu}_2$ is on average heavily influenced by the adulterating process, and also the robust methods fail to estimate it correctly as soon as the contamination rate η is larger than the trimming level $\alpha_l = 0.15$. Clearly, the estimation of the covariance matrices is as well badly affected in most extreme scenarios, where their entries are inflated in order to accommodate more and more bad points. Our robust proposals are less affected by the harmful effect of adding anomalous observations, also in the most adulterated scenario.

2.3.2 Simulation study II

Experimental Setup

We consider here a simulating model with a larger number of features ($p = 10$), where the data generating process is given by a mixture of $G = 4$ components of a multivariate t-distribution with $\nu = 6$ degrees of freedom. More details on the parameter values are contained in Appendix B (Section 2.7). 1000 observations were generated from the model, randomly assigning $N = 250$ to the labelled set and $M = 750$ to the unlabelled set. The training set was subsequently adulterated wrongly labelling 10 units and adding 15 randomly labelled outlying points, uniformly generated in the p-dimensional hypercube over $[10, 15]^{10}$. We therefore consider a scenario in which 10% of the learning units are contaminated, via both label and attribute noise.
Together with the model-based methods previously described in Section 2.3.1, we included in the performance evaluation widely used classification techniques that, even though not engineered to achieve robustness, are noise tolerant. Particularly, the ensemble learner AdaBoost

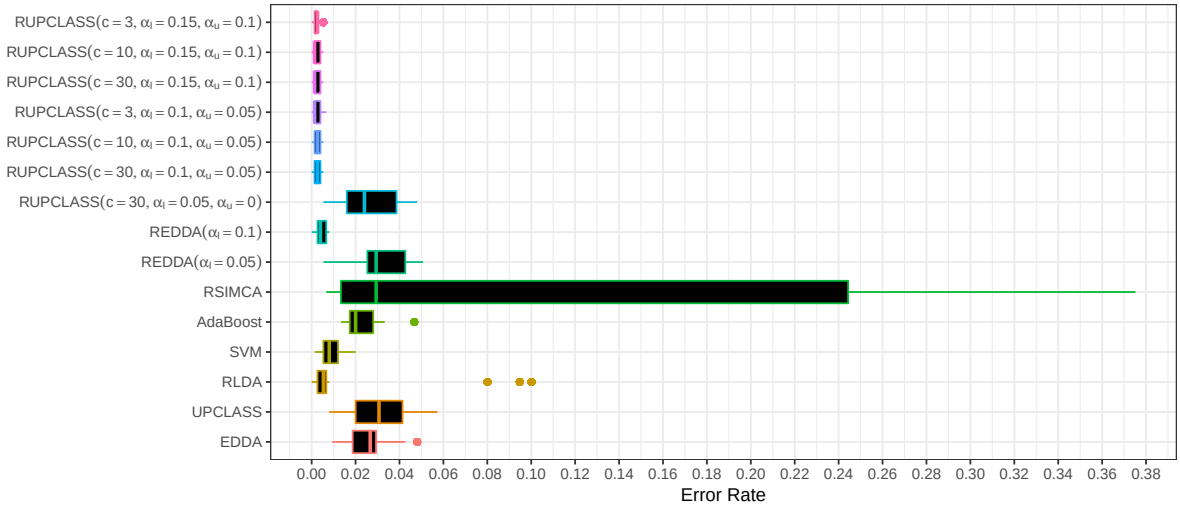

FIGURE 2.6: Box plots of the misclassification errors under $B = 1000$ repetitions of the simulating experiment II. Error rate is computed on the $M = 750$ data points of the test set for different classification methods.

(Freund and Schapire, 1997) and the kernel method Support Vector Machine (Cortes and Vapnik, 1995) were added to the comparison. Furthermore, the robust adaptation of the SIMCA method for high-dimensional classification (Vanden Branden and Hubert, 2005) was also considered. The classification performance of the afore-described techniques are tested against the proposed methodologies, under different combinations of (c, α_l, α_u): accuracy results are reported in the next Section.

Classification Performance

Boxplots of the misclassification errors for the considered methods are reported in Figure 2.6. The error rate is computed on the $M = 750$ units of the test set, under $B = 1000$ repetitions of the generating process and subsequent adulteration scheme described in Section 2.3.2. As it was already apparent from the previous simulation study, accuracy for non-robust methods is badly affected by the contamination present in the learning set. Even though not specifically designed for dealing with adulterated datasets, SVM and AdaBoost perform better than the non-robust model-based approaches, thanks to their non-parametric nature and flexibility. As expected, the best classification accuracy are obtained by the robust methodologies, namely RLDA and our proposals REDDA and RUPCLASS. We also check the sensitivity of our techniques comparing different combinations of (c, α_l, α_u). As it is easily visible in the boxplots, setting a smaller than needed labelled trimming level α_l leads to a loss in prediction accuracy, as a portion of adulterated units still affects the learning phase. Once the corrupted observations are correctly trimmed (i.e., α_l is set ≥ 0.1), accuracy seems to remain stable with little influence induced by the choice of c and α_u, with only a slight preference for the semi-supervised RUPCLASS over its supervised version REDDA. This shows that setting a higher value of α_l is less detrimental than underestimating it, and that the impartial trimming almost exactly identifies the corrupted units when $\alpha_l = 0.1$, that is the true adulteration proportion. The bad performance of RSIMCA is only due to the simulating process: given the fact that data truly lie on a 10-dimensional space, performing (robust) dimensional reduction prior to classification evidently leads to a concealment in the grouping structure.

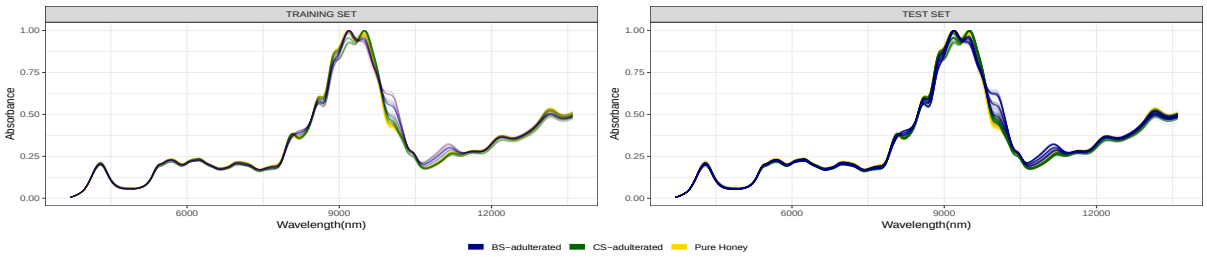

FIGURE 2.7: Midinfrared spectra for pure and contaminated honey, Irish Honey data.

The proposed methodologies were shown to be capable of dealing with data whose distribution is not exactly Gaussian, but where an effective robust decision rule can be built employing Gaussian mixture models.

2.4 Application to Midinfrared Spectroscopy of Irish Honey

The semi-supervised method introduced in Section 2.2 is employed in performing adulteration detection and classification in a food authenticity context: we consider the task of discriminating between pure and adulterated Irish Honey, where the training set itself contains unreliable samples.

2.4.1 Honey samples

Honey is defined as "the natural sweet substance, produced by honeybees from the nectar of plants or from secretions of living parts of plants, or excretions of plant-sucking insects on the living parts of plants, which the bees collect, transform by combining with specific substances of their own, deposit, dehydrate, store and leave in honeycombs to ripen and mature" (Alimentarius, 2001). Being a relatively expensive commodity to produce and extremely variable in nature, honey is prone to adulteration for economic gain: in 2015 the European Commission organized an EU coordinated control plan to assess the prevalence on the market of honey adulterated with sugars and honeys mislabelled with regard to their botanical source or geographical origin. It is therefore of prime interest to employ robust analytical methods to protect food quality and uncover its illegal adulteration.
We consider here a dataset of midinfrared spectroscopic measurements of 530 Irish honey samples. Midinfrared spectroscopy is a fast, non-invasive method for examining substances that does not require any sample preparation, it is therefore an effective procedure for collecting data to be subsequently used in food authenticity studies (Downey, 1996). The spectra measurements lie in the wavelength range of 3700 nm and 13600 nm, recorded at intervals of 35 nm, with a total of 285 absorbance values. The dataset contains 290 Pure Honey observations, while the rest of the samples are honey diluted with adulterant solutions: 120 with Dextrose Syrup and 120 with Beet Sucrose, respectively. Kelly et al., 2006 gives a thorough explanation of the adulteration process. The aim of the study is to discriminate pure honey from the adulterated samples, when varying sample size of the labelled set whilst including a percentage of wrongly labelled units. Such a scenario is plausible to be encountered in real situations, since in a context in which the final purpose is to detect potential adulterated samples it may happen that the learning data is itself not fully reliable. An example of the data structure is reported in Figure 2.7.

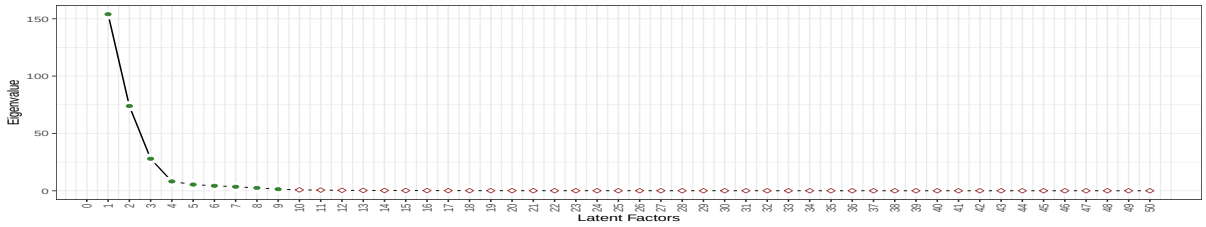

FIGURE 2.8: Cattell's scree plot (Cattell, 1966) for the first 50 eigenvalues of the robustly estimated correlation matrix, Irish Honey data. Green solid dots denote eigenvalues bigger than 1.

2.4.2 Robust dimensional reduction

Prior to perform classification and adulteration detection, a preprocessing step is needed due to the high-dimensional nature of the considered dataset ($p = 285$ variables). To do so, we robustly estimate a factor analysis model, retaining a set d of factors, $d \ll p$, to be subsequently employed with the Robust Updating Classification Rules. Formally, for each Honey sample $\mathbf{x}_i$, we postulate a factor model of the form:

$$\mathbf{x}_i = \boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{u}_i + \boldsymbol{e}_i \tag{2.15}$$

where $\boldsymbol{\mu}$ is a $p \times 1$ mean vector, $\boldsymbol{\Lambda}$ is a $p \times d$ matrix of factor loadings, $\boldsymbol{u}_i$ are the unobserved factors, assumed to be realizations of a d-variate standard normal and the errors $\boldsymbol{e}_i$ are independent realizations of $\mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$, with $\boldsymbol{\Psi}$ a $p \times p$ diagonal matrix. In such a way, the observed variables are assumed independent given the factors. For a general review on factor analysis, see for example Chapter 9 in (Mardia et al., 1979). Parameters in (2.15) are estimated employing a robust procedure based on trimming and constraints (García-Escudero et al., 2016), yielding dimensionality reduction at the same time. Note that, in Appendix A, the very same methodology is employed for detecting adulterations in wine samples. Given the robustly estimated parameters, the latent traits are computed using the regression method (Thomson, 1939):

$$\hat{\boldsymbol{u}}_i = \hat{\boldsymbol{\Lambda}}^{'} \left(\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Lambda}}^{'} + \hat{\boldsymbol{\Psi}} \right)^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) \tag{2.16}$$

The estimated factors scores $\hat{\boldsymbol{u}}_i$ will be used for the classification task reported in the upcoming Section. For the considered dataset, after a graphical exploration of Cattell's scree plot for the correlation matrix robustly estimated via MCD (Rousseeuw and Driessen, 1999), reported in Figure 2.8, we deem sufficient to set the number d of latent factors equal to 10. Parameters were estimated setting a trimming level $\alpha = 0.1$ and $c_{noise} = 1000$.

2.4.3 Classification performance

SVM, AdaBoost and RSIMCA are designed to optimally perform in a high dimensional setting. Therefore, to respect the specificity of each family of methodologies, we directly applied SVM, AdaBoost and RSIMCA on the whole spectra. For EDDA, UPCLASS, RMDA, RLDA, REDDA and RUPCLASS we preprocessed the data with the dimension reduction method described in Section 2.4.2. To discriminate between pure and adulterated honey samples, we divided the available data into a training (labelled) sample and a validation (unlabelled) sample. We investigated the effect of having different sample sizes in the labelled set, both in terms of classification accuracy and adulteration detection. Particularly, 3 proportions have been considered:

TABLE 2.3: Misclassification rates in the unlabelled set for different classification methods. Average values for 50 random splits in training and validation (three proportions are considered), standard deviations reported in parentheses.

Method	EDDA	UPCLASS	RMDA	RLDA	SVM	AdaBoost
50% Tr - 50% Te	0.033	0.065	0.291	0.1	0.025	0.036
	(0.012)	(0.049)	(0.091)	(0.02)	(0.008)	(0.011)
25% Tr - 75% Te	0.078	0.112	0.303	0.12	0.048	0.042
	(0.025)	(0.028)	(0.08)	(0.04)	(0.021)	(0.012)
10% Tr - 90% Te	0.24	0.126	0.375	0.157	0.109	0.058
	(0.031)	(0.023)	(0.065)	(0.08)	(0.036)	(0.021)

TABLE 2.4: Misclassification rates in the unlabelled set, % of wrongly labelled samples correctly trimmed in the labelled set and % of those correctly trimmed observations properly a-posteriori assigned to the Beet Sucrose group. Average values for 50 random splits in training and validation (three proportions are considered), standard deviations reported in parentheses.

		RSIMCA	REDDA	RUPCLASS
50% Tr - 50% Te	Error Rate	0.069	0.05	0.029
		(0.029)	(0.013)	(0.01)
	% Correctly Trimmed	1	0.977	1
		(0)	(0.075)	(0)
	% Correctly Assigned	1	1	1
		(0)	(0)	(0)
25% Tr - 75% Te	Error Rate	0.075	0.053	0.032
		(0.038)	(0.034)	(0.009)
	% Correctly Trimmed	1	0.88	0.96
		(0)	(0.25)	(0.145)
	% Correctly Assigned	1	0.963	1
		(0)	(0.162)	(0)
10% Tr - 90% Te	Error Rate	0.111	0.121	0.053
		(0.051)	(0.039)	(0.038)
	% Correctly Trimmed	0.99	0.47	0.73
		(0.071)	(0.238)	(0.381)
	% Correctly Assigned	0.99	0.72	0.84
		(0.019)	(0.071)	(0.37)

50% - 50% , 25% - 75% and 10% - 90% for splitting data into training and validation set, respectively, within each group. For each split, 10% of the Beet Sucrose adulterated samples were incorrectly labelled as Pure Honey in the training set, adding class noise in the discrimination task. The trimming levels α_l and α_u were set equal to 0.12 and 0.05, respectively. Table 2.3 and 2.4 summarize the accuracy results employing different classification approaches under the described scenarios. Careful investigation has been dedicated to measuring the ability of the robust methodologies in correctly determining (i.e., trimming) the 10% of incorrectly labelled samples, that is, units adulterated with Beet Sucrose and erroneously labelled as Pure Honey:

such information, only relevant for RSIMCA, REDDA and RUPCLASS models, is reported in Table 2.4. *% Correctly Trimmed* indicates the class noise percentage correctly detected by the impartial trimming. For the recognized class noise, *% Correctly Assigned* indicates the percentage of units properly a-posteriori assigned to the Beet Sucrose group. RSIMCA performs remarkably well in identifying the adulterated units, even though the classification accuracy is lower than the one obtained employing RUPCLASS model. As expected, the semi-supervised approach performs much better in terms of classification rate when the labelled sample size is small. Comparing the error rate of the robust techniques with the other methods in Table 2.3 we notice how powerful classifiers like SVM and AdaBoost work well also in dealing with adulterated datasets: SVM error rate in the 50% Tr - 50% Te is on average lower than the one obtained with RUPCLASS. However, when the labelled sample size decreases a semi-supervised approach is preferable: RUPCLASS reports the lowest error rate for both 25% Tr - 75% Te and 10% Tr - 90% Te scenarios. VEV and VVV models have been almost always chosen: model selection was performed through the Robust criteria defined in Section 2.2.4.
Results in Table 2.4 show that the proposed methodology is effective not only for accurately robustifying the parameter estimates, but also for efficiently detecting observations affected by class noise, firstly by trimming and subsequently by correctly assigning them: a critical information that cannot be obtained with standard classification methods like SVM and AdaBoost.

2.5 Concluding remarks

In this chapter we have proposed a robust modification to a family of semi-supervised patterned models, for performing classification in presence of both class and attribute noise.
We have shown that our methodology effectively addresses the issues generated by these two noise types, by identifying wrongly labelled units (noise in the response variable) and corrupted attributes in units (noise in the explanatory variables). Robust parameter estimates can therefore be obtained by excluding the noisy observations from the estimation procedure, both in the training set, and in the test set. Our proposal has been based on incorporating impartial trimming and eigenvalue-ratio constraints in previous semi-supervised methods. We have adapted the trimming procedure to the two different frameworks, i.e., for the labelled units and the unlabelled ones. After completing the robust estimation process, trimmed observations can be classified as well, by the usual Bayes rule. This final step allows the researcher to detect whether one observation is indeed extreme in terms of its attributes or it has been wrongly assigned to a different class. Such feature seems particularly desirable in food authenticity applications, where, due to imprecise readings and fraudulent units, it is likely to have label noise also within the labelled set. Some simulations, and a study on real data from pure and adulterated Honey samples, have shown the effectiveness of our proposal.
As an open point for further research, an automatic procedure for selecting reasonable values for the labelled and unlabelled trimming levels, along the lines of Dotto et al., 2018, is currently under study.

2.6 Appendix A

Proof of Proposition 1: Considering the random variable $\mathcal{Z}_{mg}$ corresponding to z_{mg}, the E-step on the $(k+1)$th iteration requires the calculation of the conditional expectation of $\mathcal{Z}_{mg}$ given $\mathbf{y}_m$:

$$\begin{aligned} E_{\hat{\boldsymbol{\theta}}^{(k)}}(\mathcal{Z}_{mg}|\mathbf{y}_m) &= \mathbb{P}\left(\mathcal{Z}_{mg}=1|\mathbf{y}_m;\hat{\theta}^{(k)}\right) = \\ &= \frac{\mathbb{P}\left(\mathbf{y}_m|\mathcal{Z}_{mg}=1;\hat{\theta}^{(k)}\right)\mathbb{P}\left(\mathcal{Z}_{mg}=1;\hat{\theta}^{(k)}\right)}{\sum_{j=1}^{G}\mathbb{P}\left(\mathbf{y}_m|\mathcal{Z}_{mj}=1;\hat{\theta}^{(k)}\right)\mathbb{P}\left(\mathcal{Z}_{mj}=1;\hat{\theta}^{(k)}\right)} = \\ &= \frac{\hat{\tau}_g^{(k)}\phi\left(\mathbf{y}_m;\hat{\boldsymbol{\mu}}_g^{(k)},\hat{\boldsymbol{\Sigma}}_g^{(k)}\right)}{\sum_{j=1}^{G}\hat{\tau}_j^{(k)}\phi\left(\mathbf{y}_m;\hat{\boldsymbol{\mu}}_j^{(k)},\hat{\boldsymbol{\Sigma}}_j^{(k)}\right)} = \\ &= \hat{z}_{mg}^{(k+1)} \qquad\qquad g=1,\ldots,G; \quad m=1,\ldots,M. \end{aligned} \tag{2.17}$$

Therefore, the Q function, to be maximized with respect to $\boldsymbol{\theta}$ in the M-step, is given by

$$\begin{aligned} Q(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(k)}) = &\sum_{n=1}^{N}\zeta(\mathbf{x}_n)\sum_{g=1}^{G}l_{ng}\log\left[\tau_g\phi(\mathbf{x}_n;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\right]+ \\ &+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\sum_{g=1}^{G}\hat{z}_{mg}^{(k+1)}\log\left[\tau_g\phi(\mathbf{y}_m;\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\right]. \end{aligned} \tag{2.18}$$

The maximization of (2.18) according to the mixture proportion τ_g, $\sum_{j=1}^{G}\tau_j = 1$ is solved considering the Lagrangian $\mathcal{L}(\boldsymbol{\theta},\kappa)$:

$$\mathcal{L}(\boldsymbol{\theta},\kappa) = Q(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}^{(k)}) - \kappa\left(\sum_{j=1}^{G}\tau_j - 1\right) \tag{2.19}$$

with κ the Lagrangian coefficient. The partial derivative of (2.19) with respect to τ_g has the form:

$$\frac{\partial}{\partial\tau_g}\mathcal{L}(\boldsymbol{\theta},\kappa) = \frac{\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}}{\tau_g} + \frac{\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}}{\tau_g} - \kappa \tag{2.20}$$

and setting (2.20) equal to 0 for all $g=1,\ldots,G$ we obtain:

$$\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng} + \sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)} - \kappa\tau_g = 0. \tag{2.21}$$

Summing (2.21) over g, $g=1,\ldots,G$, provides the value of $\kappa = \lceil N(1-\alpha_l)\rceil + M(1-\alpha_u)\rceil$ and substituting it in the previous expression yields the ML estimate for τ_g:

$$\hat{\tau}_g^{(k+1)} = \frac{\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng} + \sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}}{\lceil N(1-\alpha_l)\rceil + \lceil M(1-\alpha_u)\rceil} \quad g=1,\ldots,G. \tag{2.22}$$

The partial derivative of (2.18) with respect to the mean vector $\boldsymbol{\mu}_g$ reads:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}_g} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}) &= -\boldsymbol{\Sigma}_g^{-1}\left[\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)\right]= \\ &= -\boldsymbol{\Sigma}_g^{-1}\left[\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}\mathbf{x}_n+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\mathbf{y}_m+\right. \\ &\left.-\boldsymbol{\mu}_g\left(\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\right)\right].\end{aligned} \tag{2.23}$$

Equating (2.23) to 0 and rearranging terms provides the ML estimate of $\boldsymbol{\mu}_g$:

$$\hat{\boldsymbol{\mu}}_g^{(k+1)}=\frac{\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}\mathbf{x}_n+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\mathbf{y}_m}{\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}}\quad g=1,\ldots,G. \tag{2.24}$$

Discarding quantities that do not depend on $\boldsymbol{\Sigma}_g$, we can rewrite (2.18) as follows:

$$\begin{aligned}&\sum_{n=1}^{N}\sum_{g=1}^{G}\zeta(\mathbf{x}_n)l_{ng}\left(\mathbf{x}_n\right)\left[-\log\left|\boldsymbol{\Sigma}_g\right|^{1/2}-\frac{1}{2}\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)'\boldsymbol{\Sigma}_g^{-1}\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)\right]+ \\ &\quad+\sum_{m=1}^{M}\sum_{g=1}^{G}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\left(\mathbf{y}_m\right)\left[-\log\left|\boldsymbol{\Sigma}_g\right|^{1/2}-\frac{1}{2}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)'\boldsymbol{\Sigma}_g^{-1}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)\right]= \\ &=-\frac{1}{2}\left[\sum_{n=1}^{N}\sum_{g=1}^{G}\zeta(\mathbf{x}_n)l_{ng}\left(\mathbf{x}_n\right)\log\left|\boldsymbol{\Sigma}_g\right|+\sum_{n=1}^{N}\sum_{g=1}^{G}\zeta(\mathbf{x}_n)l_{ng}\left[\underbrace{\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)'\boldsymbol{\Sigma}_g^{-1}\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)}_{\text{a scalar}}\right]+\right. \\ &\left.+\sum_{m=1}^{M}\sum_{g=1}^{G}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\left(\mathbf{y}_m\right)\log\left|\boldsymbol{\Sigma}_g\right|+\sum_{m=1}^{M}\sum_{g=1}^{G}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\left[\underbrace{\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)'\boldsymbol{\Sigma}_g^{-1}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)}_{\text{a scalar}}\right]\right]= \\ &=-\frac{1}{2}\left[\sum_{g=1}^{G}\log\left|\boldsymbol{\Sigma}_g\right|\left(\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}\left(\mathbf{x}_n\right)+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\left(\mathbf{y}_m\right)\right)+\right. \\ &\quad+\sum_{n=1}^{N}\sum_{g=1}^{G}\zeta(\mathbf{x}_n)l_{ng}tr\left[\boldsymbol{\Sigma}_g^{-1}\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)'\right]+ \\ &\left.\quad+\sum_{m=1}^{M}\sum_{g=1}^{G}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}tr\left[\boldsymbol{\Sigma}_g^{-1}\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)'\right]\right]=\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2}\left[\sum_{g=1}^{G}\log|\boldsymbol{\Sigma}_g|\left(\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}(\mathbf{x}_n)+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}(\mathbf{y}_m)\right)+\right.\\
&\qquad\left.+\sum_{g=1}^{G}tr\left[\boldsymbol{\Sigma}_g^{-1}\boldsymbol{W}_g^X\right]+\sum_{g=1}^{G}tr\left[\boldsymbol{\Sigma}_g^{-1}\boldsymbol{W}_g^Y\right]\right]=\\
&= -\frac{1}{2}\left[\sum_{g=1}^{G}\log|\boldsymbol{\Sigma}_g|\left(\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}(\mathbf{x}_n)+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}(\mathbf{y}_m)\right)+\sum_{g=1}^{G}tr\left[\boldsymbol{\Sigma}_g^{-1}\left(\boldsymbol{W}_g^X+\boldsymbol{W}_g^Y\right)\right]\right]
\end{aligned}
\tag{2.25}
$$

where $\boldsymbol{W}_g^X=\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}\left[\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_g\right)'\right]$ and $\boldsymbol{W}_g^Y=\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}\left[\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)\left(\mathbf{y}_m-\boldsymbol{\mu}_g\right)'\right]$.
Finally, considering the eigenvalue decomposition $\boldsymbol{\Sigma}_g=\lambda_g\boldsymbol{D}_g\boldsymbol{A}_g\boldsymbol{D}_g'$, (2.25) simplifies to:

$$
\begin{aligned}
&-\frac{1}{2}\left[\sum_{g=1}^{G}p\log\lambda_g\left(\sum_{n=1}^{N}\zeta(\mathbf{x}_n)l_{ng}(\mathbf{x}_n)+\sum_{m=1}^{M}\varphi(\mathbf{y}_m)\hat{z}_{mg}^{(k+1)}(\mathbf{y}_m)\right)+\right.\\
&\left.+\sum_{g=1}^{G}\frac{1}{\lambda_g}tr\left[\boldsymbol{D}_g\boldsymbol{A}^{-1}\boldsymbol{D}_g'\left(\boldsymbol{W}_g^X+\boldsymbol{W}_g^Y\right)\right]\right]
\end{aligned}
\tag{2.26}
$$

The partial derivative of (2.26) with respect to $(\lambda_g,\boldsymbol{A}_g,\boldsymbol{D}_g)$ depends on the considered patterned structure: for a thorough derivation the reader is referred to Bensmail and Celeux, 1996. If (1.20) is not satisfied, the constraints are enforced as detailed in Appendix C (Section 2.8). Lastly, notice that in performing the concentration step the optimal observations of both training and test sets are retained, i.e. the ones with the highest contribution to the objective function.
The afore-described procedure falls within the structure of a general EM algorithm, for which the likelihood function does not decrease after an EM iteration, as shown in Dempster et al., 1977 and reported in page 78 of McLachlan and Krishnan, 2008.

□

2.7 Appendix B

This appendix details the structure of the Simulation Study in Section 2.3.2. We consider a data generating process given by a mixture of $G=4$ components of multivariate t-distributions (McLachlan and Peel, 1998; Peel and McLachlan, 2000), according to the following parameters:

$$
\begin{aligned}
\boldsymbol{\tau}&=(0.2,0.4,0.1,0.3)',\quad \nu=6,\\
\boldsymbol{\mu}_1&=(0,0,0,0,0,0,0,0,0,0)',\\
\boldsymbol{\mu}_2&=(4,-4,4,-4,4,-4,4,-4,4,-4)',\\
\boldsymbol{\mu}_3&=(0,0,7,7,7,3,6,8,-4,-4)',\\
\boldsymbol{\mu}_4&=(8,0,8,0,8,0,8,0,8,0,8)',\\
\boldsymbol{\Sigma}_1&=diag(1,1,1,1,1,1,1,1,1,1),\\
\boldsymbol{\Sigma}_2&=diag(2,2,2,2,2,2,2,2,2,2),
\end{aligned}
$$

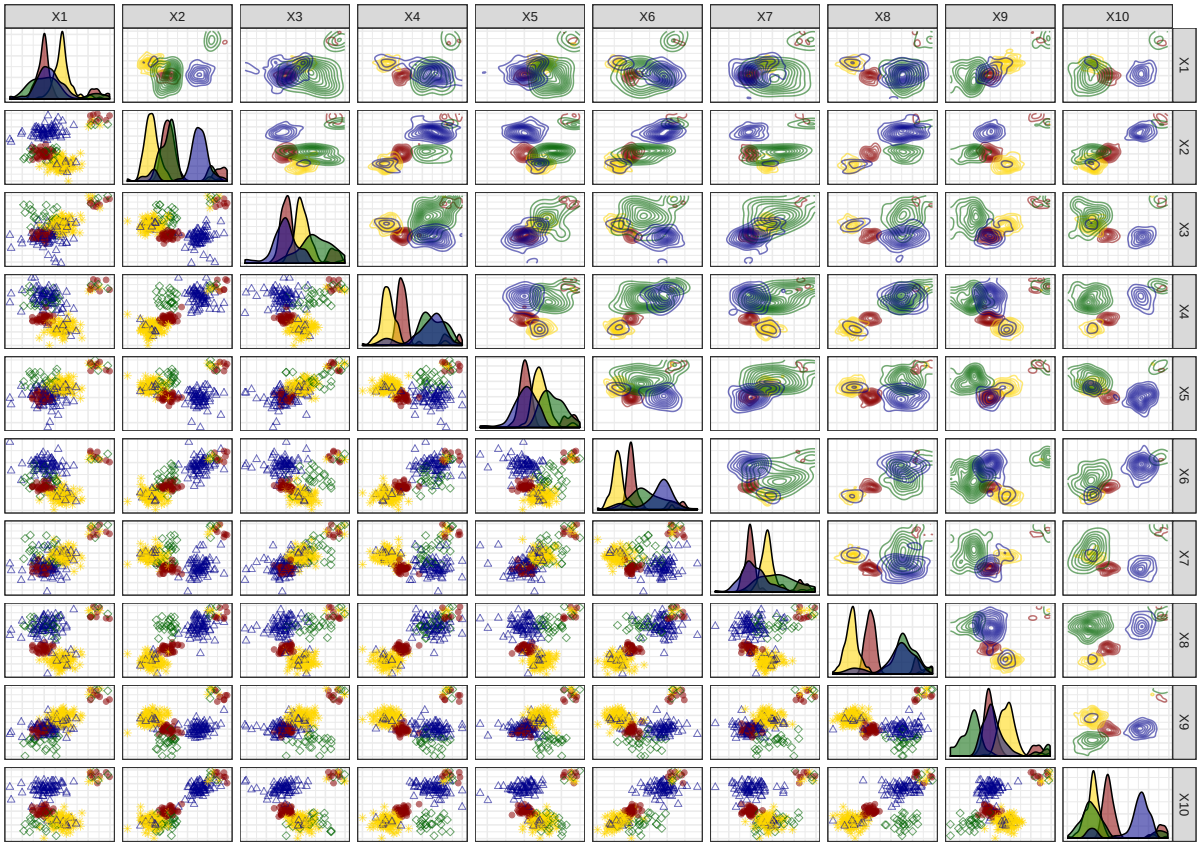

FIGURE 2.9: Generalized pairs plot of the simulated data under the Simulation Setup described in 2.3.2. Both label noise and outliers are present in the data units.

$$\boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} 5.05 & 1.26 & -0.35 & -0.00 & -1.04 & -1.35 & 0.29 & 0.07 & 0.69 & 1.17 \\ 1.26 & 2.57 & 0.17 & 0.00 & 0.27 & 0.11 & 0.61 & 0.11 & 0.59 & 0.89 \\ -0.35 & 0.17 & 6.74 & -0.00 & -0.26 & -0.31 & -0.01 & 0.00 & 0.08 & 0.14 \\ -0.00 & 0.00 & -0.00 & 5.47 & -0.00 & -0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -1.04 & 0.27 & -0.26 & -0.00 & 6.80 & -0.76 & -0.12 & -0.01 & 0.09 & 0.21 \\ -1.35 & 0.11 & -0.31 & -0.00 & -0.76 & 7.75 & -0.26 & -0.04 & -0.03 & 0.03 \\ 0.29 & 0.61 & -0.01 & 0.00 & -0.12 & -0.26 & 4.76 & 0.06 & 0.38 & 0.60 \\ 0.07 & 0.11 & 0.00 & 0.00 & -0.01 & -0.04 & 0.06 & 4.18 & 0.07 & 0.11 \\ 0.69 & 0.59 & 0.08 & 0.00 & 0.09 & -0.03 & 0.38 & 0.07 & 3.23 & 0.60 \\ 1.17 & 0.89 & 0.14 & 0.00 & 0.21 & 0.03 & 0.60 & 0.11 & 0.60 & 3.24 \end{bmatrix}.$$

A generalized pairs plot of contaminated labelled units under the afore-described Simulation Setup is reported in Figure 2.9.

2.8 Appendix C

This final Section presents feasible and computationally efficient algorithms for enforcing the eigenvalue-ratio constraint according to the different patterned models in Table 2.1. At the $k-$th iteration of the M-step, the goal is to update the estimates for the covariance matrices

$\hat{\boldsymbol{\Sigma}}_g^{(k+1)} = \hat{\lambda}_g^{(k+1)} \hat{\boldsymbol{D}}_g^{(k+1)} \hat{\boldsymbol{A}}_g^{(k+1)} \hat{\boldsymbol{D}}_g^{\prime(k+1)}$, $g = 1, \ldots, G$ such that,

$$\frac{\max_{g=1\ldots G} \max_{l=1\ldots p} \hat{\lambda}_g^{(k+1)} \hat{a}_{lg}^{(k+1)}}{\min_{g=1\ldots G} \min_{l=1\ldots p} \hat{\lambda}_g^{(k+1)} \hat{a}_{lg}^{(k+1)}} \leq c \tag{2.27}$$

where $\hat{a}_{lg}^{(k+1)}$ indicates the diagonal entries of matrix $\hat{\boldsymbol{A}}_g^{(k+1)}$. Denote with $\hat{\Sigma}_g^U = \hat{\lambda}_g^U \hat{\boldsymbol{D}}_g^U \hat{\boldsymbol{A}}_g^U \hat{\boldsymbol{D}}_g^{\prime U}$ the estimates for the variance covariance matrices obtained following Bensmail and Celeux, 1996 without enforcing the eigenvalues-ratio restriction in (2.27). Lastly, denote with $\hat{\boldsymbol{\Delta}}_g^U = \hat{\lambda}_g^U \hat{\boldsymbol{A}}_g^U$ the matrix of eigenvalues for $\hat{\boldsymbol{\Sigma}}_g^U$, with diagonal entries $\hat{d}_{lg}^U = \hat{\lambda}_g^U \hat{a}_{lg}^U$, $l = 1, \ldots, p$.

Constrained maximization for VII, VVI and VVV models

1. Compute $\boldsymbol{\Delta}_g$ applying the *optimal truncation operator* defined in Fritz et al., 2013 to $\left\{\hat{\boldsymbol{\Delta}}_1^U, \ldots, \hat{\boldsymbol{\Delta}}_G^U\right\}$, under condition (2.27)
2. Set $\hat{\lambda}_g^{(k+1)} = |\boldsymbol{\Delta}_g|^{1/p}$, $\hat{\boldsymbol{A}}_g^{(k+1)} = \frac{1}{\hat{\lambda}_g^{(k+1)}} \boldsymbol{\Delta}_g$, $\hat{\boldsymbol{D}}_g^{(k+1)} = \hat{\boldsymbol{D}}_g^U$

Constrained maximization for VVE model

1. Compute $\boldsymbol{\Delta}_g$ applying the *optimal truncation operator* defined in Fritz et al., 2013 to $\left\{\hat{\boldsymbol{\Delta}}_1^U, \ldots, \hat{\boldsymbol{\Delta}}_G^U\right\}$, under condition (2.27)
2. Given $\boldsymbol{\Delta}_g$, compute the common principal components $\boldsymbol{D}$ via, for example, a majorization-minimization (MM) algorithm (Browne and McNicholas, 2014)
3. Set $\hat{\lambda}_g^{(k+1)} = |\boldsymbol{\Delta}_g|^{1/p}$, $\hat{\boldsymbol{A}}_g^{(k+1)} = \frac{1}{\hat{\lambda}_g^{(k+1)}} \boldsymbol{\Delta}_g$, $\hat{\boldsymbol{D}}_g^{(k+1)} = \boldsymbol{D}$

Constrained maximization for EVI, EVV models

1. Compute $\boldsymbol{\Delta}_g$ applying the *optimal truncation operator* defined in Fritz et al., 2013 to $\left\{\hat{\boldsymbol{\Delta}}_1^U, \ldots, \hat{\boldsymbol{\Delta}}_G^U\right\}$, under condition (2.27)
2. Compute $\boldsymbol{\Delta}_g^\star$ constraining $\boldsymbol{\Delta}_g$ such that $\boldsymbol{\Delta}_g^\star = \lambda^\star \boldsymbol{A}_g^\star$. That is, constraining $|\boldsymbol{\Delta}_g^\star|$ to be equal across groups (Maronna and Jacovkis, 1974; Gallegos, 2002). Details are given in Section 3.2 of Fritz et al., 2012
3. Iterate $1 - 2$ until (2.27) is satisfied
4. Set $\hat{\lambda}_g^{(k+1)} = \lambda^\star$, $\hat{\boldsymbol{A}}_g^{(k+1)} = \boldsymbol{A}_g^\star$, $\hat{\boldsymbol{D}}_g^{(k+1)} = \hat{\boldsymbol{D}}_g^U$

Constrained maximization for EVE model

1. Compute $\boldsymbol{\Delta}_g$ applying the *optimal truncation operator* defined in Fritz et al., 2013 to $\left\{\hat{\boldsymbol{\Delta}}_1^U, \ldots, \hat{\boldsymbol{\Delta}}_G^U\right\}$, under condition (2.27)
2. Compute $\boldsymbol{\Delta}_g^\star$ constraining $\boldsymbol{\Delta}_g$ such that $\boldsymbol{\Delta}_g^\star = \lambda^\star \boldsymbol{A}_g^\star$. Details are given in Section 3.2 of Fritz et al., 2012
3. Iterate $1 - 2$ until (2.27) is satisfied

4. Given $\boldsymbol{A}_g^\star$, compute the common principal components $\boldsymbol{D}$ via, for example, a majorization-minimization (MM) algorithm (Browne and McNicholas, 2014)

5. Set $\hat{\lambda}_g^{(k+1)} = \lambda^\star$, $\hat{\boldsymbol{A}}_g^{(k+1)} = \boldsymbol{A}_g^\star$, $\hat{\boldsymbol{D}}_g^{(k+1)} = \boldsymbol{D}$

Constrained maximization for VEI, VEV models

1. Set $\boldsymbol{\Delta}_g = \hat{\boldsymbol{\Delta}}_g^U$
2. Set $\lambda_g^\star = \hat{\lambda}_g^U$, $g = 1, \ldots, G$
3. Compute $\boldsymbol{\Delta}_g^\star$ applying the *optimal truncation operator* defined in Fritz et al., 2013 to $\{\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_G\}$, under condition (2.27)
4. Compute $\boldsymbol{A}^\star = \sum_{g=1}^G \frac{1}{\lambda_g^\star}\boldsymbol{\Delta}_g^\star \Big/ \left|\sum_{g=1}^G \frac{1}{\lambda_g^\star}\boldsymbol{\Delta}_g^\star\right|^{1/p}$
5. Compute $\lambda_g^\star = \frac{1}{p} tr\left(\boldsymbol{\Delta}_g^\star \boldsymbol{A}^{\star -1}\right)$
6. Set $\boldsymbol{\Delta}_g = \lambda_g^\star \boldsymbol{A}^\star$
7. Iterate $3-6$ until (2.27) is satisfied
8. Set $\hat{\lambda}_g^{(k+1)} = \lambda_g^\star$, $\hat{\boldsymbol{A}}_g^{(k+1)} = \boldsymbol{A}^\star$, $\hat{\boldsymbol{D}}_g^{(k+1)} = \hat{\boldsymbol{D}}_g^U$

Constrained maximization for VEE model

1. Set $\boldsymbol{K}_g = \hat{\boldsymbol{\Sigma}}_g^U$
2. Set $\lambda_g^\star = \hat{\lambda}_g^U$, $g = 1, \ldots, G$
3. Compute $\boldsymbol{K}_g^\star$ applying the *optimal truncation operator* defined in Fritz et al., 2013 to $\{\boldsymbol{K}_1, \ldots, \boldsymbol{K}_G\}$, under condition (2.27)
4. Compute $\boldsymbol{C}^\star = \sum_{g=1}^G \frac{1}{\lambda_g^\star}\boldsymbol{K}_g^\star \Big/ \left|\sum_{g=1}^G \frac{1}{\lambda_g^\star}\boldsymbol{K}_g^\star\right|^{1/p}$
5. Compute $\lambda_g^\star = \frac{1}{p} tr\left(\boldsymbol{K}_g^\star \boldsymbol{C}^{\star -1}\right)$
6. Set $\boldsymbol{K}_g = \lambda_g^\star \boldsymbol{C}^\star$
7. Iterate $3-6$ until (2.27) is satisfied
8. Considering the spectral decomposition for $\boldsymbol{C}^\star = \boldsymbol{D}^\star \boldsymbol{A}^\star \boldsymbol{D}^{\star\prime}$, set $\hat{\lambda}_g^{(k+1)} = \lambda_g^\star$, $\hat{\boldsymbol{A}}_g^{(k+1)} = \boldsymbol{A}^\star$, $\hat{\boldsymbol{D}}_g^{(k+1)} = \boldsymbol{D}^\star$

Chapter 3

Anomaly and Novelty detection for robust semi-supervised learning

Based on:

Cappozzo, A., Greselin, F., Murphy, T. B.
"Anomaly and Novelty detection for robust semi-supervised learning"
Submitted

3.1 Introduction

The standard classification framework assumes that a set of outlier-free and correctly labeled units are available for each and every group within the population of interest. Given these strong assumptions, the labeled observations are employed to build a classification rule for assigning unlabeled samples to one of the known groups. However, as seen in the previous chapter, real-world training set may contain noise, that can adversely impact the classification performances of induced classifiers. Two sources of anomalies may appear:

- label noise, that is wrongly labeled data, represented in the left panel of Figure 3.1;
- feature noise, whenever erroneous measurements are given to some units, as shown in central panel of Figure 3.1.

Moreover, when new data are given to the classifier, extra classes, not observed earlier in the training set, may appear (see right panel of Figure 3.1). Therefore, for a classification method to succeed when the aforementioned assumptions are violated, both anomalies and novelties need to be identified and categorized as such. Since neither anomaly nor novelty detection is universally defined in the literature, we hereafter characterize their meaning in a classification context.
Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior (Chandola et al., 2009). Particularly, in discriminant analysis, we identify anomalies with both attribute and class noise as they were defined in Chapter 2.
As mentioned in Section 1.3.4, novelty detection is the identification of new or unknown data or signal that a machine learning system is not aware of during training (Markou and Singh, 2003). Particularly, in a classification context, we indicate with novelty a group of observations in the test set that displays a common pattern not previously encountered in the training set, and can therefore be identified as a novel or hidden class. From a stochastic viewpoint, this

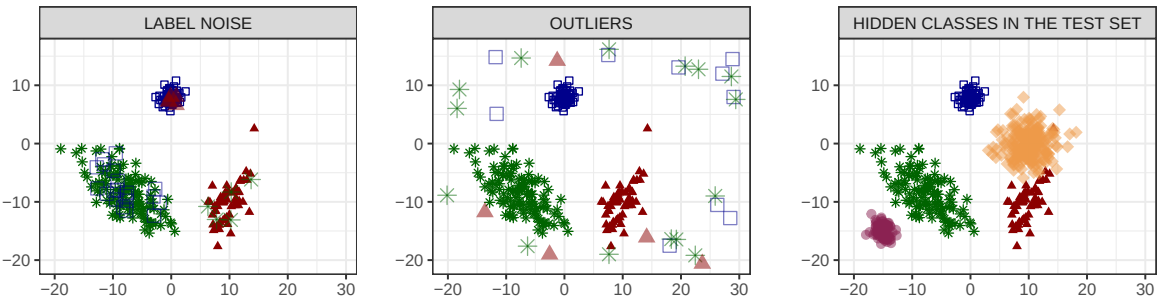

FIGURE 3.1: Different classification scenarios in which the training set presents label noise (left panel), outliers (central panel) and in which the test set contains groups not previously encountered in the learning phase (right panel).

is equivalent to assuming that the probability distribution of the labels differs in the labelled and unlabelled sets, as a result of an unknown sample rejection process. More generally, the difference between the joint distribution of labels and input variables in the training and test sets is denoted as "dataset shift" problem: for a thorough description of the topic, the interested reader is referred to Quionero-Candela et al., 2009.

The ever-increasing complexity of real-world datasets motivates the development of methods that bridges the advantages of both novelty and anomaly detection classifiers. For instance, human supervision is required in bio-medical applications: this costly and difficult procedure is prone to introduce label noise in the training set, while some less common or yet unknown patterns might be left completely undiscovered. Another example comes from the food authenticity domain: adulterated samples are nothing but wrongly-labeled units in the training set, whilst new and unidentified adulterants and/or state-of-the-art adulteration procedures are unobserved classes that need to be discovered. Also, in food science, the state-of-the-art approach for determining food origin is to employ microbiome analysis as a discriminating signature: a promising application for identifying wine provenance is reported in Section 3.4.

In the present chapter we introduce a joint anomaly and novelty detection model-based method for performing classification in situations where class-memberships are unreliable for some training units (label noise), a proportion of observations departs from the main structure of the data (outliers) and new groups in the test set were not encountered earlier in the learning phase (unobserved classes). Our proposal models the unobserved classes as arising from a mixture of multivariate normal densities, whilst avoiding to impose any distributional assumption for the noise component. In detail, we extend in three ways the original AMDA model briefly summarized in Section 1.3.4. Firstly, we account for both attribute and class noise that can be present in the samples employing impartial trimming. Secondly, we consider a more flexible class of learners with the parsimonious parametrization based on the eigen-decomposition of Banfield and Raftery, 1993 and Celeux and Govaert, 1995, described in Section 1.2.1. Thirdly, we deal with a constrained parameter estimation to avoid convergence to degenerate solutions and to protect the estimates from spurious local maximizers that are likely to arise when searching for unobserved classes (see Section 3.2.6). Such extended model is denoted as Robust and Adaptive Eigen Decomposition Discriminant Analysis (RAEDDA).

The rest of the chapter is organized as follows. In Section 3.2, we describe formulation, inference aspects and selection criteria for the novel RAEDDA model. Experimental results for evaluating the features of the proposed method are covered in Section 3.3. Section 3.4 presents

a real data application, involving the detection of grapes origin when only a subset of the sampling sites are known in advance and learning units are not to be entirely trusted. Section 3.5 concludes the chapter with some remarks and directions for future research. Appendix A (Section 3.6) reports closed-form solutions for the covariance estimation in the discovery phase, within an inductive framework (see Section 3.2.3), for all 14 models of Celeux and Govaert, 1995.

3.2 Robust and Adaptive EDDA

3.2.1 Model formulation

In this Section we introduce a novel flexible procedure that serves the purpose of performing reliable supervised classification when dealing with label noise, outliers and unobserved classes.
The model is based on the definition of *trimmed log-likelihood* (Neykov et al., 2007) under a Gaussian mixture framework, employing impartial trimming and eigenvalue-ratio restrictions for robust parameter estimation and identification of mislabeled and outlying observations, as proposed in Chapter 2 of the present manuscript. Furthermore, as for the ADMA model detailed in Section 1.3.4, we assume that only a subset of the whole set of classes was observed in the training sample: hidden groups in the test data may be detected. Considering the same notation introduced in Section 1.3.4 and given a sample of N training and M test data, we construct a procedure for maximizing the *trimmed observed data log-likelihood:*

$$\begin{aligned}\ell_{trim}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{X}, \mathbf{Y}, \mathbf{l}) = \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \sum_{g=1}^{G} \mathrm{l}_{ng} \log \Big(\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \Big) + \\ + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \log \left(\sum_{g=1}^{E} \tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \right)\end{aligned} \tag{3.1}$$

where $\phi(\cdot; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ represents the multivariate Gaussian density with mean vector $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$; the functions $\zeta(\cdot)$ and $\varphi(\cdot)$ are indicator functions that determine whether each observation contributes or not to the trimmed likelihood, such that only $\sum_{n=1}^{N} \zeta(\mathbf{x}_n) = \lceil N(1-\alpha_l) \rceil$ and $\sum_{m=1}^{M} \varphi(\mathbf{y}_m) = \lceil M(1-\alpha_u) \rceil$ terms are not null in (3.1). The *labeled trimming level* α_l and the *unlabeled trimming level* identify the fixed fraction of observations, respectively belonging to the training and test sets, that are tentatively assumed to be unreliable at each iteration during the maximization of (3.1), likewise what done in Chapter 2. Once the trimming levels are specified, the proposed maximization process returns robustly estimated parameter values (see Sections 3.2.2 and 3.2.3 for details). Finally notice that only G groups in (3.1), out of the $E \geq G$ present in the population, were already captured within the labeled units, as in the AMDA model.
To introduce flexibility and parsimony, we consider the eigen-decomposition for the covariance matrices of Banfield and Raftery, 1993 and Celeux and Govaert, 1995 as detailed in Section 1.2.1: since our proposal generalizes the original EDDA including robust estimation and adaptive learning, the name Robust and Adaptive Eigenvalue Decomposition Discriminant Analysis (RAEDDA) seems appropriate. Two alternative estimation procedures for maximizing (3.1) are proposed. The transductive approach (see Section 3.2.2) works on the joint union of learning and test sets to estimate model parameters. The inductive approach (see Section 3.2.3), instead, consists of two distinctive phases: in the first one the training set is employed

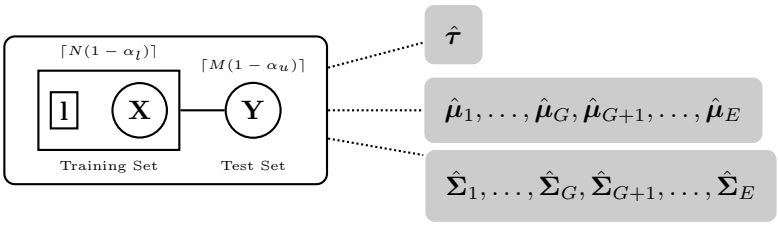

FIGURE 3.2: General framework of the *robust transductive estimation* approach: $\lceil N(1-\alpha_l)\rceil$ observations in the training and $\lceil M(1-\alpha_u)\rceil$ observations in the test are jointly employed in estimating parameters for the known and hidden classes.

for estimating parameters of the G known groups; in the second phase the unlabeled observations are assigned to the known groups whilst searching for new classes and estimating their parameters. Computational aspects for both procedures are detailed in the next Sections.

3.2.2 Estimation procedure: transductive approach

Transductive inference considers the joint exploitation of training and test sets to solve a specific learning problem (Vapnik, 2000; Kasabov and Shaoning Pang, 2003). Transductive reasoning is applied for instance in semi-supervised classification methods: the data generating process is assumed to be the same for labeled and unlabeled observations and hence units coming from both sets can be used to build the classification rule. For instance, the methodology developed in Chapter 2 falls within this framework. The present context is more general than semi-supervised learning since the total number of classes E might be strictly larger than the G ones observed in the training set. Therefore, an ad-hoc procedure needs to be introduced: a graphical representation of the transductive approach is reported in Figure 3.2.
An adaptation of the EM algorithm (Dempster et al., 1977) that includes a Concentration Step and an eigenvalue-ratio restriction is employed for maximizing (3.1). The former serves the purpose of enforcing impartial trimming in both labeled and unlabeled units at each step of the algorithm, whereas the latter prevents the procedure to be trapped in spurious local maximizers that may arise whenever a random pattern in the test is wrongly fitted to form a hidden class (see Section 3.2.6). Likewise for the methodology in Chapter 2, the considered eigenvalue-ratio restriction is the one detailed in Section 1.3.2. In addition, the feasible algorithms developed in Section 2.8 are employed also here when different specifications for the covariance matrices are considered.
Under the transductive learning phase, the *trimmed complete data log-likelihood* is as follows:

$$\begin{aligned}\ell_{trim_c}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{X}, \mathbf{Y}, \mathbf{l}, \mathbf{z}) = &\sum_{n=1}^{N} \zeta(\mathbf{x}_n) \sum_{g=1}^{G} \mathrm{l}_{ng} \log\left(\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right) + \\ &+ \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \sum_{g=1}^{E} z_{mg} \log\left(\tau_g \phi(\mathbf{y}_m; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right)\end{aligned} \tag{3.2}$$

The next steps detail a constrained EM algorithm for jointly estimating model parameters (see Figure 3.2) whilst searching for new classes and outliers.
Unlike what is suggested in Bouveyron, 2014, the EM initialization is here performed in two subsequent steps for preventing outliers to spoil the starting values and henceforth driving the entire algorithm to reach uninteresting solutions. We firstly make use of a robust procedure to

obtain a set of parameter estimates $\{\bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ for the known groups G using only the training set. Afterwards, if $E > G$, we randomly initialize the parameters for the $H = E - G$ hidden classes taking advantage of the known groups structure learned in the previous step. Notice that, as in Bouveyron, 2014, at this moment the number of hidden classes E is assumed to be known: we will discuss its estimation later (see Section 3.2.5).

- *Robust Initialization for the G known groups*: set $k = 0$. Employing only the labeled data, we obtain robust starting values for $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, G$ as follows:

  1. For each known class g, draw a random $(p+1)$-subset J_g and compute its empirical mean $\bar{\boldsymbol{\mu}}_g^{(0)}$ and variance covariance matrix $\bar{\boldsymbol{\Sigma}}_g^{(0)}$ according to the considered parsimonious structure.
  2. Set
  $$\begin{aligned}\{\bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\} &= \{\bar{\tau}_1, \ldots, \bar{\tau}_G, \bar{\boldsymbol{\mu}}_1, \ldots, \bar{\boldsymbol{\mu}}_G, \bar{\boldsymbol{\Sigma}}_1, \ldots, \bar{\boldsymbol{\Sigma}}_G\} = \\ &= \{\bar{\tau}_1^{(0)}, \ldots, \bar{\tau}_G^{(0)}, \bar{\boldsymbol{\mu}}_1^{(0)}, \ldots, \bar{\boldsymbol{\mu}}_G^{(0)}, \bar{\boldsymbol{\Sigma}}_1^{(0)}, \ldots, \bar{\boldsymbol{\Sigma}}_G^{(0)}\}\end{aligned}$$
  where $\bar{\tau}_1^{(0)} = \ldots = \bar{\tau}_G^{(0)} = 1/G$.
  3. For each $\mathbf{x}_n$, $n = 1, \ldots, N$, compute the conditional density
  $$f(\mathbf{x}_n | l_{ng} = 1; \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}) = \phi\left(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\Sigma}}_g\right) \quad g = 1, \ldots, G. \tag{3.3}$$
  $\lfloor N\alpha_l \rfloor$% of the samples with lower values of (3.3) are temporarily discarded from contributing to the parameters estimation. The rationale being that observations suffering from either class or attribute noise are implausible under the currently fitted model. That is, $\zeta(\mathbf{x}_n) = 0$ in (3.2) for such observations.
  4. The parameter estimates for the G known classes are updated, based on the non-discarded observations:
  $$\bar{\tau}_g = \frac{\sum_{n=1}^N \zeta(\mathbf{x}_n) l_{ng}}{\lceil N(1-\alpha_l) \rceil} \quad g = 1, \ldots, G \tag{3.4}$$
  $$\bar{\boldsymbol{\mu}}_g = \frac{\sum_{n=1}^N \zeta(\mathbf{x}_n) l_{ng} \mathbf{x}_n}{\sum_{n=1}^N \zeta(\mathbf{x}_n) l_{ng}} \quad g = 1, \ldots, G. \tag{3.5}$$
  Estimation of $\boldsymbol{\Sigma}_g$ depends on the considered patterned model, details are given in Bensmail and Celeux, 1996.
  5. Iterate $3 - 4$ until the $\lfloor N\alpha_l \rfloor$ discarded observations are exactly the same on two consecutive iterations, then stop.

  The procedure described in steps $1 - 5$ is performed `n_init` times, and the parameter estimates $\{\bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ that lead to the highest value of the objective function $\ell_{trim}(\bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}} | \mathbf{X}, \mathbf{l}) = \sum_{n=1}^N \zeta(\mathbf{x}_n) \sum_{g=1}^G \mathbf{l}_{ng} \log\left[\bar{\tau}_g \phi(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\Sigma}}_g)\right]$ are retained. We say that $\{\bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ is the output of the robust initialization phase for the G known classes.

- *Initialization for the H hidden classes*: If $E > G$, starting values for the $H = E - G$ hidden classes need to be properly initialized as follows:

  1. For each hidden class h, $h = G + 1, \ldots, E$, draw a random $(p+1)$-subset J_h and compute its empirical mean $\hat{\boldsymbol{\mu}}_h^{(0)}$ and variance covariance matrix $\hat{\boldsymbol{\Sigma}}_h^{(0)}$ according to

the considered parsimonious structure. Mixing proportions τ_h are drawn from $\mathcal{U}_{[0,1]}$ and initial values set equal to

$$\hat{\tau}_h^{(0)} = \frac{\tau_h}{\sum_{j=G+1}^{E} \tau_j} \times \frac{H}{E}, \ h = G+1, \ldots, E.$$

The previously estimated τ_g should also be renormalized:

$$\hat{\tau}_g^{(0)} = \bar{\tau}_g \times \frac{G}{E}, \ g = 1, \ldots, G.$$

In such a way the initialized vector of mixing proportion sums to 1 over the E groups.

- If the selected patterned model allows for heteroscedastic $\boldsymbol{\Sigma}_g$, and $\hat{\boldsymbol{\Sigma}}_g^{(0)}$, $g = 1, \ldots, E$ do not satisfy the eigenvalues-ratio constraint in (1.20), constrained maximization needs to be enforced. Given the semi-supervised nature of the problem at hand, we propose to further rely on the information that can be extracted from the robustly initialized estimates $\{\bar{\boldsymbol{\tau}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ to set sensible values for the fixed constant $c \geq 1$ required in the eigenvalue-ratio restriction. That is, if no prior information for the value c is available, as it is almost always the case in real applications (García-Escudero et al., 2018b), the following quantity could be, at least initially, used:

$$\tilde{c} = \frac{\max_{g=1\ldots G} \max_{l=1\ldots p} \bar{d}_{lg}}{\min_{g=1\ldots G} \min_{l=1\ldots p} \bar{d}_{lg}} \tag{3.6}$$

with $\bar{d}_{lg}$, $l = 1, \ldots, p$ being the eigenvalues of the matrix $\bar{\boldsymbol{\Sigma}}_g$, $g = 1, \ldots, G$. This implicitly means that we expect extra hidden groups whose difference among group scatters is no larger than that observed for the known groups. Such a choice prevents the appearance of spurious solutions, protecting the adapted learner to wrongly identify random patterns as unobserved classes whilst allowing for groups variability to be preserved. Nevertheless, one might want to allow more flexibility in the group structure and use (3.6) as a lower bound for c, rather than an actual reasonable value. Once having obtained $\hat{\boldsymbol{\Sigma}}_g^{(0)}$ under the eigenvalue ratio constraint, the following EM iterations produce an algorithm that maximizes the observed likelihood in (3.1).

- *EM Iterations:* denote by $\hat{\boldsymbol{\Theta}}^{(k)} = \{\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_E^{(k)}, \hat{\boldsymbol{\mu}}_1^{(k)}, \ldots, \hat{\boldsymbol{\mu}}_E^{(k)}, \hat{\boldsymbol{\Sigma}}_1^{(k)}, \ldots, \hat{\boldsymbol{\Sigma}}_E^{(k)}\}$ the parameter estimates at the k-th iteration of the algorithm.

  - *Step 1 - Concentration*: the trimming procedure is implemented by discarding the $\lfloor N\alpha_l \rfloor$ observations $\mathbf{x}_n$ with smaller values of

$$D\left(\mathbf{x}_n; \hat{\boldsymbol{\Theta}}^{(k)}\right) = \prod_{g=1}^{E} \left[\phi\left(\mathbf{x}_n; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right)\right]^{l_{ng}} \quad n = 1, \ldots, N \tag{3.7}$$

  and discarding the $\lfloor M\alpha_u \rfloor$ observations $\mathbf{y}_m$ with smaller values of

$$D\left(\mathbf{y}_m; \hat{\boldsymbol{\Theta}}^{(k)}\right) = \sum_{g=1}^{E} \hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right) \quad m = 1, \ldots, M. \tag{3.8}$$

Notice that we implicitly set $l_{ng} = 0 \; \forall \, n = 1, \ldots, N$, $g = G+1, \ldots, E$ in (3.7). That is, none of the learning units belong to one of the hidden classes h, $h = G+1, \ldots, E$.

– *Step 2 - Expectation*: for each non-trimmed observation $\mathbf{y}_m$ compute the posterior probabilities

$$\hat{z}_{mg}^{(k+1)} = \frac{\hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right)}{D\left(\mathbf{y}_m; \hat{\boldsymbol{\theta}}^{(k)}\right)} \quad g = 1, \ldots, E; \quad m = 1, \ldots, M. \tag{3.9}$$

– *Step 3 - Constrained Maximization*: the parameter estimates are updated, based on the non-discarded observations and the current estimates for the unknown labels:

$$\hat{\tau}_g^{(k+1)} = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \hat{z}_{mg}^{(k+1)}}{\lceil N(1-\alpha_l) \rceil + \lceil M(1-\alpha_u) \rceil} \quad g = 1, \ldots, E \tag{3.10}$$

$$\hat{\boldsymbol{\mu}}_g^{(k+1)} = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} \mathbf{x}_n + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \hat{z}_{mg}^{(k+1)} \mathbf{y}_m}{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} + \sum_{m=1}^{M} \varphi(\mathbf{y}_m) \hat{z}_{mg}^{(k+1)}} \quad g = 1, \ldots, E. \tag{3.11}$$

Estimation of $\boldsymbol{\Sigma}_g$ depends on the considered patterned model and on the eigenvalues-ratio constraint. Details are given in Bensmail and Celeux, 1996 and, if (1.20) is not satisfied, in Section 2.8.

– *Step 4 - Convergence of the EM algorithm*: if convergence has not been reached (see Section 3.2.4), set $k = k+1$ and repeat steps 1-4.

Notice that, once the hidden classes have been properly initialized, the transductive approach relies on an EM algorithm that makes use of both training and test sets for jointly estimating the parameters of known and hidden classes, with no distinction between the two. The final output from the procedure is a set of parameters $\{\hat{\tau}_g, \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Sigma}}_g\}$, $g = 1, \ldots, E$, and values for the indicator functions $\zeta(\cdot)$ and $\varphi(\cdot)$. Furthermore, the estimated values $\hat{z}_{mg}$ provide a classification for the unlabeled observations $\mathbf{y}_m$ using the MAP rule.
Summing up, the procedure jointly identifies a mislabeled and/or an outlying unit in the training set when $\zeta(\mathbf{x}_n) = 0$, an outlier in the test set when $\varphi(\mathbf{y}_m) = 0$ and an observation in the test belonging to a hidden class whenever $\text{argmax}_{g=1,\ldots,E} \hat{z}_{mg} \in \{G+1, \ldots, E\}$.

3.2.3 Estimation procedure: inductive approach

In contrast with transductive inference, the inductive learning approach aims at solving a broader type of problem: a general model is built from the training set to be ideally applied on any new data point, without the need of a specific test set to be previously defined (Mitchell, 1997; Shaoning Pang and Kasabov, 2004). As a consequence, this approach is most suitable for real-time dynamic classification of data streams, since only the classification rule (i.e., model parameters) is stored and the training set need not be kept in memory. Operationally, inductive learning is performed in two steps: a robust learning phase and a robust discovery phase (see Figure 3.3). In the learning phase, only training observations are considered and we therefore fall into the robust fully-supervised framework for classification. In the robust discovery phase only the parameters for the $E - G$ extra classes need to be estimated, since the parameters obtained in the learning phase are kept fixed throughout this second phase. The entire procedure is detailed in the next Sections.

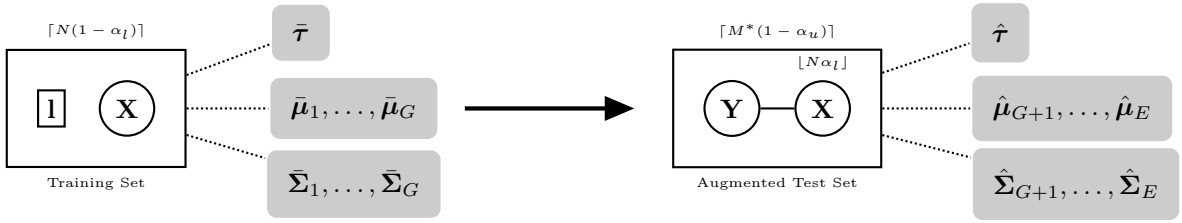

FIGURE 3.3: General framework of the *robust inductive estimation* approach. $\lceil N(1-\alpha_l)\rceil$ observations in the training are used to estimate parameters for the known groups in the Robust Learning Phase. Keeping fixed the estimates obtained in the previous phase, $\lceil M^*(1-\alpha_u)\rceil$ observations in the augmented test are then employed in estimating parameters only for the hidden classes, $M^* = M + \lfloor N\alpha_l \rfloor$.

Robust learning phase

The first phase of the inductive approach consists of estimating parameters for the observed classes using only the training set. That is, we aim at building a robust fully-supervised model considering only the (complete set) of observations $\{\mathbf{x}_n, \mathbf{l}_n\}, n = 1, \ldots, N$. The associated *trimmed log-likelihood* to be maximized with respect to parameters $\{\tau_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\}$, $g = 1, \ldots, G$, reads:

$$\ell_{trim}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{X}, \mathbf{l}) = \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \sum_{g=1}^{G} \mathbf{l}_{ng} \log \Big(\tau_g \phi(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \Big) \tag{3.12}$$

Notice that (3.12) is the first of the two terms that compose (3.1). In this situation, the likelihood in (3.12) is equivalent to the one of the REDDA model introduced in Section 2.2. The estimation procedure coincides with the *Robust Initialization for the G known groups* step in the transductive approach (see Section 3.2.2).

At this point, one could employ the trimmed adaptation of the Bayesian Information Criterion (Neykov et al., 2007; Schwarz, 1978; Fraley and Raftery, 2002) for selecting the best model among the 14 covariance decomposition of Figure 1.3. Notice that the parametrization chosen in the learning phase will influence the available models for the discovery phase (see Figure 3.4).

This concludes the learning phase and the role the training set has in the estimation procedure: from now on $\{\mathbf{x}_n, \mathbf{l}_n\}, n = 1, \ldots, N$ may be discarded. The only exception being the $\lfloor N\alpha_l \rfloor$ units for which $\zeta(\mathbf{x}_n) = 0$: denote such observations with $\{\mathbf{x}_i^*, \mathbf{l}_i^*\}, i = 1, \ldots, \lfloor N\alpha_l \rfloor$. These are the observations not included in the estimation procedure, that is, samples whose conditional density (3.3) is smallest. This could be due to either a wrong label $\mathbf{l}_i^*$ or $\mathbf{x}_i^*$ to be an actual outlier: in the former case, $\mathbf{x}_i^*$ could still be potentially useful for detecting unobserved classes. We therefore propose to join the $\lfloor N\alpha_l \rfloor$ units excluded from the learning phase with the test set to define an *augmented test set* $\mathbf{Y}^* = \mathbf{Y} \cup \mathbf{X}^{(\alpha_l)}$, with elements $\mathbf{y}_m^*$, $m = 1, \ldots, M^*$, $M^* = (M + \lfloor N\alpha_l \rfloor)$, to be employed in the discovery phase. Clearly, $\mathbf{Y}^*$ reduces to $\mathbf{Y}$ if $\alpha_l = 0$. In addition, depending on the real problem at hand, the recovery of the $\mathbf{x}_i^*$ units may be too time consuming or too costly or simply impossible when an online classification is to be performed. In such cases the robust discovery phase described in the next Section can still be applied making use of the original test units $\mathbf{y}_m$, $m = 1, \ldots, M$.

Robust discovery phase

In the robust discovery phase, we search for $H = E - G$ hidden classes robustly estimating the related parameters in an unsupervised way. Particularly, the set $\{\bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\Sigma}}_g\}$, $g = 1, \ldots, G$ will remain fixed throughout the discovery phase and, therefore, the observed *trimmed log-likelihood* given by:

$$\begin{aligned}\ell_{trim}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{Y}^*, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}) = \sum_{m=1}^{M^*} \varphi(\mathbf{y}_m^*) \log \Bigg(\sum_{g=1}^{G} \tau_g \phi(\mathbf{y}_m^*; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\Sigma}}_g) + \\ + \sum_{h=G+1}^{E} \tau_h \phi(\mathbf{y}_m^*; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h) \Bigg)\end{aligned} \tag{3.13}$$

will be maximized with respect to $\{\boldsymbol{\tau}, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h\}$, $h = G+1, \ldots, E$. Notice again that the parameters for the known classes obtained in the robust discovery phase are kept fixed, as indicated by the bar in the notation. Direct maximization of (3.13) is an intractable problem, therefore, we extend Bouveyron, 2014 making again use of an EM algorithm defining a proper *complete trimmed log-likelihood*:

$$\begin{aligned}\ell_{trim_c}(\boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{Y}^*, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}, \mathbf{z}^*) = \sum_{m=1}^{M^*} \varphi(\mathbf{y}_m^*) \Bigg(\sum_{g=1}^{G} z_{mg}^* \log(\tau_g \phi(\mathbf{y}_m^*; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\Sigma}}_g)) + \\ + \sum_{h=G+1}^{E} z_{mh}^* \log(\tau_h \phi(\mathbf{y}_m^*; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)) \Bigg)\end{aligned} \tag{3.14}$$

The following steps delineate the procedure needed for maximizing (3.13):

- *Initialization for the H hidden classes*:

  1. For each hidden class h, $h = G+1, \ldots, E$, draw a random $(p+1)$-subset J_h and compute its empirical mean $\hat{\boldsymbol{\mu}}_h^{(0)}$ and variance covariance matrix $\hat{\boldsymbol{\Sigma}}_h^{(0)}$ according to the considered parsimonious structure. Mixing proportions τ_h are drawn from $\mathcal{U}_{[0,1]}$ and initial values set equal to

  $$\hat{\tau}_h^{(0)} = \frac{\tau_h}{\sum_{j=G+1}^{E} \tau_j} \times \frac{H}{E}, \quad h = G+1, \ldots, E.$$

  The τ_g estimated in the robust learning phase should also be renormalized:

  $$\hat{\tau}_g^{(0)} = \bar{\tau}_g \times \frac{G}{E}, \quad g = 1, \ldots, G.$$

- If the selected patterned model allows for heteroscedastic $\boldsymbol{\Sigma}_g$, and $\hat{\boldsymbol{\Sigma}}_g^{(0)}$, $g = G+1, \ldots, E$ do not satisfy (1.20), constrained maximization needs to be enforced, given the unsupervised nature of the problem, as spurious solutions are likely to appear during the estimation procedure. Notice that, thanks to the inductive approach, only the estimates for the H hidden groups covariance matrices need to satisfy the eigen-ratio constraint. Similarly to what done for the transductive approach, we propose to further rely on the information that can be extracted from the robust learning phase to set a lower bound for the fixed constant $c \geq 1$ required in the eigenvalue-ratio restriction. Particularly, the quantity in (3.6) can be used, therefore implicitly assuming that the hidden groups variability is no

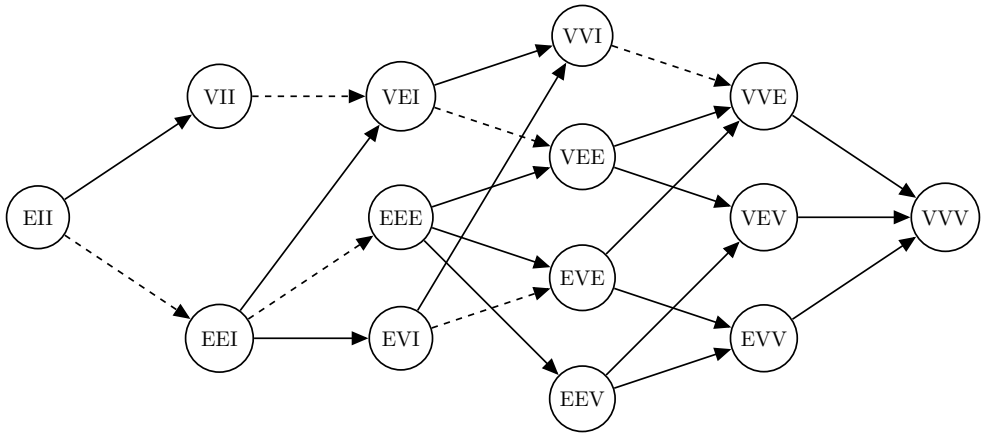

FIGURE 3.4: Partial-order structure in the eigen-decomposition for the covariance matrices of Banfield and Raftery, 1993 and Celeux and Govaert, 1995. Model complexity increases from left to right. Dashed arrows denote equivalent models in terms of parameters to be estimated in the Discovery Phase.

larger than that observed for the known groups. In this way, estimates are protected from the appearance of spurious solutions that can easily arise when searching for unobserved classes also in the simplest scenarios (see Section 3.2.6).

Once initial values satisfying the constraint in (1.20) have been properly determined for the parameters of the hidden classes, the following EM iterations produce an algorithm that maximizes (3.13).

- *EM Iterations:* denote by $\hat{\boldsymbol{\Theta}}^{(k)} = \{\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_E^{(k)}, \hat{\boldsymbol{\mu}}_{G+1}^{(k)}, \ldots, \hat{\boldsymbol{\mu}}_E^{(k)}, \hat{\boldsymbol{\Sigma}}_{G+1}^{(k)}, \ldots, \hat{\boldsymbol{\Sigma}}_E^{(k)}\}$ the parameter estimates at the k-th iteration of the algorithm.

  - *Step 1 - Concentration*: Define

$$D_g\left(\mathbf{y}_m^*; \hat{\boldsymbol{\Theta}}^{(k)}\right) = \begin{cases} \hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m^*; \bar{\boldsymbol{\mu}}_g, \bar{\boldsymbol{\Sigma}}_g\right) & g = 1, \ldots, G \\ \hat{\tau}_g^{(k)} \phi\left(\mathbf{y}_m^*; \hat{\boldsymbol{\mu}}_g^{(k)}, \hat{\boldsymbol{\Sigma}}_g^{(k)}\right) & g = G+1, \ldots, E \end{cases}$$

    The trimming procedure is implemented by discarding the $\lfloor M^* \alpha_u \rfloor$ observations $\mathbf{y}_m^*$ with smaller values of

$$D\left(\mathbf{y}_m^*; \hat{\boldsymbol{\Theta}}^{(k)}\right) = \sum_{g=1}^{E} D_g\left(\mathbf{y}_m^*; \hat{\boldsymbol{\Theta}}^{(k)}\right) \quad m = 1, \ldots, M^*.$$

  - *Step 2 - Expectation*: for each non-trimmed observation $\mathbf{y}_m^*$ compute the posterior probabilities

$$\hat{z}_{mg}^{*(k+1)} = \frac{D_g\left(\mathbf{y}_m^*; \hat{\boldsymbol{\Theta}}^{(k)}\right)}{D\left(\mathbf{y}_m^*; \hat{\boldsymbol{\Theta}}^{(k)}\right)} \quad g = 1, \ldots, E; \quad m = 1, \ldots, M^*.$$

  - *Step 3 - Constrained Maximization*: the parameter estimates are updated, based on the non-discarded observations and the current estimates for the unknown labels. Due

to the constraint $\left(\sum_{g=1}^{G}\tau_g+\sum_{h=G+1}^{E}\tau_h\right)=1$, the mixing proportions are updated as follows using the Lagrange multipliers method as per Section 2.6:

$$\hat{\tau}_g^{(k+1)}=\begin{cases}\bar{\tau}_g\left(1-\sum_{h=G+1}^{E}\frac{\sum_{m=1}^{M}\varphi(\mathbf{y}_m^*)\hat{z}_{mh}^{*(k+1)}}{\lceil M^*(1-\alpha_u)\rceil}\right) & g=1,\ldots,G\\ \frac{\sum_{m=1}^{M}\varphi(\mathbf{y}_m^*)\hat{z}_{mg}^{*(k+1)}}{\lceil M^*(1-\alpha_u)\rceil} & g=G+1,\ldots,E\end{cases}$$

where the proportions for the G known classes computed in the learning phase are renormalized according to the proportions of the H new groups. The estimate update for the mean vectors of the hidden classes reads:

$$\hat{\boldsymbol{\mu}}_h^{(k+1)}=\frac{\sum_{m=1}^{M^*}\varphi(\mathbf{y}_m^*)\hat{z}_{mh}^{*(k+1)}\mathbf{y}_m^*}{\sum_{m=1}^{M^*}\varphi(\mathbf{y}_m^*)\hat{z}_{mh}^{*(k+1)}}\quad h=G+1,\ldots,E.$$

Estimation of $\boldsymbol{\Sigma}_h$, $h=G+1,\ldots,E$ depends on the selected patterned model conditioning on the one estimated in the learning phase. More specifically, the parsimonious Gaussian models of Banfield and Raftery, 1993 and Celeux and Govaert, 1995 define a partial order structure in terms of model complexity, graphically reported in Figure 3.4. Such structure allows for constraints relaxation when estimating the covariance matrices for the hidden classes H, moving from left to right in the graph of Figure 3.4. A simple example will clarify the procedure. Imagine to have selected a VEE model in the Learning Phase: $\bar{\boldsymbol{\Sigma}}_g=\bar{\lambda}_g\bar{\mathbf{D}}\bar{\mathbf{A}}\bar{\mathbf{D}}'$, $g=1,\ldots,G$. Due to the Inductive approach, only VEE, VVE, VEV and VVV models can be selected in the Discovery Phase since the first G covariance matrices need to be kept fixed, and their volume is already free to vary across components. If for instance we employ a VEV model (i.e., equal shape across components) in the discovery phase, the estimates for $\boldsymbol{\Sigma}_h$, $h=G+1,\ldots,E$ will be:

$$\hat{\boldsymbol{\Sigma}}_h^{(k+1)}=\hat{\lambda}_h^{(k+1)}\hat{\mathbf{D}}_h^{(k+1)}\bar{\mathbf{A}}\hat{\mathbf{D}}_h^{(k+1)'}\quad h=G+1,\ldots,E$$

where the estimate for the shape $\bar{\boldsymbol{A}}$ comes from the learning phase. Closed form solutions are obtained for all 14 models of Celeux and Govaert, 1995, no matter the parsimonious structure selected in the Learning Phase: details are reported in Appendix A (Section 3.6). Notice further that whenever the model in the discovery phase is EII, EEI or EEE, no extra parameters need to be estimated for the covariance matrices of the hidden groups. Finally, if (1.20) is not satisfied for the covariance matrices in the new classes, the eigenvalue restriction needs to be enforced: see Section 2.8.

– *Step 4 - Convergence of the EM algorithm*: check for algorithm convergence (see Section 3.2.4). If convergence has not been reached, set $k=k+1$ and repeat steps 1-4.

Notice that the EM algorithm is solely based on the augmented test units for estimating parameters of the hidden classes. That is, if $E=G$ no extra parameters will be estimated in the discovery phase and the inductive approach will reduce to a fully-supervised classification method.
The final output from the learning phase is a set of parameters $\{\bar{\tau}_g,\bar{\boldsymbol{\mu}}_g,\bar{\boldsymbol{\Sigma}}_g\}$, $g=1,\ldots,G$ for the known classes and values for the indicator function $\zeta(\cdot)$ where $\zeta(\mathbf{x}_n)=0$ identify $\mathbf{x}_n$ as an outlying observation. The final output from the discovery phase is a set of parameters $\{\hat{\boldsymbol{\mu}}_h,\hat{\boldsymbol{\Sigma}}_h\}$,

$h = G+1, \ldots, E$, for the hidden classes together with an update for the mixing proportion $\hat{\tau}_g$, $g = 1, \ldots, E$ and values for the indicator functions $\varphi(\cdot)$ where $\varphi(\mathbf{y}_m^*) = 0$ identify $\mathbf{y}_m^*$ as an outlying observations. Likewise for the transductive approach, the estimated values $\hat{z}_{mg}^*$ provide a classification for the unlabeled observations $\mathbf{y}_m^*$, assigning them to one of the known or hidden classes.
R (R Core Team, 2018) source code implementing the EM algorithms under the transductive and inductive approaches is available at https://github.com/AndreaCappozzo/raedda. A dedicated R package is currently under development.

3.2.4 Convergence criterion

The convergence for both transductive and inductive approaches is assessed via the Aitken acceleration (Aitken, 1926; McNicholas et al., 2010):

$$a^{(k)} = \frac{\ell_{trim}^{(k+1)} - \ell_{trim}^{(k)}}{\ell_{trim}^{(k)} - \ell_{trim}^{(k-1)}} \tag{3.15}$$

where $\ell_{trim}^{(k)}$ is the trimmed observed data log-likelihood from iteration k: equation (3.1) and (3.13) for the transductive and the inductive approach, respectively.
The asymptotic estimate of the trimmed log-likelihood at iteration k is given by (Böhning et al., 1994):

$$\ell_{\infty_{trim}}^{(k)} = \ell_{trim}^{(k)} + \frac{1}{1 - a^{(k)}} \left(\ell_{trim}^{(k+1)} - \ell_{trim}^{(k)} \right). \tag{3.16}$$

The EM algorithm is considered to have converged when $|\ell_{\infty_{trim}}^{(k)} - \ell_{trim}^{(k)}| < \varepsilon$; a value of $\varepsilon = 10^{-5}$ has been chosen for the experiments reported in the next Sections.

3.2.5 Model selection: determining the covariance structure and the number of components

A robust likelihood-based criterion is employed for choosing the number of hidden classes, the best model among the 14 patterned covariance structures depicted in Figure 1.3 and a reasonable value for the constraint c in (1.20). Particularly, in our context, the problem of estimating the number of hidden classes corresponds to setting the number of components in a finite Gaussian mixture model (see for example Mclachlan and Rathnayake, 2014 for a discussion on the topic). The general form of the robust information criterion is:

$$RBIC = 2\ell_{trim}(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) - v_{XXX}^{c} \log(n^*) \tag{3.17}$$

where $\ell_{trim}(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ denotes the maximized trimmed observed data log-likelihood under either the transductive or inductive approach: equation (2.1) and (3.13), respectively. The total number of observations n^* employed in the estimation procedure is:

$$n^* = \begin{cases} \lceil N(1-\alpha_l) \rceil + \lceil M(1-\alpha_u) \rceil & \text{Transductive EM} \\ \lceil M^*(1-\alpha_u) \rceil & \text{Inductive EM (discovery phase).} \end{cases}$$

In (3.17), the penalty term v_{XXX}^{c} accounts for the number of parameters to be estimated. It depends on the estimation procedure (either transductive or inductive), the chosen patterned

TABLE 3.1: Nomenclature and number of free parameters to be estimated for the variance covariance matrices, under the 14 patterned structures of Banfield and Raftery, 1993 and Celeux and Govaert, 1995. γ denotes the number of parameters related to the orthogonal rotation and δ the number of parameters related to the eigenvalues, for both transductive and inductive approach (discovery phase). The last column indicates whether the eigenvalue-ratio (ER) constraint is required. The learning phase of the inductive approach possesses the number of parameters indicated for the transductive approach, with E replaced by G.

Model	$\gamma_{transductive}$	$\delta_{transductive}$	$\gamma_{inductive}$	$\delta_{inductive}$	ER
EII	0	1	0	0	Not Required
VII	0	E	0	H	Required
EEI	0	p	0	0	Not Required
VEI	0	$E+p-1$	0	H	Required
EVI	0	$Ep-(E-1)$	0	$Hp-H$	Required
VVI	0	Ep	0	Hp	Required
EEE	$p(p-1)/2$	p	0	0	Not Required
VEE	$p(p-1)/2$	$E+p-1$	0	H	Required
EVE	$p(p-1)/2$	$Ep-(E-1)$	0	$Hp-H$	Required
EEV	$Ep(p-1)/2$	p	$Hp(p-1)/2$	0	Not Required
VVE	$p(p-1)/2$	Ep	0	Hp	Required
VEV	$Ep(p-1)/2$	$E+p-1$	$Hp(p-1)/2$	H	Required
EVV	$Ep(p-1)/2$	$Ep-(E-1)$	$Hp(p-1)/2$	$Hp-H$	Required
VVV	$Ep(p-1)/2$	Ep	$Hp(p-1)/2$	Hp	Required

covariance structure and the value for the constraint c:

$$v^c_{XXX} = \kappa + \gamma + (\delta - 1)\left(1 - \frac{1}{c}\right) + 1. \tag{3.18}$$

κ is the number of parameters related to mixing proportions and mean vectors: $\kappa = Ep + (E-1)$ in the transductive setting and $\kappa = Hp + H$ for the discovery phase in the inductive approach. γ and δ denote, respectively, the number of free parameters related to the orthogonal rotation and to the eigenvalues for the estimated covariance matrices. Their values, for the two approaches and the different patterned structure, are reported in Table 3.1.
The robust information criterion in (3.17) is an adaptation of the complexity-penalized likelihood approach introduced in Cerioli et al., 2018a that here also accounts for the trimming levels and patterned structures. Note that, when $c \to +\infty$ and $\alpha_l = \alpha_u = 0$, (3.17) reduces to the well-known Bayesian Information Criterion (Schwarz, 1978).

3.2.6 On the role of the eigenvalue restrictions

As mentioned in Section 1.3.2, when employing mixture models for supervised learning and discriminant analysis there is actually no need in worrying about the appearance of spurious solutions, since the joint distribution of both observations and associated labels is directly available. The parameters estimation therefore reduces to estimate the within class mean vector and covariance matrix, without the need of any EM algorithm (Fraley and Raftery, 2002). Nonetheless, adaptive learning is based on a partially unsupervised estimation, since hidden classes are sought in the test set without previous knowledge of their group structure extracted from the labelled set. Therefore, efficiently dealing with the possible appearance of spurious solutions

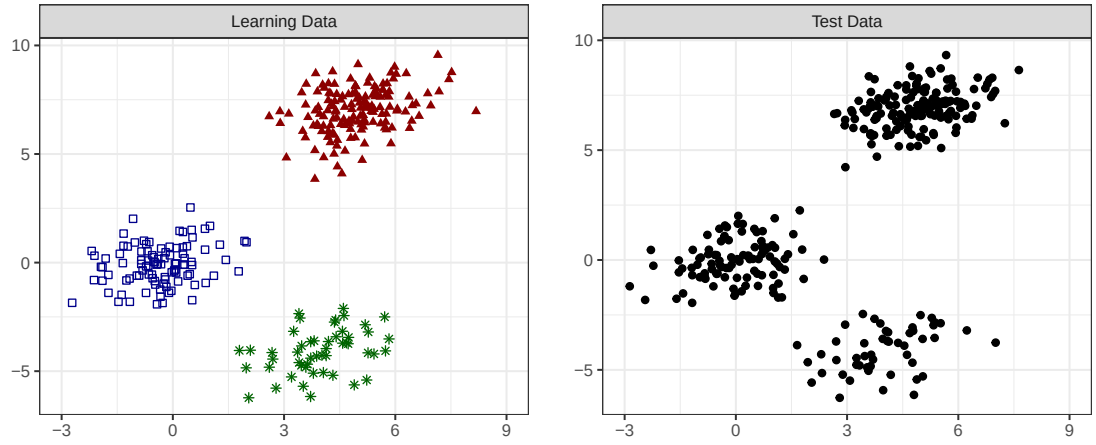

FIGURE 3.5: Original learning problem, with a set of $N = 300$ labelled observations and $M = 300$ unlabeled observations generated from the same mixture of bivariate normal distributions with three components.

becomes fundamental in our context, where the identification of a hidden class might just be the consequence of a spurious solution. As thoroughly reported in the previous Sections, we protect our estimates from the appearance of spurious solutions employing a constrained version of the EM algorithm, by means of the truncation operator defined in Section 1.3.2 and the novel computational procedures introduced in Section 2.8.
We now provide an illustrative example for underlying the importance of protecting the adaptive learner from spurious solutions, that may arise also in the simplest scenarios. Consider a data generating process given by a three components mixture of bivariate normal distributions ($E = G = 3$ and $p = 2$) with the following parameters:

$$\boldsymbol{\tau} = (0.35, 0.15, 0.5)', \quad \boldsymbol{\mu}_1 = (0, 0)', \quad \boldsymbol{\mu}_2 = (4, -4)', \quad \boldsymbol{\mu}_3 = (5, 7)'$$

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

Figure 3.5 graphically presents the learning problem, in which both the training and test sets contain 300 data points. Clearly, even from a visual exploration, the test set does not contain any hidden group and we therefore expect that the model selection criterion defined in Section 3.2.5 will choose a mixture of $E = 3$ components as the best model for the problem at hand. Employing transductive estimation, the RAEDDA model is fitted to the data, with trimming levels set to 0 for both labelled and unlabeled sets ($\alpha_l = \alpha_u = 0$) and considering two different values for the eigen-ratio constraint: $c = 10$ in the first case and $c = 10^{10}$ in the second. That is, we set a not too restrictive constraint in the former model (notice that the true ratio between the biggest and smallest eigenvalues of $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, 3$ is equal to 1.86) and we consider a virtually unconstrained estimation for the latter. The classification obtained for the best model in the test set, selected via the robust information criterion in (3.17), under the two different scenarios is reported in Figure 3.6. The value for the maximized log-likelihood in the first scenario is equal to -2257.279, and it is equal to -2186.615 in the unconstrained case. With only 2 data points in the hidden group and $|\hat{\boldsymbol{\Sigma}}_4| < 10^{-10}$ we are clearly dealing with a spurious solution and not with a hidden class. Nonetheless, the appearance of spurious maxima even in this simple toy experiment casts light on how paramount it is to protect the estimates against this harmful possibility.

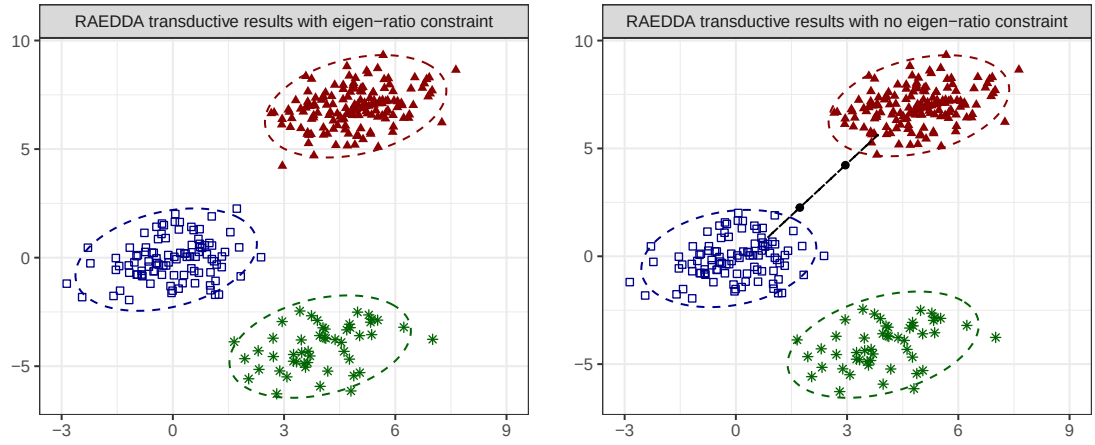

FIGURE 3.6: The classification obtained for the best model in the test set, with two different values for the eigen-ratio constraint. In the unconstrained case the classification is based on a spurious solution, with a localized random pattern wrongly identified as a hidden class.

3.2.7 Further aspects

Notice that the RAEDDA methodology is a generalization of several model-based classification methods, in particular:

- EDDA (Bensmail and Celeux, 1996) when only fitting the robust learning phase with $\alpha_l = 0$.
- REDDA (Chapter 2 of the present manuscript) when only fitting the robust learning phase with $\alpha_l > 0$.
- UPCLASS (Dean et al., 2006) when fitting the transductive approach with $E = G$ and $\alpha_l = 0, \alpha_u = 0$
- RUPCLASS (Chapter 2 of the present manuscript) when fitting the transductive approach with $E = G$ and $\alpha_l > 0, \alpha_u > 0$.
- AMDA transductive (Bouveyron, 2014) when fitting the transductive approach with $E \geq G$ and $\alpha_l = 0, \alpha_u = 0$. Notice in addition that RAEDDA considers a broader class of learners employing patterned covariance structures.
- AMDA inductive (Bouveyron, 2014) when fitting the inductive approach with $\alpha_l = 0$, $\alpha_u = 0$. Also here the class of considered models is larger, thanks to the partial-order structure in the eigen-decomposition of the covariance matrices (see Figure 3.4).

3.3 Simulation study

In this Section, we present a simulation study in which the performance of novelty detection methods is assessed when dealing with different combinations of data generating processes and contamination rates. For each scenario, an entire class is not present in the labelled units, and it thus needs to be discovered by the adaptive classifiers in the test set. The problem definition is therefore as follows: we aim at judging the performance of various methods in recovering the true partition under a semi-supervised framework, where the groups distribution

is (approximately) Gaussian, allowing for a distribution-free noise structure, both in terms of label noise and outliers.

3.3.1 Experimental setup

The $E = 3$ classes are generated via multivariate normal distributions of dimension $p = 6$ with the following parameters:

$$\boldsymbol{\mu}_1 = (0,8,0,0,0,0)', \quad \boldsymbol{\mu}_2 = (8,0,0,0,0,0)', \quad \boldsymbol{\mu}_3 = (-8,-8,0,0,0,0)'$$

$$\boldsymbol{\Sigma}_1 = \text{diag}(1,a,1,1,1,1), \quad \boldsymbol{\Sigma}_2 = \text{diag}(b,c,1,1,1,1), \quad \boldsymbol{\Sigma}_3 = \left(\begin{array}{cc|c} d & e & \\ e & f & \mathbf{0} \\ \hline & \mathbf{0} & \boldsymbol{I} \end{array}\right)$$

We consider 5 different combinations of (a,b,c,d,e,f):

- $(a,b,c,d,e,f) = (1,1,1,1,0,1)$, spherical groups with equal volumes (EII)
- $(a,b,c,d,e,f) = (5,1,5,1,0,5)$, diagonal groups with equal covariance matrices (EEI)
- $(a,b,c,d,e,f) = (5,5,1,3,-2,3)$, groups with equal volume, but varying shapes and orientations (EVV)
- $(a,b,c,d,e,f) = (1,20,5,15,-10,15)$, groups with different volumes, shapes and orientations (VVV)
- $(a,b,c,d,e,f) = (1,45,30,15,-10,15)$, groups with different volumes, shapes and orientations (VVV) but with two severe overlap

The afore-described data generating process has been introduced in García-Escudero et al., 2008: we adopt it here as it elegantly provides a well-defined set of resulting parsimonious covariance structures. In addition, two different group proportions are included:

- `equal`: $N_1 = N_2 = 285$ and $M_1 = M_2 = M_3 = 360$
- `unequal`: $N_1 = 190$, $N_2 = 380$ and $M_1 = 216, M_2 = M_3 = 432$

where N_g, $g = 1,2$ and M_h, $h = 1,2,3$ denote the sample sizes for each group in the training and test sets, respectively. According to the notation introduced in Section 1.3.4, we observe $G = 2$ classes in the training and $H = 1$ extra class in the test set. Furthermore, we apply contamination adding both attribute and class noise as follows. A fixed number Q_l and Q_u of uniformly distributed outliers, having squared Mahalanobis distances from $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_3$ greater than $\chi^2_{6,0.975}$, are respectively added to the labelled and unlabeled sets. Additionally, we assign a wrong label to Q_l genuine units, randomly chosen in the training set. Four different contamination levels are considered, varying Q_l and Q_u:

- No contamination: $Q_l = Q_u = 0$,
- Low contamination: $Q_l = 10$ and $Q_u = 40$,
- Medium contamination: $Q_l = 20$ and $Q_u = 80$,
- Strong contamination: $Q_l = 30$ and $Q_u = 120$.

A total of $B = 1000$ Monte Carlo replications are generated for each combination of covariance structure, groups proportion and contamination rate. Results for the considered scenarios are reported in the next Section.

3.3.2 Simulation results

Given the simulation framework presented in the previous Section, we compare the performance of RAEDDA against the original AMDA model (Bouveyron, 2014) and two popular novelty detection methods, namely Classifier Instability (Tax and Duin, 1998) and Support Vector Method for novelty detection (Schölkopf et al., 2000), respectively denoted as QDA-ND and SVM-ND hereafter. For assessing the performance in terms of classification accuracy, outliers detection and hidden groups discovery for the competing methods, a set of 4 metrics is recorded at each replication of the simulation study:

- % Label Noise: the proportion of Q_l mislabeled units in the training set correctly identified as such by the RAEDDA model (for which the final value of the trimming function $\zeta(\cdot)$ is equal to 0);
- % Hidden Group: the proportion of units in the test set belonging to the third group correctly assigned to a previously unseen class by AMDA and RAEDDA methods;
- ARI: Adjusted Rand Index (Rand, 1971), measuring the similarity between the partition returned by a given method and the underlying true structure;
- % Novelty: the proportion of units in the test set belonging either to the third group or to the set of Q_u outliers correctly identified by the novelty detection methods.

Box plots for the four metrics, resulting from the B = 1000 Monte Carlo repetitions under different covariance structure, groups proportion and contamination rate are reported in Figure 3.7 and 3.8. The "% Label Noise" metric highlights the effectiveness of our proposal in correctly identifying the Q_l wrongly labelled units in the training set, thus protecting the parameter estimates from bias. Both transductive and inductive approaches perform well regardless of the contamination rate; the number of correctly detected mislabeled units however slightly decreases under the VVV and VVV with overlap simulation scenarios. This is nonetheless due to the more complex covariance structure and to the presence of overlapping groups: this makes the identification of label noise more difficult and less crucial for obtaining reliable inference. The "% of Hidden group" metric in Figure 3.7 shows remarkably good performance in detecting the third unobserved class for the adaptive Discriminant Analysis methods, both for AMDA and its robust generalization RAEDDA. Careful investigation of this peculiar result revealed that the AMDA method tended to merge outlying units and the third (unobserved) class in one single extra group. That is, even though the AMDA method correctly discovers the presence of an extra class, the associated parameter estimates are completely spoiled by the presence of outliers. Furthermore, the same result does not hold in the two most complex scenarios, where the negative effect of attribute and class noise strongly undermines the adaptive effectiveness of the AMDA model, especially when the transductive estimation is performed. The "ARI" metric in Figure 3.8 highlights the predictive power of the RAEDDA model: by means of the MAP rule and impartial trimming the true partition of the test set, that jointly includes known groups, one extra class and Q_u outlying units, is efficiently recovered. As previously mentioned, AMDA fails in separating the uniform noise from the extra Gaussian class, with consequent lower values for the ARI metric. Lastly, the "% Novelty" metric serves the purpose of extending the comparison from the two adaptive models to the novelty detection methods, stemming from the machine learning literature. Particularly, the latter class of algorithms only distinguishes the known patterns (i.e., the first two groups in the training set) and the novelty: in our case the hidden class and the uniform noise. It is evident that, as soon as few noisy data points are added to the training set, both novelty detection methods completely

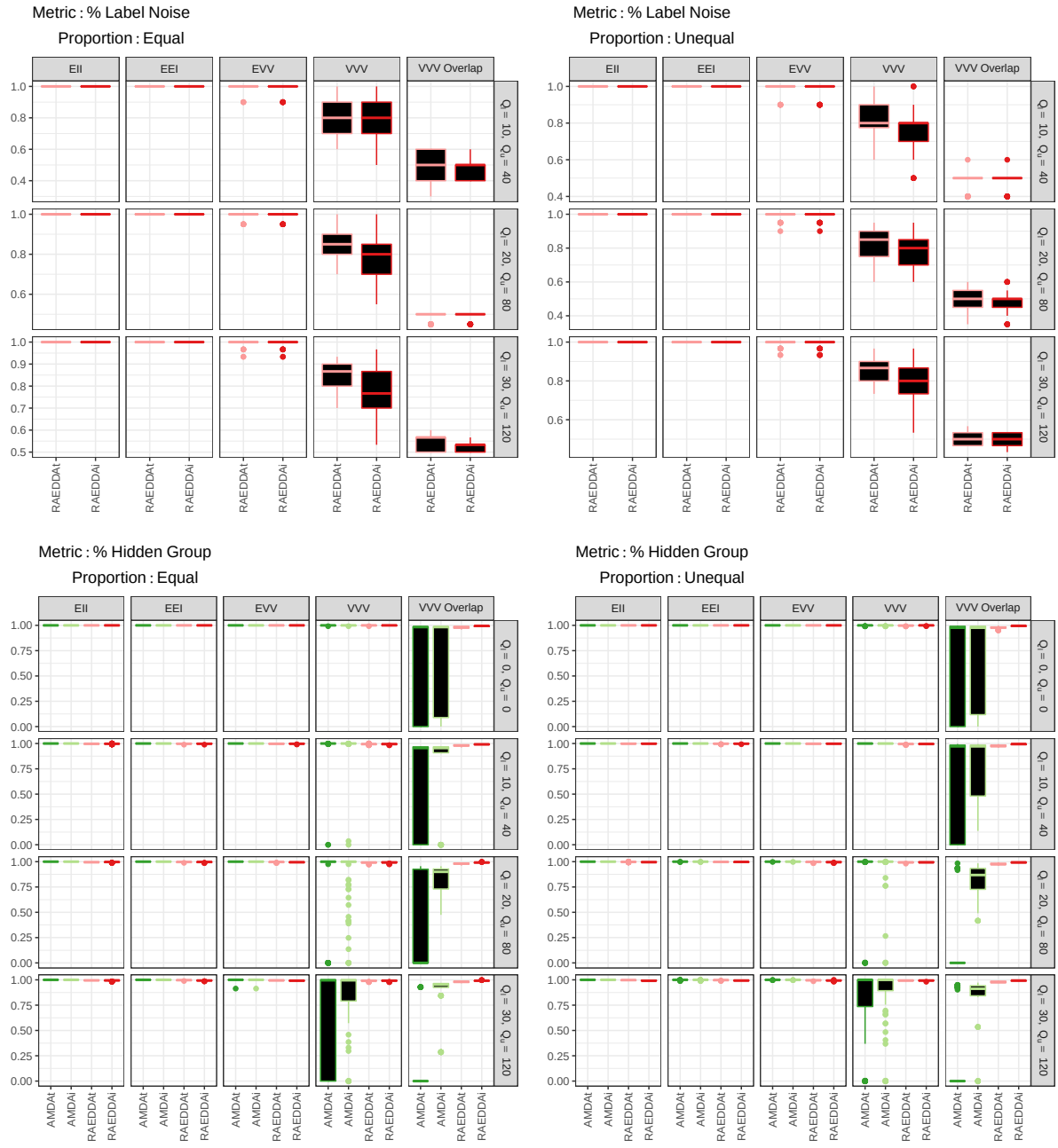

FIGURE 3.7: Box plots for % Label Noise and % Hidden Group metrics for $B = 1000$ Monte Carlo repetitions under different covariance structure, groups proportion and contamination rate.

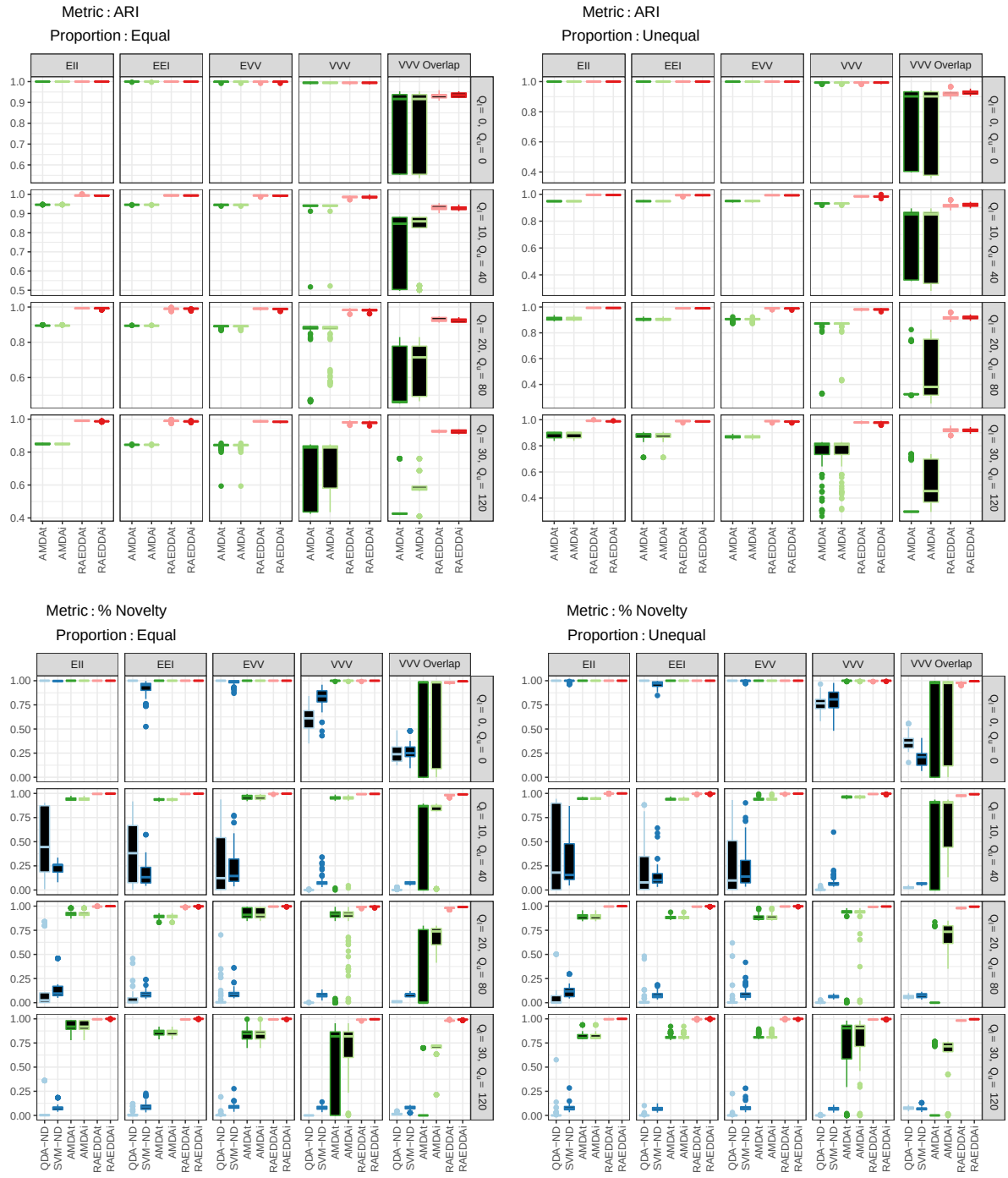

FIGURE 3.8: Box plots for ARI and % Novelty metrics for B = 1000 Monte Carlo repetitions under different covariance structure, groups proportion and contamination rate.

fail in separating known and novel patterns. In addition, the QDA-ND and SVM-ND performances deteriorate when more complex covariance structures are considered, even under the outlier-free scenarios.
Notice that the model selection criterion for the RAEDDA method defined in Section 3.2.5 was used for identifying not only the number of components but also the parsimonious covariance structure: this always yielded to choose the true parametrization according to the values of (a, b, c, d, e, f). As a last worthy note, the simulation study was performed employing the rationale defined in (3.6) for setting c in the eigenvalue-ratio restriction, whilst the impartial trimming levels α_l and α_u were set high enough to account for the presence of both label noise and outliers. The correct tuning for the model hyper-parameters remains a critical challenge, especially for the trimming levels: a promising idea was recently proposed by Cerioli et al., 2019, however, further research in the robust classification framework is still to be pursued.

3.4 Application to grapevine microbiome analysis

In recent years, the tremendous advancements in metagenomics have brought to statisticians a whole new set of questions to be addressed with dedicated methodologies, fostering the fast development of research literature in this field (Waldron, 2018; Calle, 2019). In particular, the role of plant microbiota in grapevine cultivar (*Vitis vinifera L.*) is notably relevant since it has been proven to act as discriminating signature for grape origin (Bokulich et al., 2014; Mezzasalma et al., 2017). Therefore, the employment of microbiome analysis for automatically identifying wine provenance is a promising approach in the food authenticity domain.
A flexible method that performs online classification of grapevine samples, discriminating potentially fraudulent units from known or previously unseen regions is likely to have a great impact on the field.
Motivated by a dataset of microbiome composition of grape samples, we validate the performance of the method introduced in Section 3.2 under different contamination and dataset shifts scenarios.

3.4.1 Data

The considered dataset reports microbiome composition of 45 grape samples collected in 3 different regions having similar pedological features. The first sampling site was the Lombardy Regional Collection in Northern Italy (hereafter NI); the second site was the germplasm collection of E. Mach Foundation in the Trento province, at the foot of the Italian Alps (AI); while the third group of grapes comes from the Government of La Rioja collection, located in Northern Spain (NS). The processes of DNA extraction, sequencing and numbering of microbial composition are thoroughly described in Mezzasalma et al., 2018: we refer the reader interested in the bioinformatics details to consult that paper and references therein.
At the end of sample preparation, the resulting dataset consists of an abundance table with $p = 836$ features (bacterial communities) defined as Operational Taxonomic Unit (OTU): collapsed clusters of similar DNA sequences that describe the total microbial diversity. For each site, 15 observations are available: a graphical representation of the count table, collapsed at OTU level for ease of visualization, is reported in Figure 3.9.

3.4.2 Dimension reduction

Given the high-dimensional nature of the considered dataset ($p = 836$) and the small sample size, a preprocessing step for reducing the dimensionality is paramount before fitting the RAEDDA model. Focusing on the counting nature of the observations at hand, a natural choice

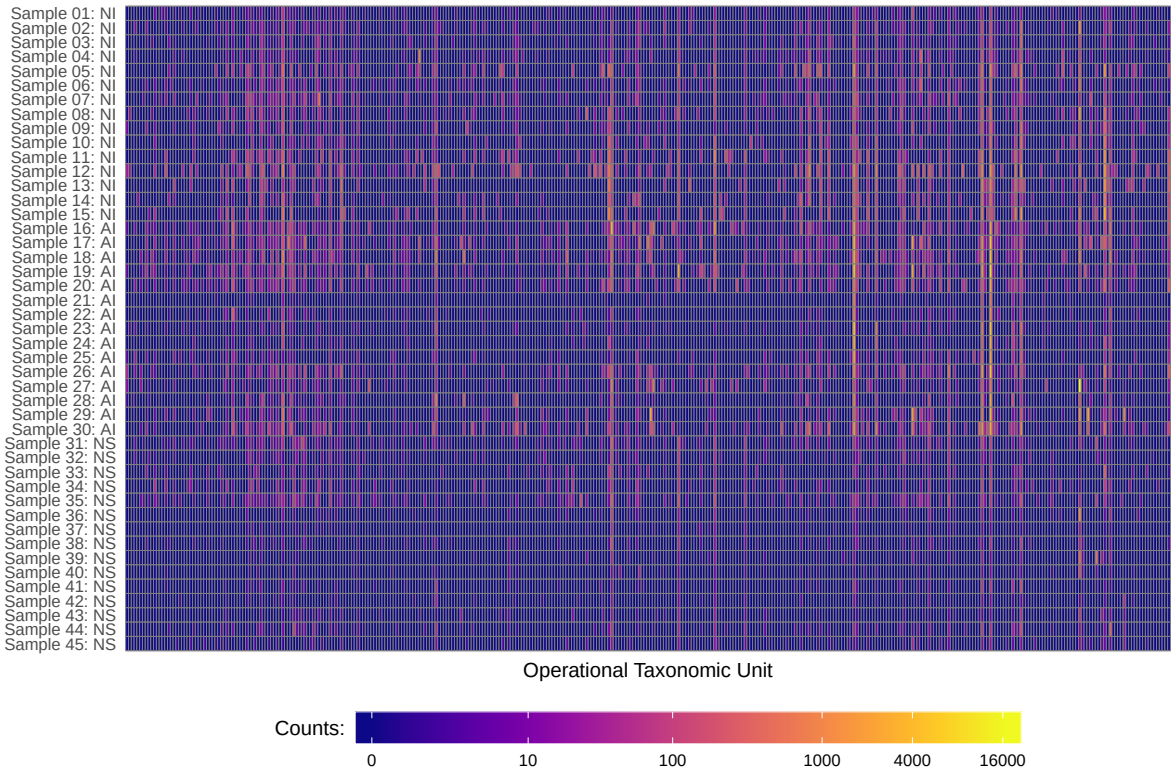

FIGURE 3.9: Count table depicting the abundance and distribution of the OTUs resulting from the sequence analysis for each sample in the 3 different regions: Northern Italy (NI), Italian Alps (AI) and Northern Spain (NS). Grapevine microbiome data.

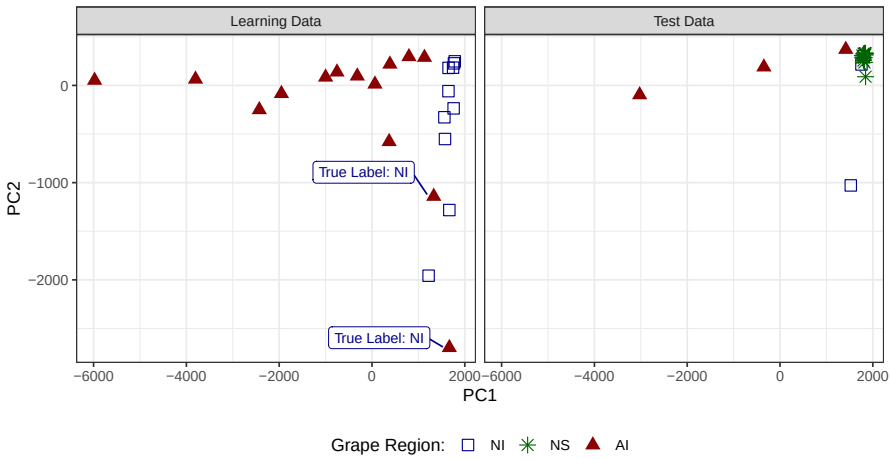

FIGURE 3.10: Learning scenario for anomaly and novelty detection of the grapevine microbiome data on the ROBPCA subspace: 1 unobserved region and label noise.

would be to perform probabilistic Poisson PCA (PLNPCA): a flexible methodology based on the Poisson Lognormal model recently introduced in the literature (Chiquet et al., 2018). Nevertheless, the variational approximation employed for PLNPCA inference makes its generalization from training to test set not so straightforward, and, furthermore, the whole procedure is not robust to outlying observations. Therefore, given the classification framework in which the preprocessing step needs to be embedded, a less domain-specific, yet robust and well-established technique was preferred for dimension reduction.
The considered preprocessing step proceeds as follows: we fit Robust Principal Component Analysis (ROBPCA) to the labelled set, and afterwards we project the test units to the obtained subspace; please refer to Hubert et al., 2005 for a detailed description of the employed methodology. In this way, robust and test-independent (i.e., suitable for either transductive or inductive inference) low-dimensional scores are available for adaptive classification.

3.4.3 Anomaly and novelty detection: label noise and one unobserved class

The first experiment involves the random selection of 24 learning units from the NI and AI regions, with a consequent test set of 21 samples including all the 15 grapes collected in Northern Spain (NS). Furthermore, 2 of the NI units in the learning set are incorrectly labelled as grapes coming from the AI site. The aim of the experiment is therefore to determine whether the RAEDDA method is capable of recovering the unobserved NS class whilst identifying the label noise in the training set. The preprocessing step described in Section 3.4.2 is applied prior to perform classification: standard setting for the `PcaHubert` function in the `rrcov` R package (Todorov and Filzmoser, 2009) retains $d = 2$ robustly estimated principal components, a graphical representation of the learning scenario is reported in Figure 3.10. A RAEDDA model is then employed for building a classification rule, considering both a transductive and an inductive approach. The robust information criterion in (3.17) is used for selecting the best patterned structure and, more importantly, the number of extra classes. RBIC values for the two estimation procedures are reported in Tables 3.2 and 3.3. We restrict the attention to the subset of diagonal models in line with our preprocessing step. Notice that, in the inductive approach, once the VVI model is selected in the learning phase, only the most flexible diagonal model needs to be fitted to the test data, thanks to the partial order structure of Figure 3.4. Our findings show that the robust information criterion correctly detects the true number of classes

TABLE 3.2: RBIC for different patterned structures and number of hidden classes for the RAEDDA model, transductive inference. The model with the highest RBIC value is highlighted in bold. Grapevine microbiome data with one unobserved class (NS).

# Classes	Covariance Structure					
	EII	VII	EEI	VEI	EVI	**VVI**
2	-1278.25	-1204.55	-1279.60	-1208.11	-1221.39	-1175.25
3	-1289.24	-1240.30	-1291.21	-1242.67	-1241.95	**-1148.50**
4	-1300.23	-1254.60	-1302.20	-1257.00	-1256.57	-1163.34

TABLE 3.3: RBIC for different patterned structures and number of hidden classes for the RAEDDA model, inductive inference. The models with the highest RBIC value are highlighted in bold. Grapevine microbiome data with one unobserved class (NS).

Robust learning phase

# Classes	Covariance Structure					
	EII	VII	EEI	VEI	EVI	**VVI**
2	-719.26	-709.13	-718.97	-712.11	-688.40	**-678.29**

Robust discovery phase

# Classes	Covariance Structure
	VVI
2	-639.85
3	**-506.59**
4	-511.43

$E = 3$, in both inferential approaches. Regarding anomaly detection, the two units affected by label noise are identified and a posteriori classified as coming from the NI site by the inductive approach. Contrarily, just one out of the two anomalies was captured by the transductive approach. In this and in the upcoming experiment, trimming levels $\alpha_l = \alpha_u = 0.1$ were considered for both training and test sets, while the eigenvalue-ratio restriction was automatically inferred by the estimated group scatters of the known classes.
Table 3.4 reports the confusion matrices for the RAEDDA classifier. The model correctly identifies the presence of a hidden class, recovering the true data partition with an accuracy of 86% (3 misclassified units) and 90% (2 misclassified units) in the transductive and inductive framework, respectively. Considering the challenging classification problem and the limited sample size, the RAEDDA model shows remarkably good performance.

3.4.4 Anomaly and novelty detection: outliers and two unobserved classes

This second experiment considers an even more extreme scenario: the training set contains only 14 observations, among which 12 units truly belong to the NI region, while the remaining 2 come from the AI area but with an incorrect NI label. That is, in the remaining 31 unlabeled units there are two sampling sites, namely AI and NS, that need to be discovered.

TABLE 3.4: Confusion tables for RAEDDA classifier (transductive and inductive inference) on the test set for the Grapevine microbiome data with one unobserved class (NS).

RAEDDA Transductive

Classification	Truth NI	NS	AI
NI	1	1	0
AI	0	0	3
HIDDEN GROUP 1	2	14	0

RAEDDA Inductive

Classification	Truth NI	NS	AI
NI	2	1	0
AI	0	0	3
HIDDEN GROUP 1	1	14	0

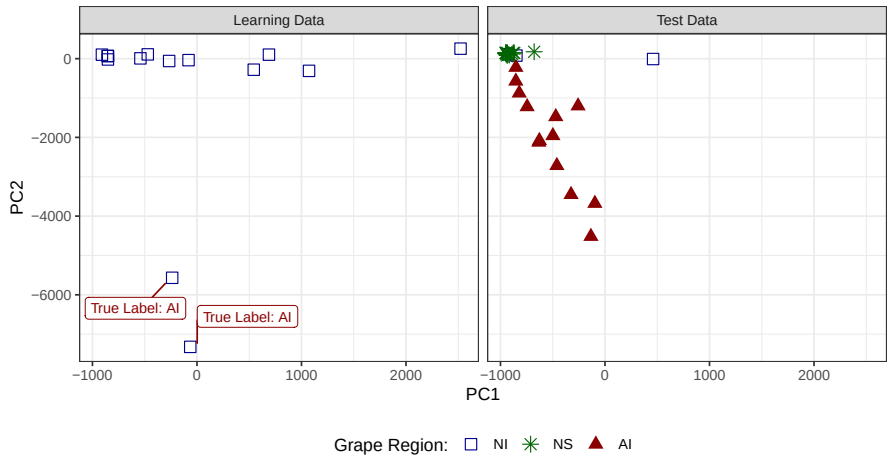

FIGURE 3.11: Learning scenario for anomaly and novelty detection of the grapevine microbiome data on the ROBPCA subspace: 2 unobserved regions and outliers in the training set.

Likewise in the previous Section, ROBPCA retains $d = 2$ principal components when fitted to the training set: the grapevine sample in the robustly estimated subspace is plotted in Figure 3.11. Notice in this context the compelling necessity of performing robust dimensional reduction: the two mislabeled observations from the AI area in the training set can be seen as outliers, and a dimensional reduction technique sensitive to them may have introduced masked and/or swamped units. The RBIC is used to select the best patterned structure and number of components: results are reported in Tables 3.5 and 3.6. Again, also in this more extreme experiment both inferential procedures recover the true number of sites from which the grapes were sampled. Due to the ROBPCA output, in both transductive and inductive approaches the wrongly labelled units in the training set are easily trimmed off and identified as belonging to an area different from NI. Classification results for the chosen model are reported in Table 3.7, where the recovered data partition notably agrees with the 3 different sampling sites, with only 4 and 3 misclassified units for the transductive and inductive estimation, respectively.
In general, considering also additional experiments not reported in the present paper, the inductive approach seems to perform slightly better in terms of anomaly and novelty detection, especially if the sample size of the hidden classes is small. This had already been noted in Bouveyron, 2014, and it may be even more evident in our proposal due to the augmented test set (see end of Section 3.2.3) employed in the discovery phase. For instance, in this experiment, the two AI units that are trimmed off in the learning phase come back again in the parameters estimation of the discovery phase, improving the classifier efficiency. Contrarily, the transductive approach simply does not account for them when estimating the parameters of the NI group.

TABLE 3.5: RBIC for different patterned structures and number of hidden classes for the RAEDDA model, transductive inference. The model with the highest RBIC value is highlighted in bold. Grapevine microbiome data with two unobserved classes (NS and NI).

	Covariance Structure					
# Classes	EII	VII	EEI	VEI	EVI	**VVI**
1	-1339.32	-	-1340.13	-	-	-
2	-1326.16	-1251.13	-1347.46	-1254.85	-1351.08	-1268.71
3	-1337.30	-1240.35	-1358.60	-1244.33	-1365.89	**-1222.04**

TABLE 3.6: RBIC for different patterned structures and number of hidden classes for the RAEDDA model, inductive inference. The models with the highest RBIC value are highlighted in bold. Grapevine microbiome data with two unobserved classes (NS and NI).

Robust learning phase

	Covariance Structure					
# Classes	EII	VII	**EEI**	VEI	EVI	VVI
1	-390.83	-	**-364.67**	-	-	-

Robust discovery phase

	Covariance Structure			
# Classes	EEI	VEI	EVI	**VVI**
1	-3910.27	-	-	-
2	-1418.22	-1042.85	-982.86	-979.16
3	-1104.38	-1037.56	-955.12	**-897.35**

TABLE 3.7: Confusion tables for RAEDDA classifier (transductive and inductive inference) on the test set for the Grapevine microbiome data with two unobserved classes (NS and NI).

RAEDDA Transductive

	Truth		
Classification	NI	NS	AI
NI	1	1	1
HIDDEN GROUP 1	0	0	12
HIDDEN GROUP 2	2	14	0

RAEDDA Inductive

	Truth		
Classification	NI	NS	AI
NI	1	0	1
HIDDEN GROUP 1	2	15	0
HIDDEN GROUP 2	0	0	12

In these preliminary experiments we have positively assessed how our proposal, coupled with a robust dimension reduction technique, can be employed in identifying grapes provenance even considering adulteration and sample selection bias. Even though domain-expert supervision will always be crucial for class interpretation when extra groups are detected, an automatic pipeline that performs microbiome composition, dimension reduction and robust and adaptive classification seems a promising procedure for enhancing the quality, speed and mechanization of food authenticity analyses.

3.5 Concluding remarks

In the present chapter we have proposed a model-based discriminant analysis method for anomaly and novelty detection. We have shown that the methodology effectively performs classification in presence of label noise, outliers and unobserved classes in the test set. By incorporating impartial trimming and eigenvalue-ratio constraints, our proposal robustly estimates model parameters of known and hidden classes, identifying as a by-product wrongly labelled and/or adulterated observations. Considering a parsimonious family of patterned models, two flexible EM-based approaches have been proposed for parameter estimation: one based on the union of training and test sets, and the other made of two phases, performing sequential inference for known and hidden groups. Furthermore, we let the latter approach exploit the partial order structure of the parsimonious models, deriving fast and closed-form solutions for estimating the parameters of the extra classes. The resulting methodology includes several model-based classification methods as special cases. A robust data-driven criterion has been adapted for selecting the number of unobserved groups and constraint strength in covariances estimation. An extensive simulation study and applications on a grapevine microbiome dataset have proved the effectiveness of our proposal. Particularly, the classifier capability in discriminating (known and previously unobserved) grape provenances, within an adulterated context, may lead to promising developments in the food authenticity domain.
Further research directions include a data-driven procedure for selecting reasonable values for the trimming levels, and a metric that automatically categorizes trimmed units as being affected by label and/or attribute noise. Additionally, the definition of a general framework for robust and adaptive variable selection and classification, suitable for data of large dimensions, is imperative in domains like chemometrics, computer vision and genetics. About that, two robust methodologies for variable selection that deal with the issue in the fully-supervised, i.e., not adaptive, framework are introduced in Chapter 4. Based on such methodologies, an extension to the adaptive learning scenario is currently under study and it will be object of future developments.

3.6 Appendix A

This appendix provides closed form solutions for the estimation of the covariance matrices $\boldsymbol{\Sigma}_h$, $h = G+1, \ldots, E$ of the unobserved classes via the inductive approach; our main reference here is the seminal paper of Celeux and Govaert, 1995, where patterned covariance matrices were firstly defined and algorithms for their ML estimation were proposed. In the robust discovery phase only the parameters for the $H = E - G$ densities need to be estimated, according to the available patterned models, given the one considered in the Learning Phase (see Figure 3.4). Denote with $\boldsymbol{W}_h = \sum_{m=1}^{M} \varphi(\mathbf{y}_m)\hat{z}_{mh}\left[(\mathbf{y}_m - \hat{\boldsymbol{\mu}}_h)(\mathbf{y}_m - \hat{\boldsymbol{\mu}}_h)'\right]$ and let $\boldsymbol{W}_h = \boldsymbol{L}_h\boldsymbol{\Delta}_h\boldsymbol{L}_h^{'}$ be its eigenvalue decomposition. Further, consider $n_h = \sum_{m=1}^{M} \varphi(\mathbf{y}_m)\hat{z}_{mh}$ for $h = G+1, \ldots, E$. Lastly, denote with a bar the estimates obtained in the robust learning phase for the G known groups: they are fixed and should not be changed. The formulae needed for the parameter updates are as follows:

- VII model: $\boldsymbol{\Sigma}_h = \lambda_h \boldsymbol{I}$

$$\hat{\lambda}_h = \frac{\text{tr}(\boldsymbol{W}_h)}{p\, n_h}, \qquad h = G+1, \ldots, E.$$

- VEI model: $\boldsymbol{\Sigma}_h = \lambda_h \bar{\boldsymbol{A}}$

$$\hat{\lambda}_h = \frac{\text{tr}(\boldsymbol{W}_h \bar{A}^{-1})}{p\, n_h}, \qquad h = G+1, \ldots, E.$$

- EVI model: $\boldsymbol{\Sigma}_h = \bar{\lambda}\boldsymbol{A}_h$

$$\hat{\boldsymbol{A}}_h = \frac{\text{diag}(\boldsymbol{W}_h)}{|\text{diag}(\boldsymbol{W}_h)|^{1/p}}, \qquad h = G+1, \ldots, E.$$

- VVI model: $\boldsymbol{\Sigma}_h = \lambda_h \boldsymbol{A}_h$

$$\hat{\lambda}_h = \frac{|\text{diag}(\boldsymbol{W}_h)|^{1/p}}{n_h}, \qquad h = G+1, \ldots, E.$$

$$\hat{\boldsymbol{A}}_h = \frac{\text{diag}(\boldsymbol{W}_h)}{|\text{diag}(\boldsymbol{W}_h)|^{1/p}}, \qquad h = G+1, \ldots, E.$$

- VEE model: $\boldsymbol{\Sigma}_h = \lambda_h \bar{\boldsymbol{D}}\bar{\boldsymbol{A}}\bar{\boldsymbol{D}}^{'}$

  Let $\bar{\boldsymbol{C}} = \bar{\boldsymbol{D}}\bar{\boldsymbol{A}}\bar{\boldsymbol{D}}^{'}$ and

$$\hat{\lambda}_h = \frac{\text{tr}(\boldsymbol{W}_h \bar{\boldsymbol{C}}^{-1})}{p\, n_h}, \qquad h = G+1, \ldots, E.$$

- EVE model: $\boldsymbol{\Sigma}_h = \bar{\lambda}\bar{\boldsymbol{D}}\boldsymbol{A}_h\bar{\boldsymbol{D}}^{'}$

$$\hat{\boldsymbol{A}}_h = \frac{\text{diag}(\bar{\boldsymbol{D}}^{'}\boldsymbol{W}_h\bar{\boldsymbol{D}})}{|\text{diag}(\bar{\boldsymbol{D}}^{'}\boldsymbol{W}_h\bar{\boldsymbol{D}})|^{1/p}}, \qquad h = G+1, \ldots, E.$$

- EEV model: $\boldsymbol{\Sigma}_h = \bar{\lambda}\boldsymbol{D}_h\bar{\boldsymbol{A}}\boldsymbol{D}_h^{'}$

$$\hat{\boldsymbol{D}}_h = \boldsymbol{L}_h, \qquad h = G+1, \ldots, E.$$

- VVE model: $\boldsymbol{\Sigma}_h = \lambda_h \bar{\boldsymbol{D}}\boldsymbol{A}_h\bar{\boldsymbol{D}}'$

Let $\boldsymbol{R}_h = \lambda_h \boldsymbol{A}_h$

$$\hat{\boldsymbol{R}}_h = \frac{1}{n_h}\text{diag}(\bar{\boldsymbol{D}}^{'}\boldsymbol{W}_h\bar{\boldsymbol{D}}), \qquad h = G+1, \ldots, E.$$

and, subsequently

$$\hat{\lambda}_h = |\hat{\boldsymbol{R}}_h|^{1/p}, \qquad h = G+1, \ldots, E.$$

$$\hat{\boldsymbol{A}}_h = \frac{1}{\hat{\lambda}_h}\hat{\boldsymbol{R}}_h, \qquad h = G+1, \ldots, E.$$

- VEV model: $\boldsymbol{\Sigma}_h = \lambda_h \boldsymbol{D}_h \bar{\boldsymbol{A}} \boldsymbol{D}_h^{'}$

$$\hat{\boldsymbol{D}}_h = \boldsymbol{L}_h, \qquad h = G+1, \ldots, E.$$

$$\hat{\lambda}_h = \frac{\text{tr}(\boldsymbol{W}_h \hat{\boldsymbol{D}}_h \bar{\boldsymbol{A}}^{-1} \hat{\boldsymbol{D}}_h{}')}{p\, n_h}, \qquad h = G+1, \ldots, E.$$

- EVV model: $\boldsymbol{\Sigma}_h = \bar{\lambda} \boldsymbol{D}_h \boldsymbol{A}_h \boldsymbol{D}_h^{'}$

  Let $\boldsymbol{C}_h = \boldsymbol{D}_h \boldsymbol{A}_h \boldsymbol{D}_h^{'}$

$$\hat{\boldsymbol{C}}_h = \frac{\boldsymbol{W}_h}{|\boldsymbol{W}_h|^{1/p}}, \qquad h = G+1, \ldots, E.$$

  $\hat{\boldsymbol{A}}_h$, $\hat{\boldsymbol{D}}_h$ are obtained through the eigenvalue decomposition of $\hat{\boldsymbol{C}}_h$, $h = G+1, \ldots, E$.

- VVV model: $\boldsymbol{\Sigma}_h = \lambda_h \boldsymbol{D}_h \boldsymbol{A}_h \boldsymbol{D}_h^{'}$

$$\hat{\boldsymbol{\Sigma}}_h = \frac{1}{n_h}\boldsymbol{W}_h$$

  $\hat{\lambda_h}$, $\hat{\boldsymbol{A}}_h$, $\hat{\boldsymbol{D}_h}$ are obtained through the eigenvalue decomposition of $\hat{\boldsymbol{\Sigma}}_h$, $h = G+1, \ldots, E$.

Chapter 4

Robust variable selection in model-based learning

Based on:

Cappozzo, A., Greselin, F., Murphy, T. B.
"Robust variable selection in model-based learning"
In preparation

4.1 Introduction

The problem of identifying the most discriminating features when performing supervised learning has been extensively investigated in the past years. In particular, several methods for variable selection in model-based classification have been proposed. Surprisingly, the impact that outliers and wrongly labeled units cause on the determination of relevant predictors has received far less attention, with almost no dedicated methodologies available in the literature: no one of the wrapper methods listed in Section 1.4.2 provide protection against outliers and label noise. Nonetheless, the presence of only few adulterated data points can severely undermine the variable selection results (see Section 4.3).
A notable exception regards two existing approaches that already provide robust selection of variables. In linear discriminant analysis (LDA), early-stage wrapper methods consider the employment of stepwise procedures in testing for no additional information, like the stepwise MANOVA described in Section 12.3 of McLachlan, 1992: these are usually based on the likelihood ratio test Wilks' Λ statistic. By respectively employing M-estimates and MCD-estimates to obtain a robust version of the Wilks' Λ statistics, Krusińska and Liebhart, 1988 and Todorov, 2007 developed LDA-based techniques for variable selection resistant to outliers. In addition, a methodology to achieve feature selection for classification problems polluted by label noise is proposed in Frénay et al., 2014.
Nevertheless, to our best knowledge, wrapper methods that perform robust feature selection in a more general framework and accounting also for label noise are still missing in the literature. In order to overcome this limitation, the present chapter proposes two approaches for robust variable selection in model-based classification: one that embeds the robust classifier developed in Chapter 2 in a greedy-forward stepwise procedure for model selection (Section 4.2.1); and the other based on the theory of maximum likelihood estimation and the notion of irrelevant variables within robust ML estimation of normal mixtures (Section 4.2.2).
The remaining of the chapter is structured as follows. The two novel variable selection techniques resistant to outliers and label noise are introduced in Section 4.2. Section 4.3 is devoted

to the comparison of several feature selection procedures within a simulation study in an artificially contaminated scenario. Section 4.4 presents a high-dimensional discrimination study where the robust variable selection methods described in Section 4.2 are successfully applied to a chemometric contest. Section 4.5 concludes the chapter outlying some remarks and future research directions. Technical issues and computational details for the two novel methods are respectively deferred to Appendix A (Section 4.6) and B (Section 4.7).

4.2 Robust variable selection in model-based classification

In the present Section we introduce two novel wrapper approaches for robust variable selection in high-dimensional model-based classification.
In Section 4.2.1, the REDDA method (Section 2.2.2) is embedded in a greedy-forward procedure for model selection. A robust classification rule is constructed in a stepwise manner, by considering the inclusion of extra variables into the model and also the removal of existing variables from the model conditioning on their discriminating power. Particularly, the selection procedure is based on a robust information criterion, that accounts for the possible presence of outliers and label noise in the dataset.
In Section 4.2.2, the theory of maximum likelihood estimation and the notion of irrelevant variables for normal mixtures is employed for defining a ML subset selector, along the lines of the procedure introduced in section 5.3.3 of Ritter, 2014 for the unsupervised framework. The identification of the relevant subset is regarded as a parameter to be estimated via ML: an EM-based procedure is derived for maximizing the objective function. The Section concludes with a comparison, highlighting strengths and weaknesses of the two proposals.

4.2.1 The robust stepwise greedy-forward approach via TBIC

The present procedure searches for the set of relevant variables in a greedy-stepwise manner. That is, we start from the empty set and we sequentially add relevant variables until no more discriminating features are available. More specifically, following the notation introduced in Sections 1.2.1 and 2.2, in each step of the algorithm we partition the learning observations $\mathbf{x}_n$, $n = 1, \ldots, N$, into three parts $\mathbf{x}_n = (\mathbf{x}_n^c, x_n^p, \mathbf{x}_n^o)$, where:

- $\mathbf{x}_n^c$ indicates the set of variables currently included in the model
- x_n^p the variable proposed for inclusion
- $\mathbf{x}_n^o$ the remaining variables

In order to decide whether to include the proposed variable x_n^p, we compare the following two competing models:

- *Grouping* ($\mathcal{M}_{GR}$): $p(\mathbf{x}_n|\mathbf{l}_n) = p(\mathbf{x}_n^c, x_n^p, \mathbf{x}_n^o|\mathbf{l}_n) = p(\mathbf{x}_n^c, x_n^p|\mathbf{l}_n)p(\mathbf{x}_n^o|x_n^p, \mathbf{x}_n^c)$
- *No Grouping* ($\mathcal{M}_{NG}$): $p(\mathbf{x}_n|\mathbf{l}_n) = p(\mathbf{x}_n^c, x_n^p, \mathbf{x}_n^o|\mathbf{l}_n) = p(\mathbf{x}_n^c|\mathbf{l}_n)p(x_n^p|\mathbf{x}_n^r \subseteq \mathbf{x}_n^c)p(\mathbf{x}_n^o|x_n^p, \mathbf{x}_n^c)$

where $\mathbf{x}_n^r$ denotes a subset of the currently included variables $\mathbf{x}_n^c$. The grouping model specifies that x_n^p provides extra grouping information beyond that provided by $\mathbf{x}_n^c$; whereas the No Grouping model specifies that x_n^p is conditionally independent of the group membership given $\mathbf{x}_n^r$. The reason for considering $\mathbf{x}_n^r$ in the conditional distribution being that x_n^p might be related to only a subset of the grouping variables $\mathbf{x}_n^c$ (Maugis et al., 2009b; Maugis et al., 2009a; Maugis et al., 2011). The differences between the two models are graphically illustrated in Figure 4.1. The model structure of $p(\mathbf{x}_n^o|x_n^p, \mathbf{x}_n^c)$ is assumed to be the same for both grouping

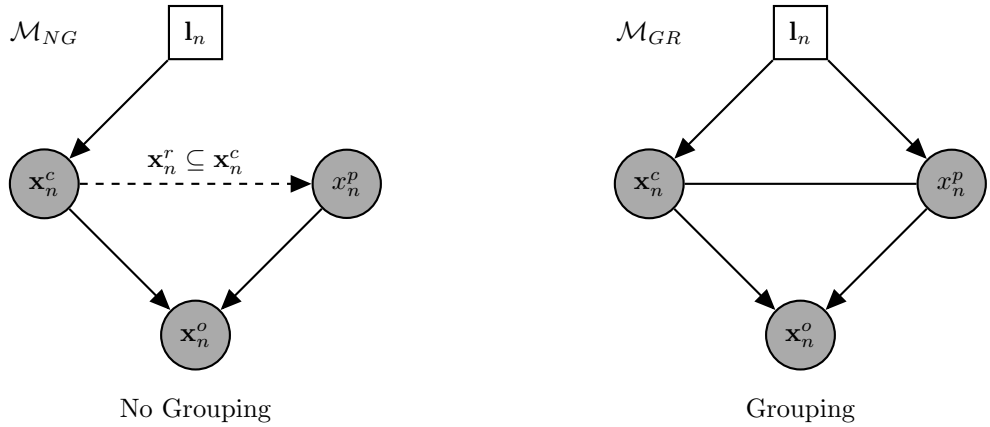

FIGURE 4.1: Graphical Representation of the Grouping and the No Grouping models

and no grouping specification, and we let $p(\mathbf{x}_n^c, x_n^p|\mathbf{l}_n)$ and $p(\mathbf{x}_n^c|\mathbf{l}_n)$ be a normal density with parsimonious covariance structure, according to the model assumptions introduced in Section 1.2.1. Additionally, we assume $p(x_n^p|\mathbf{x}_n^r \subseteq \mathbf{x}_n^c)$ to be a normal linear regression model, as a result from conditional multivariate normal means. The selection of which model to prefer is carried out employing a robust approximation to the Bayes Factor. More specifically, the Bayes Factor (Kass and Raftery, 1995) is equal to the ratio between the integrated likelihood of the two competing models:

$$\mathcal{B}_{GR,NG} = \frac{p(\mathbf{x}_n|\mathcal{M}_{GR})}{p(\mathbf{x}_n|\mathcal{M}_{NG})} = \frac{\int p(\mathbf{x}_n|\boldsymbol{\theta}_{GR}, \mathcal{M}_{GR}) p(\boldsymbol{\theta}_{GR}|\mathcal{M}_{GR}) d\boldsymbol{\theta}_{GR}}{\int p(\mathbf{x}_n|\boldsymbol{\theta}_{NG}, \mathcal{M}_{NG}) p(\boldsymbol{\theta}_{NG}|\mathcal{M}_{NG}) d\boldsymbol{\theta}_{NG}} \tag{4.1}$$

where $\boldsymbol{\theta}_{GR}$ and $\boldsymbol{\theta}_{NG}$ denote the set of parameters for the Grouping and the No Grouping model, respectively. When no prior preference for one of the two models is considered, (4.1) is equal to the posterior odds in favour of $\mathcal{M}_{GR}$. The Bayes Factor can therefore be used for assessing to which extent the data supports the Grouping structure compare to the No Grouping formulation. Along the lines of Raftery and Dean, 2006, the Bayesian Information Criterion

$$BIC = 2 \times log\ maximized\ likelihood - v \log N$$

is used as an approximation for the integrated likelihoods, where v is a penalty term (number of parameters in the model) and N is the sample size (Schwarz, 1978). Thus, twice the logarithm of $\mathcal{B}_{GR,NG}$ can be approximated with

$$2 \log(\mathcal{B}_{GR,NG}) \approx BIC(Grouping) - BIC(No\ Grouping) \tag{4.2}$$

and a variable x_n^p with a positive difference in $BIC(Grouping) - BIC(No\ Grouping)$ is a candidate for being added to the model. For avoiding the detrimental effect that class and attribute noise might produce in the variable selection procedure, the Trimmed BIC (TBIC), firstly introduced in Neykov et al., 2007, is employed as a robust proxy for the quantities in (4.2). Let us

define:

$$TBIC(Grouping) = \underbrace{2\sum_{n=1}^{N}\zeta(\mathbf{x}_n^c, x_n^p)\sum_{g=1}^{G} l_{ng}\log\left(\hat{\tau}_g^{cp}\phi(\mathbf{x}_n^c, x_n^p; \hat{\boldsymbol{\mu}}_g^{cp}, \hat{\boldsymbol{\Sigma}}_g^{cp})\right)}_{2\times\text{trimmed log maximized likelihood of } p(\mathbf{x}_n^c, x_n^p, \mathbf{l}_n)} + \\ - v^{cp}log(N^*) \tag{4.3}$$

$$TBIC(No\ Grouping) = \underbrace{2\sum_{n=1}^{N}\iota(\mathbf{x}_n^c, x_n^p)\sum_{g=1}^{G} l_{ng}\log\left(\hat{\tau}_g^{c}\phi(\mathbf{x}_n^c; \hat{\boldsymbol{\mu}}_g^{c}, \hat{\boldsymbol{\Sigma}}_g^{c})\right)}_{2\times\text{trimmed log maximized likelihood of } p(\mathbf{x}_n^c, \mathbf{l}_n)} - v^c log(N^*) + \\ \underbrace{-N^*\left[\log 2\pi + 2\log\left(\frac{\sum_{n=1}^{N}\iota(\mathbf{x}_n^c, x_n^p)(x_n^p - (\hat{\alpha} + \hat{\boldsymbol{\beta}}'\mathbf{x}_n^r))^2}{N^*}\right) + 1\right]}_{2\times\text{trimmed log maximized likelihood of } p(x_n^p|\mathbf{x}_n^r\subseteq\mathbf{x}_n^c)} + \\ - v^p log(N^*). \tag{4.4}$$

Alternatively, (4.4) can be written as:

$$TBIC(No\ Grouping) = \underbrace{2\sum_{n=1}^{N}\iota(\mathbf{x}_n^c, x_n^p)\sum_{g=1}^{G} l_{ng}\log\left(\hat{\tau}_g^{c}\phi(\mathbf{x}_n^c; \hat{\boldsymbol{\mu}}_g^{c}, \hat{\boldsymbol{\Sigma}}_g^{c})\right)}_{2\times\text{trimmed log maximized likelihood of } p(\mathbf{x}_n^c, \mathbf{l}_n)} - v^c log(N^*) + \\ \underbrace{+2\sum_{n=1}^{N}\iota(\mathbf{x}_n^c, x_n^p)\log\left[\phi\left(x_n^p; \hat{\alpha} + \hat{\boldsymbol{\beta}}'\mathbf{x}_n^r, \hat{\sigma}^2\right)\right]}_{2\times\text{trimmed log maximized likelihood of } p(x_n^p|\mathbf{x}_n^r\subseteq\mathbf{x}_n^c)} - v^p log(N^*). \tag{4.5}$$

The penalty terms v^{cp} and v^c indicate the number of parameters for a REDDA model respectively estimated on the set of variables $\mathbf{x}_n^c, x_n^p$ and $\mathbf{x}_n^c$; while v^p accounts for the number of parameters in the linear regression of x_n^p on $\mathbf{x}_n^r$. The 0-1 indicator functions $\zeta(\cdot)$ and $\iota(\cdot)$ identify the subset of observations that have null weight in the trimmed likelihood under the grouping and no grouping models, with $N^* = \sum_{n=1}^{N}\zeta(\mathbf{x}_n) = \sum_{n=1}^{N}\iota(\mathbf{x}_n)$.
In detail, the parameters $\left\{\tau_g^{cp}, \boldsymbol{\mu}_g^{cp}\ \boldsymbol{\Sigma}_g^{cp}\right\}$, $g = 1, \ldots, G$ of the grouping model are estimated through a standard REDDA fitted on the variables $\mathbf{x}_n^c, x_n^p$, in which the C-step is enforced discarding $\lfloor N\alpha_l \rfloor$% of the samples with lowest value of

$$D_{Grouping}\left(\mathbf{x}_n^c, x_n^p; \hat{\boldsymbol{\theta}}_{GR}\right) = \sum_{g=1}^{G} l_{ng}\log\left[\phi\left(\mathbf{x}_n^c, x_n^p; \hat{\boldsymbol{\mu}}_g^{cp}, \hat{\boldsymbol{\Sigma}}_g^{cp}\right)\right] \quad n = 1, \ldots, N. \tag{4.6}$$

For the no grouping model, REDDA needs to be fitted only on the set of currently included variables $\mathbf{x}_n^c$, coupled with the linear regression of x_n^p on $\mathbf{x}_n^r$. For this case, the discriminating function reads:

$$D_{No\ Grouping}\left(\mathbf{x}_n^c, x_n^p; \hat{\boldsymbol{\theta}}_{NG}\right) = \sum_{g=1}^{G} l_{ng}\log\left[\phi\left(\mathbf{x}_n^c; \hat{\boldsymbol{\mu}}_g^{c}, \hat{\boldsymbol{\Sigma}}_g^{c}\right)\right] + \\ + \log\left[\phi(x_n^p; \hat{\alpha} + \hat{\boldsymbol{\beta}}'\mathbf{x}_n^r, \hat{\sigma}^2)\right] \tag{4.7}$$

for $n = 1, \ldots, N$. That is, at each iteration of the procedure that leads to the final robust estimates, we discard the $\lfloor N\alpha_l \rfloor$% of the sample with the lowest contribution to the conditional likelihood under the no grouping model. Once the C-step is enforced, the set of parameters $\{\alpha, \boldsymbol{\beta}, \sigma^2\}$ for the regression part is robustly estimated via ML on the untrimmed observations, in which a stepwise method is employed for automatically choosing the subset of regressors $\mathbf{x}_n^r$. Details on the procedure are reported in Appendix A (Section 4.6).
After each addition stage, we make use of the same procedure described above to check whether an already chosen variable in $\mathbf{x}_n^c$ should be removed: in this case x_n^p takes the role of the variable to be dropped, and a positive difference in terms of TBIC implies the exclusion of x_n^p to the set of currently included variables. The procedure iterates between variable addition and removal stage until two consecutive steps have been rejected, then it stops. Notice that, whenever $\alpha_l = 0$, BIC and TBIC coincide and the entire approach reduces to the methodology described in Maugis et al., 2011.
A last worthy note regards the theoretical justification for the employment of TBIC as an approximation of the integrated likelihood. The rationale arises from the spurious outliers model, firstly defined in Gallegos and Ritter, 2005 and briefly introduced in Section 1.3.1, as the probabilistic specification for the contaminated sub-sample. Let q_n denote an indicator of genuine observations, such that $q_n = 1$ when $\{(\mathbf{x}_n, \mathbf{l}_n)\}$ is a "regular" unit and $q_n = 0$ whenever $\{(\mathbf{x}_n, \mathbf{l}_n)\}$ presents some sort of contamination/adulteration. Notice that the complete observation $\{(\mathbf{x}_n, \mathbf{l}_n)\}$ might be regarded as an outlier whenever either the associated label and/or some of its predictors present unusual values. In such a way, we account for both attribute and class noise. The data generating distribution for a specific observation $\{(\mathbf{x}_n, \mathbf{l}_n)\}$ is then assumed to be as follows:

$$p(\mathbf{x}_n, \mathbf{l}_n | q_n; \boldsymbol{\theta}) = p(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\theta})^{q_n} w(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\psi}_n)^{(1-q_n)} \tag{4.8}$$

where $p(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\theta})$ denotes the probability distribution for the regular bulk of the data, in our context being alternatively the Grouping or the No Grouping model; and $w(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\psi}_n)$ is an almost arbitrary, subject specific probability density function, parametrized by $\boldsymbol{\psi}_n \in \boldsymbol{\Psi}_n$. For an independent sample of N observations, the likelihood for the model in (4.8) is therefore given by:

$$\prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\theta})^{q_n} \prod_{n=1}^{N} w(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\psi}_n)^{(1-q_n)} \tag{4.9}$$

where a fixed α_l% of contamination is assumed such that $N^* = \sum_{n=1}^{N} q_n = \lceil N(1-\alpha_l) \rceil$.
Let $\mathcal{N} = \{N_1, N_0\}$ be a partition of N into regular and non-regular observations, indexed by q_n being either 1 or 0 for $n = 1, \ldots, N$, with $|N_1| = \lceil N(1-\alpha_l) \rceil$ and $|N_0| = \lfloor N\alpha_l \rfloor$, respectively. Further, denote with $\mathcal{D}(N)$ the set of all partitions of such type, with $|\mathcal{D}(N)| = \binom{N}{\lceil N(1-\alpha_l) \rceil}$. The non-regular contribution of the contaminated observations can be avoided in maximizing (4.9) with respect to $\boldsymbol{\theta}$ when the $w(\cdot; \boldsymbol{\psi}_n)$'s satisfy

$$\arg\max_{\mathcal{N} \in \mathcal{D}(N)} \max_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\theta})^{q_n} \subseteq \arg\max_{\mathcal{N} \in \mathcal{D}(N)} \max_{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N} \prod_{n=1}^{N} w(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\psi}_n)^{(1-q_n)}. \tag{4.10}$$

The condition in (4.10) means that the configuration that maximizes the first factor in (4.9) also maximizes the second one (Gallegos and Ritter, 2005). More specifically, the partitions assigning $\lceil N(1-\alpha_l) \rceil$ regular units that maximize the likelihood of the genuine observations are contained in the set of partitions assigning $\lfloor N\alpha_l \rfloor$ non regular units that maximize the likelihood corresponding to the noise. Condition (4.10) holds under general and non-restrictive assumptions on the non regular units, particularly, $w(\cdot; \boldsymbol{\psi}_n)$ can easily accommodate observations that

can be merely regarded as outliers (Gallegos and Ritter, 2005; García-Escudero et al., 2008). The contaminated observations are therefore no more considered in the estimation process, and the model log-likelihood simplifies to:

$$\sum_{n=1}^{N} q_n \log p(\mathbf{x}_n, \mathbf{l}_n; \boldsymbol{\theta}) \tag{4.11}$$

to be maximized with respect to the set of parameters $\boldsymbol{\theta}$; details are reported in Appendix A (Section 4.6). Finally, the integrated likelihood for (4.11) can be approximated via the Bayesian Information Criterion:

$$2 \sum_{n=1}^{N} q_n \log p(\mathbf{x}_n, \mathbf{l}_n; \hat{\boldsymbol{\theta}}) - v \log N^* \tag{4.12}$$

where $\hat{\boldsymbol{\theta}}$ denotes MLE for the simplified log-likelihood, v is the number of parameters and N^* is the number of data values that contribute to the summation in (4.11) (Kass, 1993). An initial attempt of generalizing and formally proving this result is reported in Appendix D. Depending which scenario is considered, (4.12) defines (4.3) or (4.4) under the Grouping and the No Grouping model, respectively.

4.2.2 The ML subset selector approach

The second approach we consider for robust variable selection in model-based classification stems from the maximum likelihood subset selector theory developed for clustering, where the main reference is Section 5.3.3 of Ritter, 2014. Particularly, being classification a generally simpler problem than unsupervised learning, the ML subset selection ideas are naturally adapted to a robust supervised context with variable selection. Here we build a model for the entire P-dimensional space in which the observations lie, and exploit theoretical results for the conditional distribution of the multivariate Gaussian under irrelevance. Let us introduce the following notation: for a positive semi-definite matrix $\boldsymbol{\Sigma} \in PD(P)$, denote its restriction to the variables in $F \subseteq 1, \ldots, P$ by $\boldsymbol{\Sigma}_F$, with size $|F| = p$. The block-wise representation of $\boldsymbol{\Sigma}$, via the natural order of F, is therefore:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_F & \boldsymbol{\Sigma}_{F,E} \\ \boldsymbol{\Sigma}_{E,F} & \boldsymbol{\Sigma}_E \end{pmatrix}$$

with $E = \bar{F}$ and $|E| = P - p$. Analogously, the vector $\boldsymbol{\mu}_F$ is the projection of $\boldsymbol{\mu} \in \mathbb{R}^P$ onto the variables in F, following the natural order of F. For a generic observation $\mathbf{x}_n \in \mathbb{R}^P$, the canonical projection of a normal distribution to a subset F of variables is described by the restrictions $\boldsymbol{\mu}_F$ and $\boldsymbol{\Sigma}_F$ of its parameters, with the equality $N_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}_{n,F}) = N_{\boldsymbol{\mu}_F,\boldsymbol{\Sigma}_F}(\mathbf{x}_{n,F})$ such that $\mathbf{x}_{n,F} \sim N(\boldsymbol{\mu}_F, \boldsymbol{\Sigma}_F)$. Considering the notation introduced in Section 1.2.1 and applying standard results for multivariate normal theory, (see, for example, Theorem 3.2.4 in Mardia et al., 1979), the conditional distribution of $\mathbf{x}_{n,E}$ given $\mathbf{x}_{n,F}, \mathbf{l}_n$ reads:

$$\mathbf{x}_{n,E} | \mathbf{x}_{n,F}, \mathbf{l}_n = g \sim \phi(\mathbf{x}_{n,E} - \boldsymbol{G}_{g,E|F}\mathbf{x}_{n,F}; \boldsymbol{\mu}_{g,E|F}, \boldsymbol{\Sigma}_{g,E|F}) \tag{4.13}$$

where $\boldsymbol{\mu}_{g,E|F} = \boldsymbol{\mu}_{g,E} - \boldsymbol{G}_{g,E|F}\boldsymbol{\mu}_{g,F}$, $\boldsymbol{\Sigma}_{g,E|F} = \boldsymbol{\Sigma}_{g,E} - G_{g,E|F}\boldsymbol{\Sigma}_{g,F,E}$ and $\boldsymbol{G}_{g,E|F} = \boldsymbol{\Sigma}_{g,E,F}\boldsymbol{\Sigma}_{g,F}^{-1}$, $g = 1, \ldots, G$. Now assume that E is an irrelevant subset with respect to F, that is, the class membership $\mathbf{l}_n$ is conditionally independent of $\mathbf{x}_{n,E}$ given $\mathbf{x}_{n,F}$. By Lemma 5.2 and Theorem 5.7 of Ritter, 2014, the parameters $\boldsymbol{G}_{g,E|F}$, $\boldsymbol{\mu}_{g,E|F}$ and $\boldsymbol{\Sigma}_{g,E|F}$ do not depend on the class g; applying the

product formula we obtain the following specification for the joint density of $(\mathbf{x}_{n,F}, \mathbf{x}_{n,E}, \mathbf{l}_n)$:

$$\begin{aligned} p(\mathbf{x}_{n,F}, \mathbf{x}_{n,E}, \mathbf{l}_n) &= p(\mathbf{x}_{n,F}, \mathbf{x}_{n,E}|\mathbf{l}_n)p(\mathbf{l}_n) = \\ &= p(\mathbf{x}_{n,E}|\mathbf{x}_{n,F}, \mathbf{l}_n)p(\mathbf{x}_{n,F}|\mathbf{l}_n)p(\mathbf{l}_n) = \\ &= p(\mathbf{x}_{n,E}|\mathbf{x}_{n,F})p(\mathbf{x}_{n,F}|\mathbf{l}_n)p(\mathbf{l}_n). \end{aligned} \tag{4.14}$$

Therefore, for a sample of N observations, the associated trimmed log-likelihood for the probability density in (4.14) is:

$$\begin{aligned} \ell_{trim}(\boldsymbol{\tau}, \boldsymbol{\mu}_F, \boldsymbol{\Sigma}_F, \boldsymbol{G}_{E|F}, \boldsymbol{\mu}_{E|F}, \boldsymbol{\Sigma}_{E|F}|\mathbf{X}, \mathbf{l}) = \\ = \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \left(\sum_{g=1}^{G} l_{ng} \log \left[\tau_g \phi(\mathbf{x}_{n,F}; \boldsymbol{\mu}_{g,F}, \boldsymbol{\Sigma}_{g,F}) \right] + \right. \\ \left. + \log \left[\phi(\mathbf{x}_{n,E} - \boldsymbol{G}_{E|F}\mathbf{x}_{n,F}; \boldsymbol{\mu}_{E|F}, \boldsymbol{\Sigma}_{E|F}) \right] \right) \end{aligned} \tag{4.15}$$

where the identification of the relevant variables belonging to the subset F is regarded as a model parameter. Maximization of (4.15) is carried out via a modification of the EMST algorithm introduced in Ritter, 2014, adapted to the classification framework and extended to flexibly account for the entire family of patterned models of Bensmail and Celeux, 1996. The main steps involving the estimation procedure are given below, further details concerning the implementation can be found in Appendix B (Section 4.7).

1. *Robust Initialization:*

   - If N is sufficiently large compared to P and G, draw a random $(P+1)$-subset for each class g, $g = 1, \ldots, G$, set $\zeta(\mathbf{x}_n) = 1$ if $\mathbf{x}_n$ belongs to any of such G subsets, otherwise set $\zeta(\mathbf{x}_n) = 0$. Go to step 2 of the algorithm.
   - If N is small compared to P and G, draw a random $(p+1)$-subset for each class g, $g = 1, \ldots, G$ and set $\zeta(\mathbf{x}_n) = 1$ if $\mathbf{x}_n$ belongs to any of such G subsets, otherwise set $\zeta(\mathbf{x}_n) = 0$.
     Draw a random subset $\hat{F}^{(0)}$ of dimension p from $1, \ldots, P$ and compute:

$$\hat{\boldsymbol{\mu}}_{g,\hat{F}^{(0)}} = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} \mathbf{x}_{n,\hat{F}^{(0)}}}{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng}} \quad g = 1, \ldots, G,$$

     and $\hat{\boldsymbol{\Sigma}}^{(0)}_{g,\hat{F}^{(0)}}$, $g = 1, \ldots, G$, depending on the considered patterned model, refer to Bensmail and Celeux, 1996 for the details. Lastly, update the trimming function $\zeta(\mathbf{x}_n)$, $n = 1, \ldots, N$, setting $\zeta(\mathbf{x}_n) = 0$ for $\lfloor N\alpha_l \rfloor$% of the samples with lowest value of

$$l_{ng} \log \left[\phi(\mathbf{x}_{n,F^{(0)}}; \hat{\boldsymbol{\mu}}^{(0)}_{g,F^{(0)}}, \hat{\boldsymbol{\Sigma}}^{(0)}_{g,F^{(0)}}) \right]$$

     and $\zeta(\mathbf{x}_n) = 1$ otherwise.

2. (*M-step*)

   Compute:

$$\hat{\tau}_g = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng}}{\lceil N(1-\alpha_l) \rceil} \quad g = 1, \ldots, G$$

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng} \mathbf{x}_n}{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng}} \quad g = 1, \ldots, G.$$

Estimation of $\boldsymbol{\Sigma}_g$ depends on the considered patterned model, details are given in Bensmail and Celeux, 1996.

Notice that the estimates are computed for the full dimension P, that is $\hat{\boldsymbol{\mu}}_g \in \mathbb{R}^P$ and $\hat{\boldsymbol{\Sigma}}_g \in PD(P)$, respectively. In addition, robustly compute also the pooled mean:

$$\hat{\boldsymbol{\mu}} = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) \mathbf{x}_n}{\lceil N(1-\alpha_l) \rceil}.$$

Depending on the considered patterned model, formulae for the associated pooled estimate $\hat{\boldsymbol{\Sigma}}$ are detailed in Appendix B (Section 4.7).

3. (*S-step*)

   Minimize the difference:

$$h(F) = \sum_{g=1}^{G} \hat{\tau}_g \log\det \hat{\boldsymbol{\Sigma}}_{g,F} - \log\det \hat{\boldsymbol{\Sigma}}_F \tag{4.16}$$

   w.r.t. the subset $\hat{F} \subseteq 1, \ldots, P$, with $|\hat{F}| = p$, where $\hat{\boldsymbol{\Sigma}}_{g,\hat{F}}$ is the restriction of $\hat{\boldsymbol{\Sigma}}_g$ to $\hat{F}$. The minimization of (4.16) involves a discrete structure optimization, that becomes quickly unfeasible as $\binom{P}{p}$ grows: a genetic algorithm is proposed for solving it. More details are reported in Appendix B (Section 4.7).

4. (*T-step*)

   Compute the MLE's for the regression parameters

$$\hat{\boldsymbol{G}}_{\hat{E}|\hat{F}} = \hat{\boldsymbol{\Sigma}}_{\hat{E},\hat{F}} \hat{\boldsymbol{\Sigma}}_{\hat{F}}^{-1}$$

$$\hat{\boldsymbol{\mu}}_{\hat{E}|\hat{F}} = \hat{\boldsymbol{\mu}}_{\hat{E}} - \hat{\boldsymbol{G}}_{\hat{E}|\hat{F}} \hat{\boldsymbol{\mu}}_{\hat{F}}$$

$$\hat{\boldsymbol{\Sigma}}_{\hat{E}|\hat{F}} = \hat{\boldsymbol{\Sigma}}_{\hat{E}} - \hat{\boldsymbol{\Sigma}}_{\hat{E},\hat{F}} \hat{\boldsymbol{\Sigma}}_{\hat{F}}^{-1} \hat{\boldsymbol{\Sigma}}_{\hat{F},\hat{E}}$$

   And update the value of the trimming function $\zeta(\cdot)$, setting $\zeta(\mathbf{x}_n) = 0$ for $\lfloor N\alpha_l \rfloor$% of the samples with lowest value of

$$\sum_{g=1}^{G} l_{ng} \log\left[\hat{\tau}_g \phi(\mathbf{x}_{n,\hat{F}}; \hat{\boldsymbol{\mu}}_{g,\hat{F}}, \hat{\boldsymbol{\Sigma}}_{g,\hat{F}})\right] + \log\left[\phi\left(\mathbf{x}_{n,\hat{E}} - \hat{\boldsymbol{G}}_{\hat{E}|\hat{F}} \mathbf{x}_{n,\hat{F}}; \hat{\boldsymbol{\mu}}_{\hat{E}|\hat{F}}, \hat{\boldsymbol{\Sigma}}_{\hat{E}|\hat{F}}\right)\right]$$

5. Iterate $2-4$ until the $\lfloor N\alpha_l \rfloor$ discarded observations are exactly the same on two consecutive iterations, then stop.

The procedure described in steps 1-5 shall be performed `n_init` times, and the parameter estimates that lead to the highest value of the objective function (4.15), out of `n_init` repetitions, provide the final estimated quantities. As a last worthy comment, notice that the specification of the cardinality of F, i.e., the number p of relevant variables that are sought by the estimation algorithm, is a-priori required as a model hyper-parameter.

4.2.3 Methods comparison

In the previous subsections two novel methods for robust variable selection in model-based classification have been introduced. As already anticipated, the main operational difference between the two relies on the fact that the ML subset selector requires the a-priori specification of the subset-size p, whereas the greedy-forward approach via TBIC automatically infers the number of relevant variables by means of a stopping criterion in the stepwise search. This could come both as an advantage and as a disadvantage: one may desire to specifically retain the p most relevant variables (i.e., $p = 2$ for visualization purposes). In this case, the ML subset selector approach shall be preferred, as the entire feature space P is accounted for in the likelihood specification in (4.15), contrarily to the greedy approach employed in Section 4.2.1. If this is not the case, run the algorithm for a reasonable range of values p and select the favourite solution, consensus methods like the one in Strehl and Ghosh, 2003 for clustering can be adapted to the classification framework. In addition, if computational burden is not an issue, the greedy-forward approach via TBIC can be firstly employed for assessing the order of magnitude of the subset size, and afterwards the ML subset selector can be run varying p in the proximity of the number of relevant variables found by the former method, qualitatively assessing the difference.
Clearly, the suggestions above are mostly heuristic, a more formal treatment on how to compare and validate results from both procedures is still missing: this however goes beyond the scope of the present manuscript and it will be the object of future research.

4.3 Simulation study

The aim of this simulated example is to numerically assess the effectiveness of the methodologies introduced in Section 4.2, whilst investigating the effect that a (small) percentage of contamination has on standard variable selection procedures. In doing so, we decided to rely on the same data generating process (DGP) considered in Maugis et al., 2011 and Celeux et al., 2019, including in addition some attribute and class noise to the original experiment.

4.3.1 Experimental setup

The synthetic dataset considers $G = 4$ classes for a total of $P = 16$ features: the first three are relevant for the classification, the subsequent four are redundant given the first ones while the last nine are independent from both the group variable and the previous predictors. The prior probabilities of the four classes are equal to $\boldsymbol{\tau} = (0.15, 0.3, 0.2, 0.35)$. On the three discriminant variables, data are generated from multivariate normal densities

$$\mathbf{x}_n^{[1-3]} | \mathbf{l}_n = g \sim \phi\left(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right), \quad g = 1, \ldots, 4$$

with mean vectors

$$\boldsymbol{\mu}_1 = (1.5, -1.5, 1.5)', \quad \boldsymbol{\mu}_2 = (-1.5, 1.5, 1.5)'$$
$$\boldsymbol{\mu}_3 = (1.5, -1.5, -1.5)', \quad \boldsymbol{\mu}_4 = (-1.5, 1.5, -1.5)'$$

and covariance matrices $\boldsymbol{\Sigma}_g$ with elements $\rho_g^{|i-j|}$, $1 \leq i, j \leq 3$, and $\rho_1 = 0.85$, $\rho_2 = 0.1$, $\rho_3 = 0.65$, $\rho_4 = 0.5$. The four redundant variables are sampled from

$$\mathbf{x}_n^{[4-7]} \sim N\left(\mathbf{x}_n^{[1,3]} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & -2 & 2 & 1 \end{pmatrix}; \mathbf{I}_4\right)$$

while the 9 independent ones are simulated from $\mathbf{x}_n^{[8-16]} \sim N(\gamma, \delta)$ with

$$\gamma = (-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2)$$

and

$$\delta = diag(0.5, 0.75, 1, 1.25, 1.5, 1.25, 1, 0.75, 0.5).$$

A total of $B = 100$ Monte Carlo (MC) experiments are conducted as follows. From the DGP outlined above, $N = 500$ units are generated and their group membership retained for constructing the training set; while $M = 5000$ unlabeled observations compose the test set. Subsequently, label noise is simulated by wrongly assigning 20 units coming from the fourth group to the third class. In addition, 5 uniformly distributed outliers, having squared Mahalanobis distances from $\boldsymbol{\mu}_g$ greater than $\chi^2_{3,0.975}$ $\forall g \in \{1, 2, 3, 4\}$, are appended to the training set, with randomly assigned labels. These contaminations produce, in each MC replication, a total of 25 adulterated units, that account for slightly less than 5% of the entire learning set. In the upcoming Section, we validate the performance of our novel methods in correctly retrieving the relevant variables, compared to non-robust procedures. Particularly, the comparison is carried out considering the following methods:

- TBIC: robust stepwise greedy-forward approach via TBIC (Section 4.2.1)
- ML subset: maximum likelihood subset selector approach (Section 4.2.2), with subset size of relevant variables p equal to 3, 6 and 9
- SRUW: stepwise greedy-forward approach via BIC (Maugis et al., 2011)
- SelvarMix: variable selection in model-based discriminant analysis with a regularization approach (Celeux et al., 2019).

Furthermore, once the important variables have been identified, the associated classifier (i.e., REDDA for the robust variable selection criteria and EDDA for the non-robust ones) is trained on the reduced set of predictors and the classification accuracy is computed on the test set. A labeled trimming level α_l equal to 0.05 was kept fixed during the experiment. Lastly, for providing benchmark values on the relevance of feature selection, both EDDA and REDDA classifiers are also fitted on the original set with $P = 16$ variables. Simulation results are presented in the next Section.

4.3.2 Simulation results

Figure 4.2 graphically displays the proportion of times a variable has been selected as relevant by the different methods in the $B = 100$ repetition of the simulated experiment. As it is clearly visible from the plot, the first three features are selected by all the procedures in almost every repetition of the simulation study. The only exception is the SRUW model, for which the third variable is identified as relevant only 92 times out of 100. Generally therefore, the contamination introduced in the training set does not cause any systematic exclusion of the true discriminative variables from the relevant subset, also for the non-robust methods. Nonetheless, outliers and label noise lead SRUW and SelvarMix to severely overestimate the number of retained features. Redundant and irrelevant variables are often included in the selection, as demonstrated by the hollow triangles and diamonds in Figure 4.2. The robust stepwise approach via TBIC instead does not seem to suffer from this unfavorable behavior: it correctly identifies the first three relevant variables in every single simulation. As already pointed out in Section 4.2.3, the main drawback of the maximum likelihood subset selector approach is given

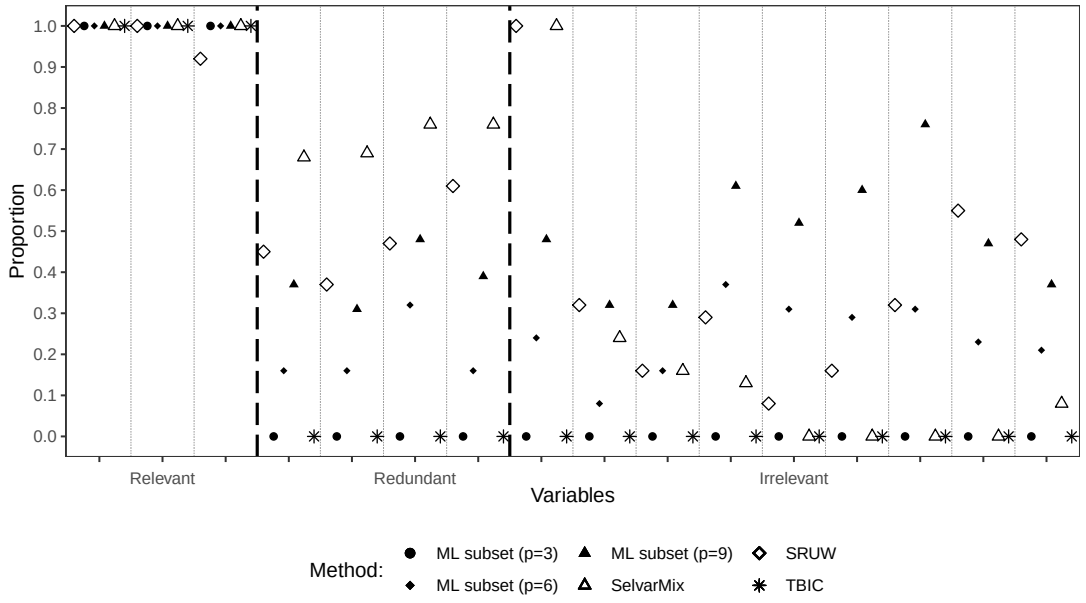

FIGURE 4.2: Proportion of times a variable has been selected as relevant, out of $B = 100$ MC repetition of the simulated experiment, for different variable selection methods.

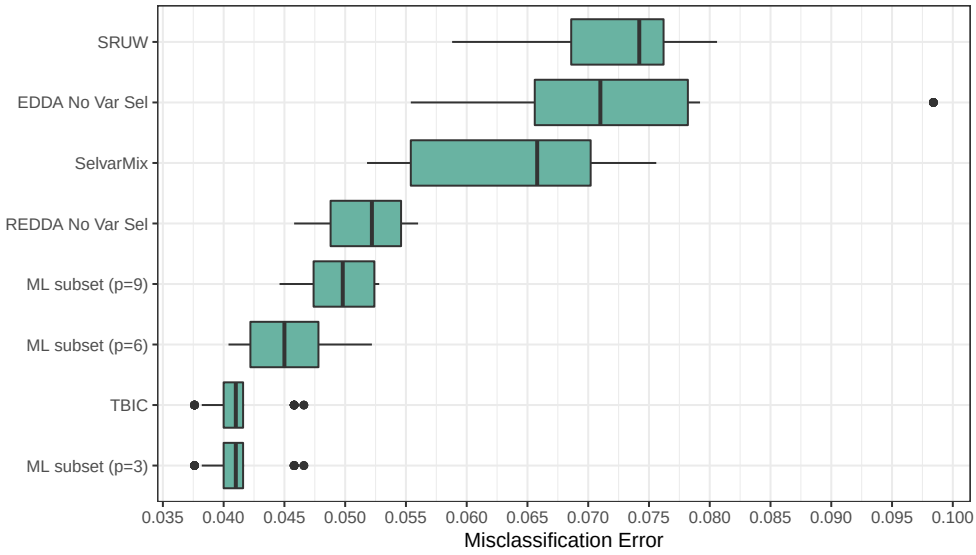

FIGURE 4.3: Boxplots of the misclassification error, out of $B = 100$ MC repetition of the simulated experiment, for the $M = 5000$ test data, varying variable selection and model-based classification methods.

by the need of pre-specifying the subset size p. When $p = 3$, i.e., the true number of discriminating variables, the algorithm always correctly selects the relevant ones. Clearly, when p is set higher than three, some irrelevant and/or redundant features will be necessarily included in the retained set. However, letting p to be greater than the true relevant predictors does not seem to severely affect the predictive power of the robust classification rule. As it can be seen from the results reported in Table 4.1 and in Figure 4.3, the misclassification errors are only slightly influenced by the choice of p in the ML subset selector, and are always lower than non-robust procedures. As expected, the best prediction accuracy is obtained when $p = 3$, result that entirely agrees with the one obtained by the forward selection algorithm via TBIC, as the very same variables are selected for each simulation and, subsequently, the REDDA classifier is fitted on the retained subset. Interestingly, the EDDA classifier coupled with (non-robust) vari-

TABLE 4.1: Average misclassification error, out of $B = 100$ MC repetition of the simulated experiment, for the $M = 5000$ test data, varying variable selection and model-based classification methods. Standard deviations reported in parentheses.

Method	Misclassification Error	Method	Misclassification Error
ML subset (p=3)	0.0409 (0.0026)	REDDA No Var Sel	0.051 (0.0026)
ML subset (p=6)	0.0455 (0.0037)	SRUW	0.072 (0.0037)
ML subset (p=9)	0.0493 (0.0028)	SelvarMix	0.0639 (0.0028)
TBIC	0.0409 (0.0026)	EDDA No Var Sel	0.073 (0.0026)

able selection via either SelvarMix or SRUW shows on average higher misclassification error than REDDA learned on the entire set of features. That is, the harmful effect of adulterated observations is increased by the presence of noisy variables, also shown by the poor performance of EDDA with no feature selection.
The present simulation study highlights how a very small percentage of attribute and class noise may somewhat spoil a wrapper procedure, driving the algorithm to include many more features than the truly relevant ones. That is, when adulterated units are not properly dealt with, both feature identification and classification may provide inappropriate results, with bias in the former propagating to badly affect the derived classifier even further. Therefore, replacing standard methods with robust solutions seem paramount whenever it is believed the considered dataset may contain some noisy units, especially in high dimensional settings.

4.4 Application to MIR spectra: starches discrimination

Chemometrics is a natural field of application for high-dimensional statistics, as data recorded from chemical systems are complex in nature and generally limited in terms of sample size. In particular, variable selection methods are notably appealing for observations recorded by spectroscopic instruments: for virtually continuous spectra the information contained in adjacent features is often correlated, and thus the determination of a relevant subset of wavelengths is desirable prior to perform any subsequent analysis (Brown, 1992; Brenchley et al., 1997). Furthermore, data reduction simplifies results interpretation, making future measurements simpler and cheaper (Indahl and Næs, 2004).

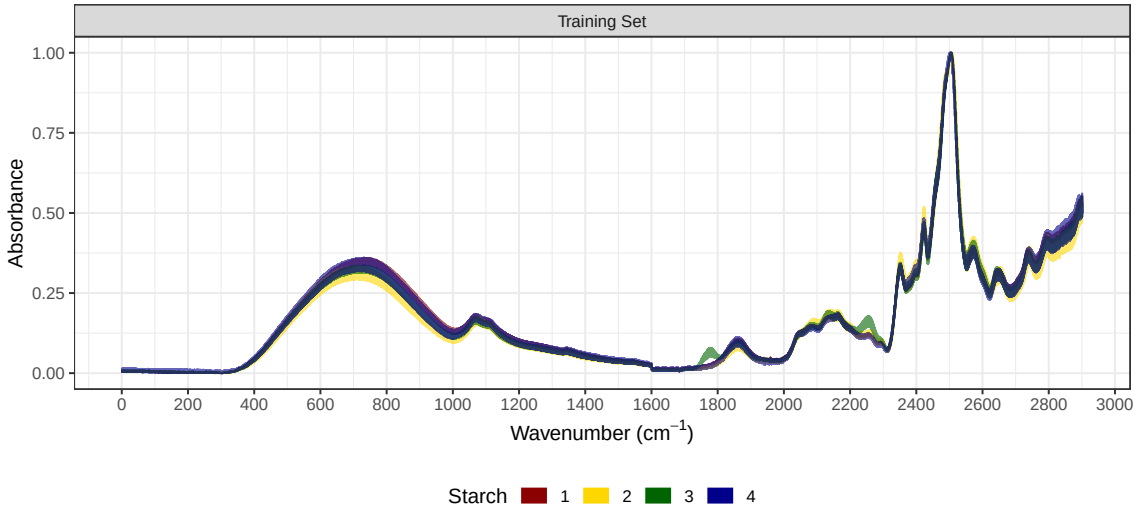

FIGURE 4.4: Midinfrared spectra of starches of four different classes, training set.

Spectroscopic data are recorded during controlled experiment, and the quality of both measurements and analysed substances is, in most cases, reliable. Nevertheless, calibration errors may appear during spectra collection, and, moreover, for some delicate applications such as food authenticity, the raw material itself may be spoiled and/or adulterated (Reid et al., 2006). In this context therefore, variable selection methods that not only robustly identify relevant wavelengths, but also recognize outliers and possibly fraudulent samples may be particularly valuable to chemometricians. Motivated by a Mid-infrared (MIR) dataset of the chemometric challenge organized during the 'Chimiométrie 2005' conference, the methodologies introduced in Section 4.2 are employed for performing high-dimensional classification and outlier detection.

4.4.1 Data

The considered datasets, described in Fernández Pierna and Dardenne, 2007, include respectively $N = 215$ (training set) and $M = 43$ (test set) MIR spectra of starches of four different classes, taken on a Perkin-Elmer Spectrum 2000 FTIR spectrometer (Perkin Elmer Corporation, Norwalk, CT, USA) between 4000 and 600 cm^{-1} at 1 cm^{-1} data interval, for a total of $P = 2901$ absorbance measurements for each sample. A subset of the learning observations is displayed in Figure 4.4. In order to create an extra difficulty to be tackled by the participants during the competition, four outliers were included in the test set:

- *Sample 2:* a shifted version of unit 1, obtained by removing its first six data points and appending six new variables at the end of the spectrum;
- *Sample 4:* a noisy version of unit 2, by generating Gaussian white noise and multiplying it by the absorbance values of the sample;
- *Sample 43:* a modified version of unit 39, obtained by manually changing a data point on the spectrum (wavelength 2456) to simulate a spike;
- *Sample 20:* a modified version of unit 17, by adding a slope to its original spectrum.

Therefore, the discrimination challenge held during 'Chimiométrie 2005' consisted in learning a classification rule from the training set to predict the labels of the test units, whilst also performing adulteration detection on the latter. In our experiment, we additionally include label

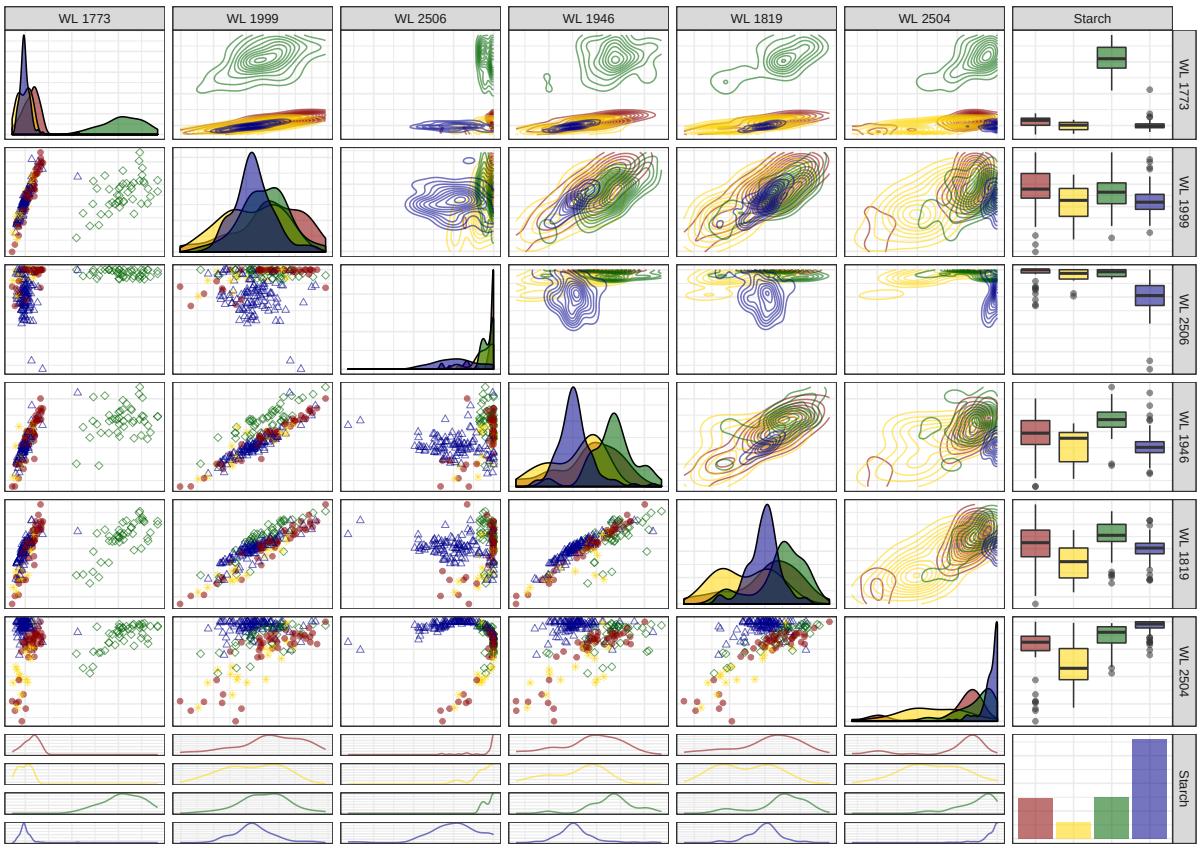

FIGURE 4.5: Generalized pairs plot of the relevant variables selected by the stepwise greedy-forward approach via TBIC. Starches dataset, training samples.

noise by wrongly assigning the last four units of the third group of starches to the fourth one: this accounts for less than 2% of the entire training set. Classification results are reported in the next Section.

4.4.2 Results

The discriminating problem described in the previous Section cannot be solved by directly applying model-based classifiers, since $N \ll P$. For overcoming this issue, we make use of the robust wrapper variable selection methods introduced in this chapter: such approaches provide a natural solution for dealing with contaminated high-dimensional data, and, as we will see, they can be further used to identify the noisy units in the test set. We firstly run the stepwise greedy-forward approach via TBIC (Section 4.2.1) with $\alpha_l = 0.05$: the procedure, out of $P = 2901$, selects a total of only six relevant wavelengths: 1773, 1999, 2506, 1946, 1819 and 2504. Figure 4.5 displays the generalized pairs plot for the selected variables. Motivated by the TBIC output and by the results presented in the Simulation Study, we decided to retain a slightly higher number of relevant variables in the ML subset selector, setting the value of p to be equal to 9. In doing so, the ML subset selector estimates the relevant subset F to be comprised of the following wavelengths: 1747, 1790, 1854, 1936, 2190, 2246, 2278, 2412 and 2503. A generalized pairs plot of such subset is reported in Figure 4.6. Interestingly, the two approaches select entirely different wavelengths as to be the most discriminative ones. Careful investigation of this behavior shows high correlation between the variables selected by the two methodologies, while the correlation reported by features within the same subset is much

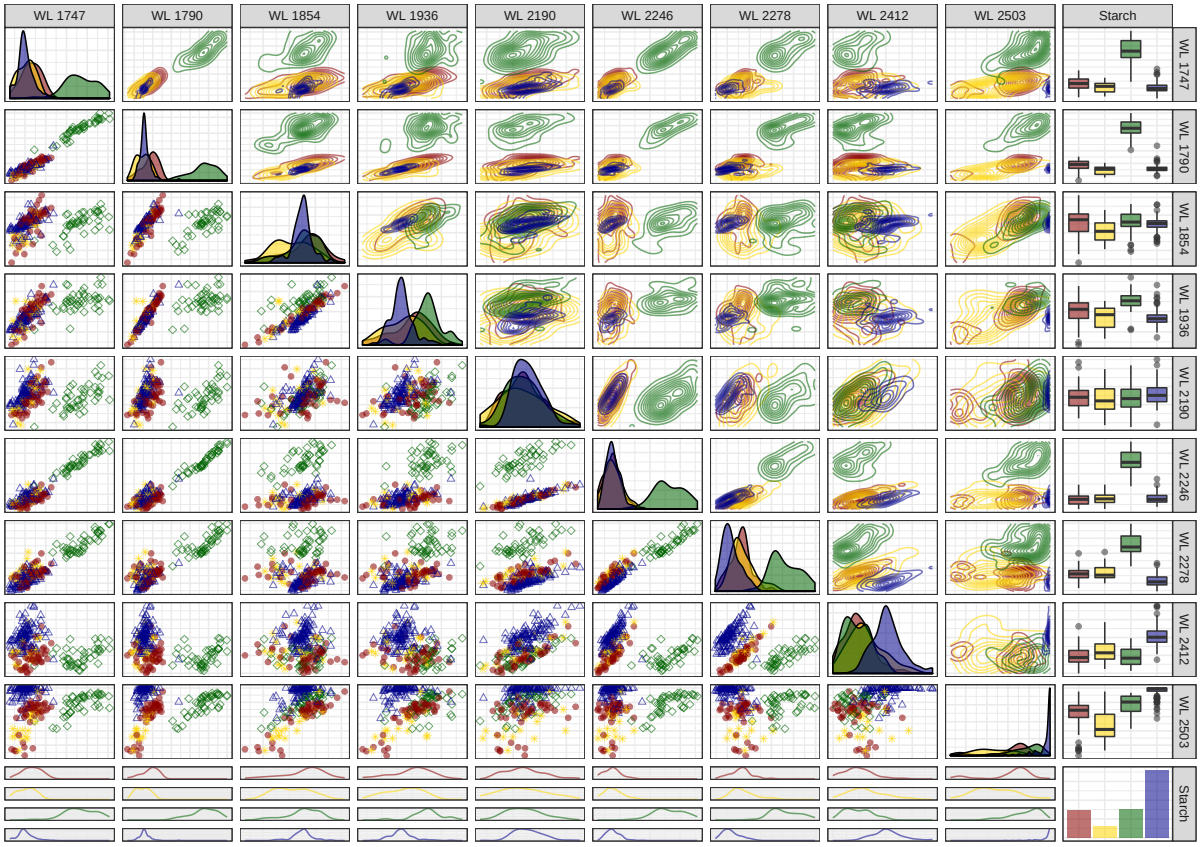

FIGURE 4.6: Generalized pairs plot of the relevant variables selected by ML subset selector with $p = 9$. Starches dataset, training samples.

lower. Clearly, in dealing with real datasets the separation between relevant, irrelevant and redundant variables is much less apparent. Particularly for spectroscopic data, highly correlated wavelengths often result in very comparable discriminating power, with no natural preference in terms of relevance. Nevertheless, it is worth noting that both methods chose wavelengths from the right-hand side part of the spectrum, as it seems to delineate the highest separation between the different starches, also by visual inspection of Figure 4.4.
A REDDA model with $\alpha_l = 0.05$ is employed to predict the class for the test samples, using as predictors the variables retained by the TBIC and ML subset selector, respectively. In both cases, units that present class noise in the training set were correctly identified as such and not accounted for in the estimation procedure. In addition, a Support Vector Machine with Gaussian radial kernel (SVM) was also considered, as it was shown to be the best performing classifier for this specific dataset (Fernández Pierna et al., 2005; Fernández Pierna and Dardenne, 2007). Lastly, we replicate the second best solution proposed by one of the participants: an ensemble method was constructed by combining ROC, PLS and SVM predictions via majority vote on a subset of variables, previously determined by a PLS model. Classification accuracy for the four competing methods, considering test sets with and without modified units, is reported in Table 4.2: the robust model-based classifiers show better results than the other

TABLE 4.2: Number of correctly predicted test samples and associated misclassification error for different methods. The test set with and without outliers has a total sample size of $M = 43$ and $M = 39$, respectively.

	REDDA (TBIC)	REDDA (ML subset)	SVM radial kernel	ROC+PLS+SVM
With outliers				
# correctly predicted	34	36	32	33
% correctly predicted	0.791	0.837	0.744	0.767
Without outliers				
#correctly predicted	32	34	31	31
% correctly predicted	0.821	0.872	0.795	0.795

solutions. The performance of the kernel and ensemble methods are negatively impacted by the presence of the 4 mislabelled units in the training set: compare results in Table 4.2 with the ones reported in Table 1 of Fernández Pierna and Dardenne, 2007, wherein the classifiers were trained on an uncontaminated learning set. The relevant subsets retained by both robust variable selection methods lead to similar results in terms of classification accuracy, with a slight better performance when REDDA is fitted on the features identified by the ML subset selector approach. As already pointed out in Fernández Pierna and Dardenne, 2007, the main source of errors is due to the difficulties in separating classes 1 and 2, as it is evident also from Figure 4.5 and 4.6.
We mentioned at the beginning of the Section that the REDDA method can be effectively employed in performing outlier detection in the test set. Particularly, given the probabilistic assumptions that underlie the methodology, for each test unit $\mathbf{y}_m$, $m = 1, \ldots, M$, we can compute its estimated marginal density as follows:

$$\hat{p}(\mathbf{y}_{m,\hat{F}}; \hat{\tau}, \hat{\boldsymbol{\mu}}_{\hat{F}}, \hat{\boldsymbol{\Sigma}}_{\hat{F}}) = \sum_{g=1}^{G} \hat{\tau}_g \phi \left(\mathbf{y}_{m,\hat{F}}; \hat{\boldsymbol{\mu}}_{g,\hat{F}}, \hat{\boldsymbol{\Sigma}}_{g,\hat{F}} \right) \tag{4.17}$$

where $\hat{F}$ denotes either the relevant variables identified by the stepwise approach with TBIC or by the ML subset selector, with parameters robustly estimated via the REDDA model on the

retained features. For both variable selection approaches, the 3 observations $\mathbf{y}_m$ with lowest value of (4.17) are units 2, 4 and 20; all of them were manually modified, as described in Section 4.4.1. The only neglected outlier is unit 43: it was contaminated on a single wavelength that was not identified as relevant by the variable selection methods. Nonetheless, by using an impartial trimming approach, we are effectively able to identify 3 out of 4 adulterated units.
In this Section, we have shown that the proposed noise-resistant variable selection approaches, coupled with robust discriminant analysis, can be effectively employed in performing high-dimensional classification in an adulterated framework. Even though being notably noise tolerant, powerful classifiers such as Support Vector Machine provide lower classification accuracy when a small percentage of class noise is present in the training set. In addition, after parameters have been robustly estimated, our proposal can be used to recognize possible adulterated units in the test set. All in all, an automatic methodology that performs robust feature detection, parameters estimation and outlier identification may become beneficial in chemometrics, easing both pre and post processing steps of complex spectroscopic analyses.

4.5 Concluding remarks

In the present chapter we have introduced two wrapper variable selection methods, resistant to outliers and label noise. We have shown that by means of these approaches we can effectively perform high-dimensional discrimination in an adulterated scenario. The first wrapper method embeds a robust model-based classifier within a greedy-forward algorithm, validating stepwise inclusion and exclusion of variables from the relevant subset via a robust information criterion. Theoretical justification that corroborates the procedure is also discussed. The second wrapper method resorts to the theory of maximum likelihood and irrelevance, defining an objective function in which the subset of relevant variables is regarded as a parameter to be estimated. A dedicated algorithm for MLE within a Gaussian family of patterned models has been developed, and practical implementation issues have been considered. Further, pros and cons of the two novel procedures have been discussed. A simulation study has been developed for assessing the effectiveness of our proposals in recovering the true discriminative features in a contaminated scenario, comparing their performances against well-known variable selection criteria. The novel methods have then been successfully applied in solving a high-dimensional classification problem of contaminated spectroscopic data. High discriminating power has been exhibited by the final models, whence the identification of the wrongly labeled and/or adulterated observations is derived as a by-product of the estimation procedures.
An open point for further research regards the extension of the fully supervised framework outlined here to the adaptive one, embedding the semi-supervised procedure introduced in Chapter 3 within a robust variable selection approach. In addition, careful investigation will be devoted to the development of a methodology that automatically assesses the contamination rate present in a sample, as the a-priori specification of the trimming level still remains an open issue in this field, particularly delicate for high-dimensional data.

4.6 Appendix A

In this Section we retrieve the ML estimates for the grouping and no grouping structures in the robust stepwise greedy-forward approach (Section 4.2.1), by means of the spurious outliers model specification.

4.6.1 Grouping model

The log-likelihood function of the spurious outliers model under the grouping structure is:

$$\begin{aligned}\ell(\mathcal{N}, \boldsymbol{\tau}^{cp}, \boldsymbol{\mu}^{cp}, \boldsymbol{\Sigma}^{cp}) = \sum_{n=1}^{N} q_n \sum_{g=1}^{G} \mathrm{l}_{ng} \log\left(\tau_g^{cp} \phi(\mathbf{x}_n^c, x_n^p; \boldsymbol{\mu}_g^{cp}, \boldsymbol{\Sigma}_g^{cp})\right) + \\ + \sum_{n=1}^{N} (1 - q_n) \log w(\mathbf{x}_n^c, x_n^p, \mathbf{l}_n; \boldsymbol{\psi}_n)\end{aligned} \tag{4.18}$$

to be maximized with respect to $\{\mathcal{N}, \boldsymbol{\tau}^{cp}, \boldsymbol{\mu}^{cp}, \boldsymbol{\Sigma}^{cp}\}$. The problem then reads:

$$\begin{aligned}\max_{\mathcal{N} \in \mathcal{D}(N)} \Bigg[\max_{\boldsymbol{\tau}^{cp}, \boldsymbol{\mu}^{cp}, \boldsymbol{\Sigma}^{cp}} \sum_{n=1}^{N} q_n \sum_{g=1}^{G} \mathrm{l}_{ng} \log\left(\tau_g^{cp} \phi(\mathbf{x}_n^c, x_n^p; \boldsymbol{\mu}_g^{cp}, \boldsymbol{\Sigma}_g^{cp}\right) + \\ + \max_{\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_N} \sum_{n=1}^{N} (1 - q_n) \log w(\mathbf{x}_n^c, x_n^p, \mathbf{l}_n; \boldsymbol{\psi}_n) \Bigg].\end{aligned} \tag{4.19}$$

By property (4.10), any configuration that maximizes the first addend in (4.19) also maximizes the second one. For a fixed partition $\mathcal{N} \in \mathcal{D}(N)$, the MLE for the first quantity are given by:

$$\hat{\tau}_g^{cp} = \frac{\sum_{n=1}^{N} q_n l_{ng}}{\lceil N(1 - \alpha_l) \rceil} \quad g = 1, \dots, G$$

$$\hat{\boldsymbol{\mu}}_g^{cp} = \frac{\sum_{n=1}^{N} q_n l_{ng} (\mathbf{x}_n^c, x_n^p)}{\sum_{n=1}^{N} q_n l_{ng}} \quad g = 1, \dots, G.$$

Estimation of $\boldsymbol{\Sigma}_g^{cp}$ depends on the considered patterned model, details are given in Bensmail and Celeux, 1996. Operatively, the final estimates are obtained via a REDDA model fitted on $\mathbf{x}_n^c, x_n^p$, see Section 2.2.

4.6.2 No grouping model

The log-likelihood function of the spurious outliers model under the no grouping structure is:

$$\begin{aligned}\ell(\mathcal{D}, \boldsymbol{\tau}^c, \boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c, \alpha, \boldsymbol{\beta}, \sigma^2) = \sum_{n=1}^{N} q_n \sum_{g=1}^{G} \mathrm{l}_{ng} \log\left[\tau_g^c \phi(\mathbf{x}_n^c; \boldsymbol{\mu}_g^c, \boldsymbol{\Sigma}_g^c)\right] + \\ + \sum_{n=1}^{N} q_n \log\left[\phi(x_n^p; \alpha + \boldsymbol{\beta}' \mathbf{x}_n^r, \sigma^2)\right] + \\ + \sum_{n=1}^{N} (1 - q_n) \log w(\mathbf{x}_n^c, x_n^p; \boldsymbol{\psi}_n)\end{aligned} \tag{4.20}$$

to be maximized with respect to $\{\mathcal{N}, \boldsymbol{\tau}^c, \boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c, \alpha, \boldsymbol{\beta}, \sigma^2\}$. The problem then reads:

$$\begin{aligned}\max_{\mathcal{N}\in\mathcal{D}(N)} \Bigg[& \max_{\boldsymbol{\tau}^c,\boldsymbol{\mu}^c,\boldsymbol{\Sigma}^c} \sum_{n=1}^{N} q_n \sum_{g=1}^{G} \mathrm{l}_{ng} \log\left[\tau_g^c \phi(\mathbf{x}_n^c; \boldsymbol{\mu}_g^c, \boldsymbol{\Sigma}_g^c)\right] + \\ & + \max_{\alpha,\boldsymbol{\beta},\sigma^2} \sum_{n=1}^{N} q_n \log\left[\phi(x_n^p; \alpha + \boldsymbol{\beta}' \mathbf{x}_n^r, \sigma^2)\right] + \\ & + \max_{\boldsymbol{\psi}_1,\dots,\boldsymbol{\psi}_N} \sum_{n=1}^{N} (1-q_n) \log w(\mathbf{x}_n^c, x_n^p, \mathbf{l}_n; \boldsymbol{\psi}_n) \Bigg].\end{aligned} \tag{4.21}$$

By property (4.10), any configuration that maximizes the sum of the first and second addend in (4.21) also maximizes the third one. For a fixed partition $\mathcal{N} \in \mathcal{D}(N)$, the first two quantities can be separately maximized, leading to the following MLE

$$\hat{\tau}_g^c = \frac{\sum_{n=1}^{N} q_n l_{ng}}{\lceil N(1-\alpha_l) \rceil} \quad g = 1, \dots, G$$

$$\hat{\boldsymbol{\mu}}_g^c = \frac{\sum_{n=1}^{N} q_n l_{ng} \mathbf{x}_n^c}{\sum_{n=1}^{N} q_n l_{ng}} \quad g = 1, \dots, G.$$

for the former addend, where as usual $\hat{\boldsymbol{\Sigma}}_g^c$ depends on the considered patterned model. ML estimates for the regression coefficients are obtained solving the following minimization problem:

$$\min_{\alpha,\boldsymbol{\beta}} \sum_{n=1}^{N} q_n (x_n^p - \alpha - \boldsymbol{\beta}' \mathbf{x}_n^r)^2 \tag{4.22}$$

which is very similar to the least trimmed squares method (Rousseeuw, 1984). Lastly, the variance is estimated as follows:

$$\hat{\sigma}^2 = \frac{1}{\lceil N(1-\alpha_l) \rceil} \sum_{n=1}^{N} q_n (x_n^p - \hat{\alpha} - \hat{\boldsymbol{\beta}}' \mathbf{x}_n^r).$$

Operatively, the MLE for (4.20) are obtained combining a REDDA model on $\mathbf{x}_n^c$ with a robust linear regression of x_n^p on $\mathbf{x}_n^r$. The discriminating function in (4.7) is used to determine the subset of untrimmed units on which to compute the estimates defined above, iterating the algorithm until the same observations are discarded in two consecutive steps. Lastly, at each iteration, the subset of variables $\mathbf{x}_n^r$ is determined with the `bicreg` function in the `BMA R` package (Raftery et al., 2018).

4.7 Appendix B

This final Section discusses the computational details of the algorithm used for fitting the ML subset selector, whose main steps are reported in Section 4.2.2. For achieving flexibility, parsimony and computational speed, the family of patterned models based on the eigenvalue decomposition in (1.13) of Bensmail and Celeux, 1996 is considered. Let us further introduce the following notations: for a $d \times d$ matrix $\boldsymbol{A}$, $diag(\boldsymbol{A})$ denotes the $d \times d$ diagonal matrix whose diagonal entries are the same of the matrix $\boldsymbol{A}$. Lastly, $\boldsymbol{A}(i, j)$ denotes the scalar entry at the ith row and jth column of the matrix $\boldsymbol{A}$.

4.7.1 Computational details on the M-step

As previously mentioned, we refer the reader to Bensmail and Celeux, 1996 for a complete treatment on the estimation of $\boldsymbol{\Sigma}_g$, $g = 1, \ldots, G$ under the 14 covariance structures. Conditioning on the chosen model, the estimation of the pooled covariance matrix $\boldsymbol{\Sigma}$ has the following form:

- Ellipsodial:
$$\hat{\boldsymbol{\Sigma}}_{ell} = \frac{1}{\lceil N(1-\alpha_l) \rceil} \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \left[(\mathbf{x}_n - \hat{\boldsymbol{\mu}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}})' \right]$$
for EEE, VEE, EVE, EEV, VVE, VEV, EVV and VVV models
- Diagonal:
$$\hat{\boldsymbol{\Sigma}}_{diag} = diag(\hat{\boldsymbol{\Sigma}}_{ell})$$
for EEI, VEI, EVI, VVI models.
- Spherical:
$$\hat{\boldsymbol{\Sigma}} = \frac{1}{P} \sum_{d=1}^{P} \hat{\boldsymbol{\Sigma}}_{diag}(d,d) \boldsymbol{I}_P$$
for EII, VII models

4.7.2 Computational details on the S-step

The S-step involves a discrete structure optimization, where we seek to determine the set of p variables that minimizes (4.16). Solving the problem by exhaustive enumeration is feasible only when $\binom{P}{p}$ is not too large, sadly it is rarely the case in a high-dimensional setting. Thus, the considered implementation relies on a stochastic algorithm for fixed-size subset selection, by means of the `kofnGA R` package (Wolters, 2015). Nonetheless, for specific patterned structures, simpler form of the objective function may be derived: see the following sections.

EEE model

For the homoscedastic model (EEE), (4.16) simplifies as follows:

$$h(F) = \log\det \hat{\boldsymbol{\Sigma}}_{EEE,F} - \log\det \hat{\boldsymbol{\Sigma}}_{ell,F} \tag{4.23}$$

where

$$\hat{\boldsymbol{\Sigma}}_{EEE,F} = \frac{1}{\lceil N(1-\alpha_l) \rceil} \sum_{g=1}^{G} \hat{n}_g \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \left[(\mathbf{x}_{n,F} - \hat{\boldsymbol{\mu}}_{g,F})(\mathbf{x}_{n,F} - \hat{\boldsymbol{\mu}}_{g,F})' \right]$$

and

$$\hat{n}_g = \frac{\sum_{n=1}^{N} \zeta(\mathbf{x}_n) l_{ng}}{\lceil N(1-\alpha_l) \rceil}.$$

It is nevertheless computationally efficient to derive $\hat{\boldsymbol{\Sigma}}_{EEE}$ for the full dimension P at once and to extract the sub-matrix $\hat{\boldsymbol{\Sigma}}_{EEE,F}$ when needed.

VVI model

For the heteroscedastic diagonal model (VVI), (4.16) simplifies to:

$$h(F) = \sum_{k \in F} \sum_{g=1}^{G} \hat{\tau}_g \log \frac{\hat{\mathbf{\Sigma}}_g(k,k)}{\hat{\mathbf{\Sigma}}_{diag}(k,k)} \tag{4.24}$$

for which $\hat{F}$ is the set of the indices k with the p smallest sums $\sum_{g=1}^{G} \hat{\tau}_g \log \frac{\hat{\mathbf{\Sigma}}_g(k,k)}{\hat{\mathbf{\Sigma}}_{diag}(k,k)}$.

EEI model

For the homoscedastic diagonal model (EEI), (4.16) reads:

$$h(F) = \sum_{k \in F} \log \frac{\hat{\mathbf{\Sigma}}_{EEI}(k,k)}{\hat{\mathbf{\Sigma}}_{diag}(k,k)} \tag{4.25}$$

with

$$\hat{\mathbf{\Sigma}}_{EEI} = \frac{1}{\lceil N(1-\alpha_l) \rceil} \sum_{g=1}^{G} \hat{n}_g diag \left(\sum_{n=1}^{N} \zeta(\mathbf{x}_n) \left[(\mathbf{x}_{n,F} - \hat{\boldsymbol{\mu}}_{g,F})(\mathbf{x}_{n,F} - \hat{\boldsymbol{\mu}}_{g,F})' \right] \right).$$

In this case, $\hat{F}$ is the set of the indices k with p smallest quotients $\frac{\hat{\mathbf{\Sigma}}_{EEI}(k,k)}{\hat{\mathbf{\Sigma}}_{diag}(k,k)}$.

4.7.3 Computational details on the T-step

When the full dimension P is large, it may occur that $\hat{\mathbf{\Sigma}}_{\hat{E}|\hat{F}}$ is not of full rank. In this case, it is still possible to estimate a singular normal distribution on a subspace of the set $\hat{E}$ of irrelevant variables. The associated density will then be:

$$\frac{(2\pi)^{-k/2}}{\left(\prod_{k=1}^{K} \omega_k\right)^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x}_{n,\hat{E}} - \hat{\mathbf{G}}_{\hat{E}|\hat{F}} \mathbf{x}_{n,\hat{F}} - \hat{\boldsymbol{\mu}}_{\hat{E}|\hat{F}})' \hat{\mathbf{\Sigma}}^{-}_{\hat{E}|\hat{F}} (\mathbf{x}_{n,\hat{E}} - \hat{\mathbf{G}}_{\hat{E}|\hat{F}} \mathbf{x}_{n,\hat{F}} - \hat{\boldsymbol{\mu}}_{\hat{E}|\hat{F}}) \right\} \tag{4.26}$$

where $\hat{\mathbf{\Sigma}}^{-}_{\hat{E}|\hat{F}}$ is the g-inverse of $\hat{\mathbf{\Sigma}}_{\hat{E}|\hat{F}}$ and $\omega_1, \ldots, \omega_K$ are the non-zero eigenvalues of $\hat{\mathbf{\Sigma}}_{\hat{E}|\hat{F}}$.

4.7.4 Models comparison

As a final remark, we mention the possibility of developing a procedure for automatically choosing the best model within the 14 parsimonious structures in the ML subset selector approach. One could rely on a BIC-like criterion (Schwarz, 1978), penalizing twice the final maximized trimmed log-likelihood by the number of estimated parameters and untrimmed observations, retaining the model that presents the highest value. However, this would rapidly increase the computational time needed for performing the analysis. For this reason, in both the simulation study and in the application, a VVV model only was considered when fitting the ML subset selector approach.

Chapter 5

Conclusions

The present manuscript has been devoted to investigating the effect that noise produces in supervised learning, thereupon proposing innovative solutions on how to effectively cope with it. The adopted methodology has been model-based discriminant analysis: a solid probabilistic framework for modeling heterogeneity in a population. In order to achieve robust parameters estimation and adulteration detection, an impartial trimming approach has been favored, wherein the least likely observations, according to the postulated model, are not included in the estimation process. Such procedure, coupled with constrained maximization by means of an eigenvalues-ratio constraint, has defined a well-posed mathematical problem for providing robust inference in a contaminated scenario. Such background concepts on model-based classification and robustness have been reviewed in Chapter 1.

For the remaining part of the manuscript, noise has been a comprehensive term for identifying any mechanism that obscures the relationship between attributes and class membership. Therefore, several robust models have been developed, depending on which type of noise they were meant to deal with. The methodologies in Chapter 2 have confronted with noise in the form of outliers and mislabeled units, both in the training and in the test set. In Chapter 3, the semi-supervised model of the previous chapter has been expanded to account for sample selection bias, where unobserved classes may have been present in the unlabeled set. In the context of high-dimensional data, only a subset of the recorded predictors might be useful in recovering the group structure, and the distinction between relevant and noisy variables plays a determinant role in the development of an efficient classifier. This has been the topic of Chapter 4, where noise has assumed the form of outliers, label noise and irrelevant features: two robust variable selection methods have been introduced for jointly tackling these issues.

A well-defined course for future research certainly involves the definition of a unified framework that includes the different models presented in this thesis. A thorough methodology for robust and adaptive high-dimensional classification via impartial trimming and constraint would be the most desirable output of the work carried out during the PhD program, accompanied by related statistical software. Hopefully, supported by the promising results obtained in the applications, such unified procedure will become valuable for practitioners that are regularly faced with complex analysis of contaminated data, as in the fields of food authenticity, metagenomics and chemometrics among others.

We conclude reporting a quote contained in the pioneering work of Andrews et al., 1972: "From the 1970s to 2000 we would see [...] extensions to linear models, time series, and multivariate models, and widespread adoption to the point where every statistical package would take the robust method as the default". Even though, after almost 50 years from the original *Princeton Robustness Study*, robust methods are not yet the default choice for a statistical analysis, they have experienced a tremendous growth in the past decades. With this manuscript, we have humbly intended to be a small part of this advancement, fostering the employment of robust techniques in engaging with wonders and pitfalls of the big data era.

Appendix A

Detecting wine adulterations via trimming

Based on:

Cappozzo, A., Greselin, F.
"Detecting wine adulterations employing robust mixture of Factor Analyzers"
Statistical Learning of Complex Data. CLADAG 2017. Studies in Classification, Data Analysis, and Knowledge Organization. (2019)
https://doi.org/10.1007/978-3-030-21140-0_2

A.1 Introduction and motivation

The wine segment is identified as a luxury market category, with savvy as well as non-expert customers willing to spend a premium price for a product of a specific vintage and cultivar. Therefore, in the context of global markets, analytical methods for wine identification are needed in order to protect wine quality and prevent its illegal adulteration.
In the present work we employ an approach based on robust estimation of mixtures of Gaussian Analyzers, for discriminating corrupted red wines samples from their authentic variety. In a modeling context, we assume a probability distribution function for the chemical and physical characteristics measured on the wines, considering a density in the form of a mixture, whenever the dataset presents more than a wine variety. As a consequence, the probability that a wine sample comes from a specific grape can be estimated from the model, performing classification through the Bayes rule. Robust estimation of the parameters in the model is adopted to recognize the corrupted data. Particularly, we expect that adulterated observations would be implausible under the robustly estimated model: the illegal subsample is revealed by selecting observations with the lowest contributions to the overall likelihood using impartial trimming, without imposing any assumption on their underlying density.
The rest of this appendix is organized as follows: in Section A.2 the notation is introduced and the main concepts about Gaussian Mixtures of Factor Analyzers (MFA), trimmed MFA likelihood and the Alternating Expectation - Conditional Maximization (AECM) algorithm are summarized. Section A.3 presents the *wine* dataset (Forina et al., 1986) and classification results obtained performing a robust estimation of Gaussian mixtures of factor analyzers . Section A.4 reports a simulation study carried out employing parameters estimated from the model in Section A.3, in a specific framework of contaminated dataset.

The original contribution of the present Appendix is given in the benchmark study on unsupervised methods, the adaptation of the robust Bayesian Information Criterion (BIC) introduced in Cerioli et al., 2018a to MFA, and a first application of robust MFA in a somehow realistic adulteration scenario.
An application on real data and some simulation results confirm the effectiveness of our approach in dealing with an adulterated dataset when compared to analogous methods, such as partition around medoids and non robust mixtures of Gaussian and mixtures of patterned Gaussian factors.

A.2 Mixtures of Gaussian Factors Analyzers

In this section we briefly recall the definition and some features of the mixture of Gaussian Factor Analyzers (MFA) and its parameters estimation procedure. MFA is a powerful tool for modeling unobserved heterogeneity in a population, as it concurrently performs clustering and local dimensionality reduction, within each cluster. Let $\mathbf{X}_1, \ldots, \mathbf{X}_N$ be a random sample of size N on a p-dimensional random vector. An MFA assumes that each observation $\mathbf{X}_n$ is given by

$$\mathbf{X}_n = \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \mathbf{U}_{ng} + \mathbf{e}_{ng} \tag{A.1}$$

with probability τ_g for $g = 1, \ldots, G$, with G total number of components in the mixture. $\boldsymbol{\mu}_g$ are $p \times 1$ mean vectors, $\boldsymbol{\Lambda}_g$ are the $p \times d$ matrices of *factor loadings*, $\mathbf{U}_{ng} \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ are the *factors*, $\mathbf{e}_{ng} \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}_g)$ are the *errors*, and $\boldsymbol{\Psi}_g$ are $p \times p$ diagonal matrices. Note that $d < p$, that is the p observable features are supposed to be jointly explained by a smaller number of d unobservable factors. Further, $\mathbf{U}_{ng}$ and $\mathbf{e}_{ng}$ are independent, for $n = 1, \ldots, N$ and $g = 1, \ldots G$. Unconditionally, therefore, $\mathbf{X}_n$ has a density in the form of a G-components multivariate normal mixture:

$$f_{\mathbf{X}_n}(\mathbf{x}_n; \boldsymbol{\theta}) = \sum_{g=1}^{G} \tau_g \phi_p(\mathbf{x}_n; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \tag{A.2}$$

where the covariance matrix $\boldsymbol{\Sigma}_g$ has the form $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$.
When estimating MFA through the usual Maximum Likelihood approach, two issues arise. Firstly, departure from normality in the data may cause biased or misleading inference. Some initial attempts in the literature to overcome this issue propose to consider mixtures of t-factor analyzers (McLachlan et al., 2007), but the breakdown properties of the estimators are not improved (Hennig, 2004). The second concern is related to the unboundedness of the log-likelihood function (Day, 1969), which leads to estimation issues as the appearance of non-interesting *spurious maximizers* and degenerate solutions. To cope with this second issue, Common/Isotropic noise matrices/patterned covariances (Baek et al., 2010) and a mild constrained estimation (Greselin and Ingrassia, 2015) have been considered. The methodology considered here employs model estimation, complemented with *trimming* and *constrained estimation*, to provide robustness, exclude singularities and reduce spurious solutions, along the lines of García-Escudero et al., 2016. Therefore, it overcomes both previously mentioned issues.
A mixture of Gaussian factor components is fitted to a given dataset $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ in $\mathbb{R}^p$ by maximizing a *trimmed mixture log-likelihood* (Neykov et al., 2007),

$$\mathcal{L}_{trim} = \sum_{n=1}^{N} \zeta(\mathbf{x}_n) \log \left[\sum_{g=1}^{G} \tau_g \phi_p(\mathbf{x}_n; \mu_g, \boldsymbol{\Lambda}_g, \boldsymbol{\Psi}_g) \right] \tag{A.3}$$

where $\zeta(\cdot)$ is a 0-1 trimming indicator function, that tells us whether observation $\mathbf{x}_n$ is trimmed off or not. If $\zeta(\mathbf{x}_n)$=0 $\mathbf{x}_n$ is trimmed off, otherwise $\zeta(\mathbf{x}_n)$=1. A fixed fraction α of observations, the *trimming level*, is unassigned by setting $\sum_{n=1}^{N} \zeta(\mathbf{x}_n) = \lceil N(1-\alpha) \rceil$, where the less plausible observations under the currently estimated model are tentatively trimmed out at each step of the iterations that lead to the final estimate. In the specific application in the framework of wine authenticity described in Section A.3, they are supposed to be originated by wine adulteration. Then, a constrained maximization of (A.3) is adopted, by imposing $\psi_{g,ll} \leq c\, \psi_{h,mm} \quad$ for $1 \leq l \neq m \leq p$ and $1 \leq g \neq h \leq G$, where $\{\psi_{g,ll}\}_{l=1,\ldots,p}$ are the diagonal element of the noise matrices $\mathbf{\Psi}_g$, and $1 \leq c < +\infty$, to avoid the $|\Sigma_g| \to 0$ case. This constraint can be seen as an adaptation to MFA of those introduced in Ingrassia, 2004: the ones used throughout the present manuscript. The Maximum Likelihood estimator of $\mathbf{\Psi}_g$ under the given constraints leads to a well-defined maximization problem.
The Alternating Expectation - Conditional Maximization - an extension of the Expectation-Maximization algorithm - is considered, in view of the factor structure of the model. The M-step is replaced by some computationally simpler conditional maximization (CM) steps, along with different specifications of missing data. The idea is to partition the vector of parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1', \boldsymbol{\theta}_2')'$, in such a way that $\mathcal{L}_{trim}$ is easy to be maximized for $\boldsymbol{\theta}_1$ given $\boldsymbol{\theta}_2$ and viceversa. Therefore, two cycles are performed at each algorithm iteration:
1^{st}*cycle* : we set $\boldsymbol{\theta}_1 = \{\tau_g, \boldsymbol{\mu}_g, g = 1, \ldots, G\}$, here the missing data are the unobserved group labels $\mathbf{Z} = (\mathbf{z}_1', \ldots, \mathbf{z}_N')$. After applying a step of Trimming, by assigning to the observations with lowest likelihood a null value of the "posterior probabilities", we get one E-step, and one CM-step for obtaining parameters in $\boldsymbol{\theta}_1$.
2^{nd}*cycle* : we set $\boldsymbol{\theta}_2 = \{\mathbf{\Lambda}_g, \mathbf{\Psi}_g, g = 1, \ldots, G\}$, here the missing data are the group labels $\mathbf{Z}$ and the unobserved latent factors $\mathbf{U}_{11}, \ldots, \mathbf{U}_{NG}$. We perform a Trimming step, then a E-step, and a constrained CM-step, i.e. a conditional exact constrained maximization of $\mathbf{\Lambda}_g, \mathbf{\Psi}_g$.
A detailed description of the algorithm is given in García-Escudero et al., 2016.

A.3 Wine recognition data

The wine recognition dataset, firstly reported and analysed in Forina et al., 1986, describes results of a chemical and physical analysis for three different wine types, grown in the same region in Italy. Originally, 28 attributes were recorded for 178 wine samples derived from three different cultivars: Barolo, Grignolino and Barbera. A reduced version of the original dataset with only thirteen variables is publicly available in the University of California, Irvine Machine Learning data repository, commonly used in testing the performance of newly introduced supervised and unsupervised classifiers. Particularly, in the unsupervised classification litera-

TABLE A.1: *RobustBIC* (Cerioli et al., 2018a) for different choices of the number of factors d and the number of groups G for the robust MFA model on wine data, trimming level $\alpha = 0.05$ and $c = 20$.

d / G	1	**2**	3
1	9082.58	8282.92	8223.46
2	8560.62	8107.62	8112.90
3	8352.26	8042.02	8199.38
4	8160.77	**7969.64**	8315.23
5	8102.77	8044.03	8456.00
6	8097.06	8165.67	8735.63

ture the wine recognition data has been considered to assess cluster analysis in information-theoretic terms via minimization of partition entropy (Roberts et al., 2000), to prove the modeling capabilities of a generalized Dirichlet mixture (Bouguila and Ziou, 2004), to evaluate the efficacy of employing distances based on non-Euclidean norms (Doherty et al., 2007) and Random Forest dissimilarity (Shi and Horvath, 2006). More recently, also parsimonious Gaussian mixture models have been applied to the Italian wines dataset (McNicholas and Murphy, 2008). Here our purpose is twofold: we want to explore the classification performance of a robust estimation based on mixtures of Gaussian Factors Analyzers, and we aim at obtaining realistic parameters for the subsequent simulation study. The dataset, available in the *pgmm* R package (McNicholas et al., 2011), contains 27 of the 28 original variables, since the sulphur measurements were not available. Initially, to perform model selection and detect the most suitable values of factors d and groups G, an adaptation to the MFA framework of the robust Bayesian Information Criterion, firstly introduced in Cerioli et al., 2018a, has been considered. That is, $BIC = -2\mathcal{L}_{trim}(x;\hat{\theta}) + v^c \log n^*$, where $v^c = (G - 1 + Gp + G(pd - d(d-1)/2) + (Gp - 1)(1 - 1/c) + 1)$ denotes the number of free parameters in the model (depending on the value of the constraint c) and $n^* = \lceil N(1-\alpha) \rceil$ the number of non trimmed observations. Robust BIC for different choices of the number of factors d and the number of groups G are reported in Table 1, considering a trimming level $\alpha = 0.05$ and $c = 20$. The value of the robust BIC is minimized

TABLE A.2: Classification table for the robust MFA with number of factors $d = 4$, number of groups $G = 3$, trimming level $\alpha = 0.05$ and $c = 20$ on the wine data. Trimmed observations are classified a-posteriori according to the Bayes rule.

	1	2	3
Barolo	59	0	0
Grignolino	0	71	0
Barbera	0	0	48

for $d = 4$ and $G = 2$, suggesting a mixture with just two components. Careful investigation on this result highlighted that robust MFA methodology tended to cluster together Barolo and Grignolino samples as arising from the same mixture component, while clearly separating Barbera observations. It is worth recalling (Forina et al., 1986) that the wines in this study were collected over the time period of 1970-1979, and the Barbera wines are predominantly from a later period than the Barolo or Grignolino wines. Therefore, considering the nature of the phenomena under study and the risks related to rigidly selecting the number of components in a mixture model only on the basis of the results provided by an information criteria, such as BIC (Lee and McLachlan, 2016), we decided to employ a robust MFA with $d = 4$, $G = 3$ and $\alpha = 0.05$, leading to the classification matrix reported in Table A.2. Employing a robust MFA rather than a Gaussian mixture leads to a 60% reduction in the number of parameters to be estimated (470 against 1217). Notice, in addition, that after robust estimation, also the trimmed observations can be a-posteriori classified according to the Bayes rule, i.e assigning each of them to the component g having greater value of $D_g(\mathbf{x}, \theta) = \tau_g \phi_p(\mathbf{x}; \mu_g, \mathbf{\Lambda}_g \mathbf{\Lambda}_g^{'} + \mathbf{\Psi}_g)$.
Results in Table A.2 show that the robust MFA algorithm led to a perfect clusterization of the samples according to their true wine type.
For completeness, the robust MFA algorithm is also applied to the more common thirteen variable subset of the wine data and comparison with the existing literature is reported in Table A.3. The clustering performance with respect to the true wine labels reports an *Adjusted Rand Index* equal to 0.98 with just one Grignolino sample wrongly assigned to the cluster identifying Barolo wines. Again then, the robust MFA methodology outperforms the results currently present in the literature for unsupervised learning on this specific dataset.

TABLE A.3: Comparison of performance metrics for different methodologies on the thirteen variable subset of the wine data. Reported metrics come from the original articles.

Methodology	Performance Metric	
	Class Recovery Accuracy	Adjusted Rand Index
Partition Entropy (Roberts et al., 2000)	0.977	-
Mixture of Generalized Dirichlet (Bouguila and Ziou, 2004)	0.978	-
Neural gas (Doherty et al., 2007)	0.954	-
Random Forest predictors (Shi and Horvath, 2006)	-	0.93
Parsimonious Gaussian Mixture (McNicholas and Murphy, 2008)	0.927	0.79
Robust MFA (García-Escudero et al., 2016)	0.994	0.98

A.4 Simulation Study

The purpose of this simulation study is to show the effectiveness of estimating a robust MFA on a set of observations drawn from two luxury wines, Barolo and Grignolino, and identifying units presenting an adulteration. Considering the parameters estimated obtained in Section A.3, the artificial dataset is generated simulating 100 observations each, from Barolo and Grignolino components. Afterwards, the "contamination" is created decreasing by 15% the values of Fixed Acidity, Tartaric Acid, Malic Acid, Uronic Acids, Potassium and Magnesium for 5 Barolo and for 5 Grignolino observations. This procedure resembles the illegal practice of adding water to wine (Jackson, 2008). The problem of distinguishing adulterated observations

TABLE A.4: Average misclassification errors and ARI (percent average values on 1000 runs)

	AECM	*pam*	*Mclust*	*pgmm*
Misclassification error	0.0309	0.2935	0.2073	0.2314
Adjusted Rand Index	0.9362	0.5466	0.7184	0.6959

from the real mixture components is addressed, together with the algorithm performance in correctly classifying the authentic units.

TABLE A.5: Bias and MSE (in parentheses) of the parameter estimators $\hat{\boldsymbol{\mu}}_g$ and $\hat{\boldsymbol{\Sigma}}_g$

	AECM	*Mclust*	*pam*		*AECM*	*Mclust*	*pgmm*
$\boldsymbol{\mu}_1$	-0.0019	-0.0194	0.0069	$\boldsymbol{\Sigma}_1$	0.0001	-0.001	0.0257
	(0.0029)	(0.0421)	(0.1022)		(0.0004)	(0.0022)	(0.0079)
$\boldsymbol{\mu}_2$	-0.0011	0.1522	-0.0025	$\boldsymbol{\Sigma}_2$	-0.0156	-0.0164	0.0113
	(0.0042)	(0.2376)	(0.1380)		(0.0043)	(0.0043)	(0.0077)

We estimate a robust MFA with $G = 2$, $p = 27$, $d = 4$ and trimming level $\alpha = 0.05$. We compare our results with other popular methods: Partition around medoids, Gaussian mixtures estimated via Mclust, and Mixtures of patterned Gaussian factors estimated by *pgmm*. To perform each of the $B = 1000$ simulations, algorithms have been initialized following the indications of their respective authors: say 10 random starts at each run of *AECM*, default setting for the "build phase" of *pam* as in Maechler et al., 2017, applying model-based hierarchical clustering

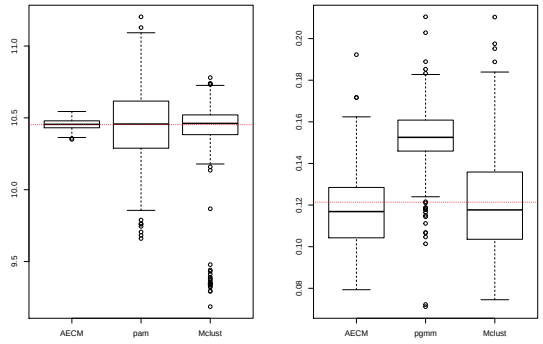

FIGURE A.1: Boxplots of the simulated distributions of $\hat{\mu}_1[1]$, estimator for $\mu_1[1] = 10.45$ (left panel); $\hat{\Sigma}_1[1,1]$, estimator for $\Sigma_1[1,1] = 0.1214$ (right panel).

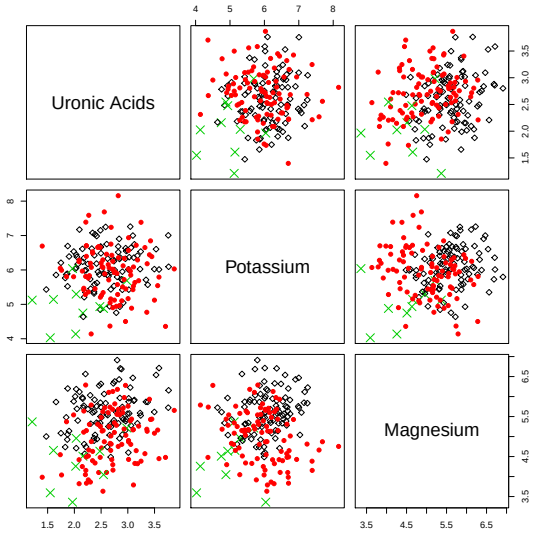

FIGURE A.2: Clustering of the simulated data with fitted trimmed and constrained MFA. Trimmed observations are denoted by "×".

as per default setting in Scrucca et al., 2016 for *Mclust* and 10 random starts at each run as suggested in McNicholas and Murphy, 2008 for *pgmm*.
Table A.4 reports the average misclassification error and Adjusted Rand Index: the AECM algorithm reports a superb classification rate, with smaller variability of the simulated distributions for the estimated quantities, as shown in Figure A.1.
For a fair comparison of the performance of the algorithms, we consider 3 clusters for *pam*, *Mclust* and *pgmm*; whereas we consider only 2 clusters for AECM, because in this approach the adulterated group should ideally be captured by the trimmed units. A value of $c = 20$ allows to discard singularities and to reduce spurious solutions (García-Escudero et al., 2016). The effects of the trimming procedure are shown in Figure A.2, where the different colors and shapes represent the obtained classification. Table A.5 reports the average bias and MSE for the mixture parameters (computed element-wise for every component).
The present simulations show initial promising results in adopting robust MFA as a tool for identifying wine adulteration. Contrarily to the methods developed in the main chapters of the thesis, here the problem is framed in a completely unsupervised scenario. Given that the topic was somehow less related to the main core of the manuscript, we decided to include it as an Appendix.

Appendix B

Code details

This appendix provides the listings for the main routines developed in the thesis. Particularly, Section B.1 reports the implementation used for enforcing the eigenvalues-ratio constraint in (1.20) under the different patterned models of Bensmail and Celeux, 1996. Such routines rely on the `.restr2_deter` and `.restr2_eigenv` functions of the `tclust R` package (Fritz et al., 2012). Sections B.2 and B.3 provide the EM algorithms for implementing the transductive and inductive estimation for the RAEDDA model, respectively. As highlighted in Section 3.2.7, the RAEDDA model is a generalization of the method developed in Chapter 2, therefore, the same routines may be used for fitting a RUPCLASS model, by simply setting $E = G$ in the transductive approach. Here, only a subset of the functions necessary to run the programs are reported. Nonetheless, the entire collection of (work in progress) `R` packages related to the present manuscript are publicly available at the author's github page: https://github.com/AndreaCappozzo.

B.1 Code for Appendix C (Section 2.8)

```
# Majorization Minimization Algorithm Browne 2014

MM1 <-
  function(D_0 = diag(dim(W)[1]),
           A,
           W,
           w,
           ctrl_restr) {
    D <- D_0
    F_obj <- Inf
    F_obj_old <- Inf
    iter <- 0
    criterion <- TRUE
    while (criterion) {
      iter <- iter + 1
      F_t <-
        matrix(apply(
          sapply(1:dim(W)[3], function(g)
            diag(1 / A[, g]) %*% t(D) %*% W[, , g] - w[g] *
             diag(1 / A[, g]) %*% t(D)),
          1,
          sum
        ), dim(W)[1], dim(W)[1], byrow = F)
```

```
      SVD_F <- svd(F_t)
      P <- SVD_F$u
      R <- SVD_F$v
      D <- R %*% t(P)
      F_obj <-
        sum(sapply(1:dim(W)[3], function(g)
          sum(diag(
            W[, , g] %*% D %*% diag(1 / A[, g]) %*% t(D)
          ))))
      criterion <-
        ((F_obj_old - F_obj) > ctrl_restr$MM_tol) &
        (iter < ctrl_restr$MM_max_iter)
      F_obj_old <- F_obj
    }
    D
  }

# Restrictions for EV_, VE_ and VV_ models ------
restr_2_EV_ <- function(fitm,
                        restr_factor,
                        extra_groups,
                        ctrl_restr) {
  eigenvalues <-
    exp(log(fitm$parameters$variance$shape[, extra_groups, drop = FALSE]) +
     log(fitm$parameters$variance$scale))
  if (ifelse(
    is.na(max(eigenvalues) / min(eigenvalues) <= restr_factor) |
    is.infinite(max(eigenvalues) / min(eigenvalues)),
    TRUE,
    max(eigenvalues) / min(eigenvalues) <= restr_factor
  )) {
    return(fitm)
  }
  iter <- 0
  while ((max(eigenvalues) / min(eigenvalues) -
   restr_factor > ctrl_restr$tol) &
         iter <= ctrl_restr$max_iter) {
    iter <- iter + 1
    eigenvalues <- restr2_eigenv(
      autovalues = eigenvalues,
      ni.ini = colSums(fitm$z[, extra_groups, drop = FALSE]),
      restr.fact = restr_factor,
      zero.tol = .Machine$double.xmin
    )

    #2nd step: .restr2_deter
    if (length(extra_groups) == fitm$G) {
      # Transductive approach
      eigenvalues <- restr2_deter_(
        autovalues = eigenvalues,
```

```
      ni.ini = colSums(fitm$z[, extra_groups, drop = FALSE]),
      #ni.ini = rep(1, fitm$G),
      restr.fact = 1,
      #I force the determinants to be equal
      zero.tol = .Machine$double.xmin
    )
  } else {
    # Inductive approach

    scale_extra_group <- apply(eigenvalues, 2, function(x)
      exp(1 / fitm$d * sum(log(x))))

    shape_extra_group <- exp(sweep(
      x = log(eigenvalues),
      2,
      log(scale_extra_group),
      FUN = "-"
    ))

    eigenvalues <- exp(sweep(
      x = log(shape_extra_group),
      2,
      log(fitm$parameters$variance$scale),
      # same volume of the known groups
      FUN = "+"
    ))
  }
}

scale_RESTR <- apply(eigenvalues, 2, function(x)
  exp(1 / fitm$d * sum(log(x))))[1]
shape_RESTR <- exp(log(eigenvalues) - log(scale_RESTR))

if (fitm$modelName == "EVE" &
    length(extra_groups) == fitm$G) {
  # Transductive approach
  W <-
    mclust::covw(fitm$X, Z = fitm$z, normalize = F)$W
    #sample scatter matrices for the un-trimmed obs
  E <-
    apply(W, 3, function(x)
      eigen(x, only.values = T)$val) #eigenvalues of W
  w <- apply(E, 2, max) #biggest eigenv of W_g, g=1, ..., G
  orientation_RESTR <-
    MM1(
      A = shape_RESTR,
      W = W,
      w = w,
      ctrl_restr = ctrl_restr
    )
```

```
    fitm$parameters$variance$orientation <- orientation_RESTR
  }
  #I update the variance components in the output of the M-step
  fitm$parameters$variance$shape[, extra_groups] <- shape_RESTR
  fitm$parameters$variance$scale <- scale_RESTR
  fitm$parameters$variance$sigma <- mclust::decomp2sigma(
    d = fitm$d,
    G = fitm$G,
    scale = fitm$parameters$variance$scale,
    shape = fitm$parameters$variance$shape,
    orientation = fitm$parameters$variance$orientation
  )
  return(fitm)
}

restr_2_VE_ <-
  function(fitm,
           restr_factor,
           extra_groups,
           ctrl_restr) {
    if (fitm$modelName == "VII") {
      fitm$parameters$variance <- SIGMA_COMP(fitm$parameters$variance)
    }
    eigenvalues <-
      fitm$parameters$variance$shape %o%
       fitm$parameters$variance$scale[extra_groups]

    if (ifelse(
      is.na(max(eigenvalues) / min(eigenvalues) <= restr_factor) |
      is.infinite(max(eigenvalues) / min(eigenvalues)),
      TRUE,
      max(eigenvalues) / min(eigenvalues) <= restr_factor
    )) {
      return(fitm)
    }

    scale_0 <-
      scale_RESTR <-
      fitm$parameters$variance$scale[extra_groups] # original scale
    shape_0 <-
      shape_RESTR <- fitm$parameters$variance$shape # original shape

    iter <- 0
    while ((max(eigenvalues) / min(eigenvalues) -
     restr_factor > ctrl_restr$tol) &
           iter <= ctrl_restr$max_iter) {
      iter <- iter + 1
      eigenvalues <-
        restr2_eigenv(
          autovalues = eigenvalues,
```

```
      ni.ini = colSums(fitm$z[, extra_groups, drop = FALSE]),
      # the groups size
      restr.fact = restr_factor,
      zero.tol = .Machine$double.xmin
    )

  scale_0 <- scale_RESTR
  shape_0 <- shape_RESTR

  if (length(extra_groups) == fitm$G) {
    # Transductive approach
    shape_RESTR <-
      apply(sapply(1:fitm$G, function(g)
        exp(
          log(eigenvalues[, g]) - log(scale_RESTR[g])
        )), 1, sum) /
      (exp(1 / fitm$d * sum(log((
        apply(
          sapply(1:fitm$G, function(g)
            eigenvalues[, g] / scale_RESTR[g]),
          1,
          sum
        )
      )))))
  } else {
    # Inductive approach
    shape_RESTR <- shape_0
  }
  scale_RESTR <-
    sapply(1:length(extra_groups), function(g)
      sum(exp(log(eigenvalues[, g]) - log(shape_RESTR))) / fitm$d)
  eigenvalues <-  shape_RESTR %o% scale_RESTR
}
#I update the variance components in the output of the M-step
fitm$parameters$variance$shape <- shape_RESTR
fitm$parameters$variance$scale[extra_groups] <- scale_RESTR
if (fitm$modelName == "VII") {
  #I also update sigmasq for VII model
  fitm$parameters$variance$sigmasq[extra_groups] <-
    fitm$parameters$variance$scale[extra_groups]
}
fitm$parameters$variance$sigma <-
  mclust::decomp2sigma(
    d = fitm$d,
    G = fitm$G,
    scale = fitm$parameters$variance$scale,
    shape = fitm$parameters$variance$shape,
    orientation = fitm$parameters$variance$orientation
  )
return(fitm)
```

```
  }

restr_2_VV_ <-
  function(fitm,
           restr_factor,
           extra_groups,
           ctrl_restr) {
    if (fitm$modelName == "V") {
      max_sigmasq <- max(fitm$parameters$variance$sigmasq)
      min_sigmasq <- min(fitm$parameters$variance$sigmasq)
      #univariate (d=1) case
      if (max_sigmasq / min_sigmasq <= restr_factor) {
        return(fitm)
      } else if (length(extra_groups) == 1) {
        if (all(fitm$parameters$variance$sigmasq[extra_groups] <
                fitm$parameters$variance$sigmasq[-extra_groups])) {
          sigmasq_RESTR <-
            min(fitm$parameters$variance$sigmasq[-extra_groups])
        } else {
          sigmasq_RESTR <-
            fitm$parameters$variance$sigmasq[extra_groups]
        }
      } else {
        sigmasq_RESTR <- as.vector(
          restr2_eigenv(
            fitm$parameters$variance$sigmasq[extra_groups],
            ni.ini = colSums(fitm$z[, extra_groups, drop = FALSE]),
            restr.fact = restr_factor,
            zero.tol = .Machine$double.eps
          )
        )
      }
      fitm$parameters$variance$sigmasq[extra_groups] <-
        sigmasq_RESTR
      fitm$parameters$variance$scale[extra_groups] <- sigmasq_RESTR
      return(fitm)
    }

    if (fitm$modelName == "VVV") {
      fitm$parameters$variance <-
        SIGMA_COMP(fitm$parameters$variance)
    }

    eigenvalues <-
      exp(sweep(
        x = log(fitm$parameters$variance$shape[, extra_groups, drop = FALSE]),
        2,
        log(fitm$parameters$variance$scale[extra_groups]),
        FUN = "+"
      ))
```

```
if (ifelse(
  is.na(max(eigenvalues) / min(eigenvalues) <= restr_factor) |
  is.infinite(max(eigenvalues) / min(eigenvalues)),
  TRUE,
  max(eigenvalues) / min(eigenvalues) <= restr_factor
)) {
  return(fitm)
}

eigenvalues_RESTR <-
  restr2_eigenv(
    eigenvalues,
    ni.ini = colSums(fitm$z[, extra_groups, drop = FALSE]),
    restr.fact = restr_factor,
    zero.tol = .Machine$double.eps
  )

scale_RESTR <- apply(eigenvalues_RESTR, 2, function(x)
  exp(1 / fitm$d * sum(log(x))))
shape_RESTR <- exp(sweep(
  x = log(eigenvalues_RESTR),
  2,
  log(scale_RESTR),
  FUN = "-"
))

if (fitm$modelName == "VVE" &
    length(extra_groups) == fitm$G) {
  # Transductive approach
  W <-
    mclust::covw(fitm$X, Z = fitm$z, normalize = F)$W
  E <-
    apply(W, 3, function(x)
      eigen(x, only.values = T)$val) #eigenvalues of W
  w <- apply(E, 2, max) #biggest eigenv of W_g, g=1, ..., G
  orientation_RESTR <-
    MM1(
      A = shape_RESTR,
      W = W,
      w = w,
      ctrl_restr = ctrl_restr
    )
  fitm$parameters$variance$orientation <- orientation_RESTR
}
#I update the variance components in the output of the M-step
fitm$parameters$variance$shape[, extra_groups] <- shape_RESTR
fitm$parameters$variance$scale[extra_groups] <- scale_RESTR
fitm$parameters$variance$sigma <-
  mclust::decomp2sigma(
```

```
        d = fitm$d,
        G = fitm$G,
        scale = fitm$parameters$variance$scale,
        shape = fitm$parameters$variance$shape,
        orientation = fitm$parameters$variance$orientation
      )
    #cholsigma for VVV
    if (fitm$modelName == "VVV") {
      fitm$parameters$variance$cholsigma <-
        tryCatch(
          array(as.vector(
            apply(fitm$parameters$variance$sigma,
                  3, chol)
          ),
          dim = c(fitm$d, fitm$d, fitm$G)),
          error = function(e)
            NA
        )
    }
    fitm
  }
```

B.2 EM algorithm for RAEDDA transductive (Section 3.2.2)

```
  while (criterion) {
     iter <- iter + 1
     fite <-
       tryCatch(
         do.call(mclust::estep, c(list(data = X_test), fitm)),
         error = function(e) {
           list(z = NA)
         }
       ) # E-step

     emptyz <- TRUE
     if ((all(!is.na(fite$z))) &
      (all(colSums(fite$z)>.Machine$double.eps))) { # error checking
       emptyz <- FALSE
       z <- fite$z
       z_fit <- fite$z

       # Concentration Step Y test
       if (alpha_test != 0) {
         D <-
           do.call(mclust::dens, c(list(
             data = X_test,
             # compute the Density for patterned
             # MVN Mixtures for each obs
             logarithm = T
           ), fitm))
```

```
    # temporarily discards those alpha_test%
    # of obs whose density is lowest
  pos_trimmed_test <-
    which(D <= (sort(D, decreasing = F)
    [[ceiling(N_test * alpha_test)]]))
  z_fit <- z[-pos_trimmed_test, , drop = F]
  Xtest_fit <- X_test[-pos_trimmed_test, , drop = F]
}

# Concentration Step Xtrain
if (alpha_train != 0) {
  D_Xtrain_cond <-
    do.call(mclust::cdens, c(list(
      data = X_train, # computing the component density
      logarithm = T
    ), fitm))
  ind_D_Xtrain_cdens <-
    cbind(1:N_train, mclust::map(ltrain))
  D_Xtrain <-
    D_Xtrain_cond[ind_D_Xtrain_cdens]
  pos_trimmed_train <-
    which(D_Xtrain <= (sort(D_Xtrain, decreasing = F)
                       [[ceiling(N_train * alpha_train)]]))
  ltrain_fit <- ltrain[-pos_trimmed_train, , drop = F]
  Xtrain_fit <- X_train[-pos_trimmed_train, , drop = F]
}

Xall <- rbind(Xtrain_fit, Xtest_fit)
zall <- rbind(ltrain_fit, z_fit)
fitm <-
    mclust::mstep(
      modelName = model_name,
      data = Xall,
      z = zall
    )
if (!any(is.na(fitm$parameters$variance$sigma))) {
  if (fitm$modelName == "VVE" | fitm$modelName == "EVE") {
    fitm$X <-
      Xall
      # I add the data on which the M-step is computed
      # since I need them for the MM
  }
  suppressWarnings(
    fitm <-
      constr_Sigma(
        fitm = fitm,
        restr_factor = restr_factor,
        extra_groups = 1:fitm$G,
        # it identifies the transductive approach
        ctrl_restr = ctrl_restr
```

```
            )
          )# this performs constrained estimation of Sigma
           # according to the selected model
        }

        ll <-
          logLik_raedda(
            X_train = Xtrain_fit,
            ltrain = ltrain_fit,
            X_test = Xtest_fit,
            fitm = fitm
          )

        llstore <- c(llstore[-1], ll)
        if (aitken) {
          criterion <- (Aitken(llstore)$linf - ll) > EM_tol
        } else {
          criterion <- (ll - llold) > EM_tol
        }
        criterion <- (criterion) & (iter < EM_max_iter)
        llold <- ll
      } else {
        criterion <- FALSE
      }
    }
```

B.3 EM algorithm for RAEDDA inductive (Section 3.2.3)

```
while (criterion) {
      iter <- iter + 1
      fite <-
        tryCatch(
          do.call(mclust::estep, c(list(data = X_test), fitm)),
          error = function(e)
            list(z = NA)
        ) # E-step
      emptyz <- TRUE
      if (all(!is.na(fite$z)))
      {
        emptyz <- FALSE
        z <- fite$z
        z_fit <- fite$z
        # Concentration Step
        # for the augmented test set
        if (alpha_discovery != 0) {
          D <-
            do.call(mclust::dens, c(list(
              data = X_test,
              logarithm = F
            ), fitm))
```

```
  pos_trimmed_test <-
    which(D <= (sort(D, decreasing = F)[[
    ceiling(N_test * alpha_discovery)]]))
  z_fit <- z[-pos_trimmed_test, , drop = F]
  X_test_fit <- X_test[-pos_trimmed_test, , drop = F]
}

#For extra_groups I manually update pro, mean and variance
pro_extra <- colMeans(z_fit[, extra_groups, drop = F])
fitm$parameters$pro <-
  c(fit_learning$Best$parameters$pro *
   (1 - sum(pro_extra)), pro_extra)
  # proportions are re-estimated for each class
if (fitm$d == 1) {
  fitm$parameters$mean[extra_groups] <-
    mclust::covw(X = X_test_fit,
                 Z = z_fit,
                 normalize = F)$mean[, extra_groups]

} else {
  fitm$parameters$mean[, extra_groups] <-
    mclust::covw(X = X_test_fit,
                 Z = z_fit,
                 normalize = F)$mean[, extra_groups]
# I update the mu vector just for the extra groups
}

fitm$parameters$variance <-
  UPDATE_SIGMA( # function that implements formulae in Section 3.6
    fitm = fitm,
    extra_groups = extra_groups,
    z = z_fit,
    X = X_test_fit
  )

if (!(any(is.na(fitm$parameters$variance$sigma))|
any(is.na(fitm$parameters$variance$cholsigma)))) {
  if(fitm$modelName=="VVE"|fitm$modelName=="EVE"){
    fitm$X <- X_test_fit
  }

  suppressWarnings(fitm <-
                     constr_Sigma(
                       fitm = fitm,
                       restr_factor = restr_factor_d,
                       extra_groups = extra_groups,
                       # constraints enforced only on
                       # the extra groups (Inductive approach)
                       ctrl_restr = ctrl_restr
                     ) )
```

```
                        # this performs constrained estimation of Sigma
                        # according to the selected model
                        # for the extra groups
      }

      ll <-
        suppressWarnings(tryCatch(
          sum(do.call(mclust::dens, c(
            list(data = X_test_fit, logarithm = TRUE),
            fitm
          ))),
          error = function(e)
            - Inf
        ))
        # value of the trimmed log-likelihood
      ll <-
        ifelse(is.nan(ll) |
                 is.na(ll), -Inf, ll)
                 # If ll is NA or Nan it proceeds
                 # till the next estep and then the loop breaks
      llstore <- c(llstore[-1], ll)
      if (aitken) {
        criterion <- (Aitken(llstore)$linf - ll) > EM_tol
      } else {
        criterion <- (ll - llold) > EM_tol
      }
      criterion <- (criterion) & (iter < EM_max_iter)
      llold <- ll
    }
    else {
      criterion <- FALSE
    }
  }
```

Appendix C

Details on computing time

This appendix provides some details related to the computing time required by the novel routines developed in the present manuscript. All the simulated experiments and real data analyses in the thesis were run on a computer cluster with 12 processors Intel MKL Intel(R) Xeon(R) E5-2697 @2.60GHz. Tables C.1 and C.2 report the average computing time in seconds and relative time with respect to the EDDA model for the two simulation studies discussed in Chapter 2: Section 2.3.1 and Section 2.3.2, respectively. Tables C.3 and C.4 report the average computing time in seconds for the simulation study in Chapter 3 (see Section 3.3), for the two different group proportions respectively. Table C.5 provides the average computing time in seconds for the variable selection methods considered in the simulation study of Chapter 4 (see Section 4.3). Lastly, the computing time for robustly selecting the relevant variables in the starches discrimination problem of Section 4.4 was equal to 21460.44 and 3556.75 seconds for the robust stepwise greedy-forward approach via TBIC (Section 4.2.1) and the maximum likelihood subset selector (Section 4.2.2), respectively. As expected, the price to pay, in terms of additional time, for achieving robust estimates is of several orders of magnitude larger than that required for non-robust methodologies. Although much care and effort have been put on code profiling and efficient implementation, the developed procedures intrinsically necessitate extra computing power in order to be robust against contamination. Some specific comments follow, with reference to the main differences between the classical and the robust approach.

1. to obtain a robust estimation on the learning set, multiple initializations are unavoidable. They are needed to prevent noisy units to spoil the starting values, and henceforth driving the entire algorithm to reach uninteresting solutions.

2. to enforce a restriction on the eigenvalue-ratio (when it is active, i.e., $c = 3$), as it can be seen in Table C.2, requires extra computing time than fitting an unconstrained model, even though all routines in Section 2.8 rely on the optimal truncation operator efficiently implemented in the `tclust` package (Fritz et al., 2012).

3. the inductive estimation of the RAEDDA model is substantially faster than its transductive version (see Tables C.3 and C.4), since parameters for the known groups need not be updated when fitting the model to the test set. This behavior is even more apparent when flexible covariance models are already selected in the learning phase, thanks to the partial-order structure of Figure 3.4.

4. wrapper methods are by their very nature computationally intensive, as the selected model needs to be fitted multiple times while looking for relevant features. Unsurprisingly therefore, the robust procedures of Chapter 4 require a non-negligible amount of time for performing variable selection resistant to outliers and label noise.

As a last worthy note, we mention the fact that all the novel routines developed for this thesis (some of which are reported in Appendix B) have been written in pure `R`, which is known to

TABLE C.1: Average computing time (in seconds) on $B = 1000$ runs for the simulation study I (Section 2.3.1) of Chapter 2, varying method and contamination rate η. Average relative time with respect to the EDDA model is reported in parenthesis.

η	0	0.05	0.10	0.15	0.20	0.25
EDDA	0.014	0.015	0.015	0.015	0.015	0.015
	(1)	(1)	(1)	(1)	(1)	(1)
UPCLASS	0.485	0.765	1.004	1.161	1.214	1.238
	(34.512)	(52.103)	(68.119)	(76.573)	(81.053)	(83.68)
RMDA	0.25	0.275	0.3	0.309	0.327	0.33
	(17.777)	(18.734)	(20.35)	(20.37)	(21.859)	(22.294)
RLDA	0.628	0.563	0.478	0.387	0.471	0.54
	(44.73)	(38.351)	(32.413)	(25.54)	(31.426)	(36.494)
REDDA	4.402	4.101	3.88	3.943	4.212	4.204
	(313.291)	(279.327)	(263.247)	(260.079)	(281.184)	(284.249)
RUPCLASS	5.409	5.084	4.821	4.884	5.486	6.034
	(384.969)	(346.339)	(327.08)	(322.151)	(366.194)	(408.011)

be slower than compiled languages. Fairly evaluating the effective runtime and speed of an algorithm is not an easy task, as anytime a comparison is performed software implementation, programming language and computational resources are (often unknowingly) not taken into account: a provocative discussion about this topic can be found in Kriegel et al., 2017.
All in all, despite the undeniable additional computational burden, we argue here that a robust procedure should always be preferred whenever contaminated units are likely to be present in the data, as a reliable and noise resistant solution tends to be well worth the required extra computing time.

TABLE C.2: Average computing time (in seconds) and relative time with respect to the EDDA model on $B = 1000$ runs for the simulation study II (Section 2.3.2) of Chapter 2.

	Computing time	Relative time w.r.t. EDDA
EDDA	0.27	1.00
UPCLASS	1.51	5.68
RLDA	6.00	22.52
SVM	1.14	4.28
AdaBoost	9.33	35.06
RSIMCA	0.21	0.80
REDDA $(\alpha_l = 0.05)$	13.40	50.35
REDDA$(\alpha_l = 0.1)$	15.86	59.60
RUPCLASS $(c = 30, \alpha_l = 0.05, \alpha_u = 0)$	117.29	440.61
RUPCLASS $(c = 30, \alpha_l = 0.1, \alpha_u = 0)$	67.54	253.72
RUPCLASS $(c = 10, \alpha_l = 0.1, \alpha_u = 0.05)$	49.32	185.28
RUPCLASS $(c = 3, \alpha_l = 0.1, \alpha_u = 0.05)$	73.80	277.24
RUPCLASS $(c = 30, \alpha_l = 0.15, \alpha_u = 0.1)$	40.10	150.66
RUPCLASS $(c = 10, \alpha_l = 0.15, \alpha_u = 0.1)$	34.48	129.54
RUPCLASS $(c = 3, \alpha_l = 0.15, \alpha_u = 0.1)$	72.42	272.03

TABLE C.3: Average computing time (in seconds) on $B = 1000$ runs for the simulation study in Chapter 3 (see Section 3.3), under different covariance structure and contamination rate. `Equal` groups proportion.

	Method	$Q_l = Q_u = 0$	$Q_l = 10, Q_u = 40$	$Q_l = 20, Q_u = 80$	$Q_l = 30, Q_u = 120$
EII	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	57.83	52.43	52.33	52.90
	AMDAt	2.97	3.97	7.86	6.69
	AMDAi	0.75	1.03	1.78	1.54
	RAEDDAt	23.20	31.01	25.04	25.02
	RAEDDAi	17.20	25.68	22.81	22.79
EEI	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	51.94	49.70	50.71	52.18
	AMDAt	2.15	3.21	10.42	6.70
	AMDAi	0.68	0.96	1.46	1.55
	RAEDDAt	60.66	63.45	68.98	66.98
	RAEDDAi	28.70	22.87	23.35	28.83
EVV	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	40.20	39.73	40.70	41.66
	AMDAt	1.69	2.75	8.60	4.70
	AMDAi	0.50	0.92	1.16	1.16
	RAEDDAt	36.19	44.57	38.99	37.08
	RAEDDAi	18.71	17.35	15.49	15.94
VVV	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	39.82	39.50	40.76	41.76
	AMDAt	1.93	3.71	12.44	6.67
	AMDAi	0.59	0.83	1.05	1.19
	RAEDDAt	60.18	62.37	64.34	63.85
	RAEDDAi	2.32	3.98	4.20	4.37
VVV	QDA-ND	0.01	0.01	0.01	0.01
Overlap	SVM-ND	40.11	39.93	41.00	42.02
	AMDAt	2.26	4.12	13.65	6.83
	AMDAi	0.77	0.98	1.28	1.50
	RAEDDAt	66.90	75.46	75.11	69.13
	RAEDDAi	2.32	4.27	4.47	4.65

TABLE C.4: Average computing time (in seconds) on $B = 1000$ runs for the simulation study in Chapter 3 (see Section 3.3), under different covariance structure and contamination rate. `Unequal` groups proportion.

	Method	$Q_l = Q_u = 0$	$Q_l = 10, Q_u = 40$	$Q_l = 20, Q_u = 80$	$Q_l = 30, Q_u = 120$
EII	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	53.17	49.06	51.33	53.85
	AMDAt	2.88	3.52	8.28	5.15
	AMDAi	0.60	1.00	1.50	1.54
	RAEDDAt	27.53	31.26	25.76	26.38
	RAEDDAi	15.61	25.08	24.45	23.84
EEI	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	52.84	53.04	52.19	53.57
	AMDAt	2.06	2.85	10.02	6.58
	AMDAi	0.76	1.12	1.22	1.35
	RAEDDAt	62.78	64.46	68.96	66.47
	RAEDDAi	20.47	20.33	20.18	20.54
EVV	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	40.32	39.80	40.97	41.81
	AMDAt	1.50	2.72	8.73	5.65
	AMDAi	0.42	0.75	0.97	0.92
	RAEDDAt	38.45	44.36	40.80	38.22
	RAEDDAi	14.67	18.19	18.96	18.91
VVV	QDA-ND	0.01	0.01	0.01	0.01
	SVM-ND	40.12	40.00	41.11	41.95
	AMDAt	1.83	3.65	14.29	7.74
	AMDAi	0.59	0.70	0.99	1.09
	RAEDDAt	67.71	70.45	71.76	70.62
	RAEDDAi	2.27	4.11	4.21	4.41
VVV	QDA-ND	0.01	0.01	0.01	0.01
Overlap	SVM-ND	40.60	40.16	41.48	42.36
	AMDAt	2.20	3.93	16.68	8.41
	AMDAi	0.73	0.89	1.26	1.27
	RAEDDAt	69.36	80.10	79.84	77.90
	RAEDDAi	2.32	4.25	4.46	4.68

TABLE C.5: Average computing time (in seconds) on $B = 100$ runs for the simulation study in Chapter 4 (see Section 4.3), for different variable selection methods.

SelvarMix	SRUW	TBIC	ML subset (p=3)	ML subset (p=6)	ML subset (p=9)
5.80	62.22	577.41	243.40	314.88	339.97

Appendix D

A note on trimmed information criteria for robust model selection

This appendix provides a discussion and some initial attempts to justify the usage of trimmed information criteria in the context of robust model selection. Model selection criteria are penalized MLE's designed to decide between one of several proposed statistical models. The penalty term is subtracted from the log-likelihood. It depends on the model dimension, penalizing methods with higher number of free parameters thus balancing a parsimonious use of parameters with goodness of fit. Quite naturally, the rationale behind trimmed information criteria like the ones in Sections 2.2.4 and 3.2.5 stems from the need of defining a model selection procedure whose output should result close to that obtained by standard methods on the genuine part of the data only. Indeed, robustly estimated parameters are not sufficient to provide reliable model selection if the maximized likelihood is subsequently evaluated on the entire dataset: noisy units still contribute to the value of standard criteria and their effect, albeit small, could still badly affect the overall behavior. This had been directly experienced by the author within a germinal (and naive) structure of the robust variable selection procedure in Section 4.2.1, in which a non-robust model selection criterion had been initially employed for model comparison. Another example of such undesirable outcome, in a weighted likelihood framework, is reported in Section 5 of Greco and Agostinelli, 2019.

Firstly proposed by Neykov et al., 2007, the authors asserted that the trimmed BIC (TBIC) could be employed for robustly assessing the number of mixture components and the percentage of contamination in the data. Since its first introduction, trimming criteria have been extensively employed in the literature for providing a general way to perform robust model selection. García-Escudero et al., 2010a suggested its usage for selecting the trimming levels in robust clusterwise linear regression, Gallegos and Ritter, 2010 devised a "corrected BIC" for model selection with trimming in cardinality-constrained clustering, García-Escudero et al., 2016 adopted the TBIC for choosing both the number of components and latent factors in robust mixtures of Gaussian factor analyzers while García-Escudero et al., 2017 proposed to employ it as a data-dependent diagnostic for tuning number of groups, trimming level and constraints value in a robust cluster weighted model. A whole new set of trimmed information criteria based on the original idea of Neykov et al., 2007 were developed by Li et al., 2016: they all were applied in simulation results with the purpose of robustly estimating the number of components in a mixture of regression model. More recently, the novel penalized likelihood criterion for constrained mixtures, introduced in Cerioli et al., 2018a, is expected to be extended to account for the trimming level α in the TCLUST framework García-Escudero et al., 2008. Such proposal is favorably endorsed by García-Escudero et al., 2018a within a contributed comment on "The power of monitoring: how to make the most of a contaminated multivariate sample" (Cerioli et al., 2018b).

Even though the TBIC is shown to work well in all the methodologies described above and in

the applications treated in this manuscript, a more general consideration on the usage of trimming criteria to perform model selection in robust mixture learning is in order. Despite their effective adoption in performing robust model selection, a proper theory corroborating such practice is still missing in the literature: Ritter, 2014 deems the trimmed BIC and the corrected BIC as to be "heuristics that work well when the true groups are well separated". Providing the formal theory underlying such methods is not an easy task and goes beyond the scope of the present manuscript, nevertheless an attempt of delineating a possible research direction is outlined at the end of Section 4.2.1, where the spurious outlier model (see Section 1.3.1) is proposed as the underlying probability density for the contaminated dataset. A more general argument in favor of this framework is provided in the upcoming Section: notice however that it is only meant to scratch the surface of the (still under development) theory of robust information criteria, and therefore shall be regarded as a possible starting point for more thorough research, rather than be treated as a formal result.

D.1 Formalizing the Trimmed BIC derivation: an initial attempt

Consider a sample of N observations to be partitioned into N_1 "regular" and N_0"non-regular" units, such that $N = N_1 + N_0$. Generalizing the spurious-outlier model (Gallegos and Ritter, 2005), firstly introduced in the context of robust clustering, let the regular sample $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \dots, N_1$ to be an i.i.d. realization of a probability density function $f(\cdot|\boldsymbol{\theta}_1)$ parametrized by $\boldsymbol{\theta}_1 \in \boldsymbol{\Theta}_1$ with prior probability $\pi(\boldsymbol{\theta}_1)$. The non-regular units $\mathbf{y}_j \in \mathbb{R}^p$, $j = 1, \dots, N_0$ are assumed to be a realization of independent subject-specific probability density functions $w_j(\cdot|\boldsymbol{\psi}_j)$, each one parametrized by $\boldsymbol{\psi}_j \in \boldsymbol{\Psi}_j$ with prior probabilities $\pi_j(\boldsymbol{\psi}_j)$, $j = 1, \dots, N_0$. The likelihood for the given sample therefore reads:

$$L(\boldsymbol{\theta}_1, \boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_{N_0}|\mathbf{X}, \mathbf{Y}) = \prod_{i=1}^{N_1} f(\mathbf{x}_i|\boldsymbol{\theta}_1) \prod_{j=1}^{N_0} w_j(\mathbf{y}_j|\boldsymbol{\psi}_j) \tag{D.1}$$

Assuming independence for the prior distributions $\pi(\boldsymbol{\theta}_1), \pi_1(\boldsymbol{\psi}_1), \dots, \pi_{N_0}(\boldsymbol{\psi}_{N_0})$, the integrated likelihood of (D.1) factors as follows:

$$f(\mathbf{X}) \prod_{j=1}^{N_0} w_j(\mathbf{Y}_j) = \int_{\boldsymbol{\Theta}_1} \prod_{i=1}^{N_1} f(\mathbf{x}_i|\boldsymbol{\theta}_1)\pi(\boldsymbol{\theta}_1)d\boldsymbol{\theta}_1 \prod_{j=1}^{N_0} \int_{\boldsymbol{\Psi}_j} w_j(\mathbf{y}_j|\boldsymbol{\psi}_j)\pi_j(\boldsymbol{\psi}_j)d\boldsymbol{\psi}_j. \tag{D.2}$$

The following proposition justifies the employment of the Trimmed BIC (Neykov et al., 2007) for approximating the integrated likelihood in (D.2), henceforth favoring its usage in the context of robust model selection.

Proposition D.1: Assume that the Trimmed Likelihood Estimator (TLE) correctly partitions the observed dataset into N_1 regular and N_0 outlying units, and that the model assumed for the regular part is well-specified. Then, the Schwarz approximation (Schwarz, 1978) for the integrated likelihood in (D.2) agrees with the TBIC of Neykov et al., 2007.

Proof: The log model evidence (i.e., the logarithm of the integrated likelihood) is approximated by the Bayesian information criterion (BIC):

$$\log\left(f(\mathbf{X}) \prod_{j=1}^{N_0} w_j(\mathbf{Y}_j)\right) = \log f(\mathbf{X}) + \sum_{j=1}^{N_0} \log w_j(\mathbf{Y}_j) \approx BIC(X) + \sum_{j=1}^{N_0} BIC(Y_j) \tag{D.3}$$

where

$$BIC(X) = \sum_{i=1}^{N_1} \log f(\mathbf{x}_i|\hat{\boldsymbol{\theta}}_1) - \frac{1}{2} v_x \log N_1 \tag{D.4}$$

is the standard BIC computed on the regular part of the data only, with $\hat{\boldsymbol{\theta}}_1$ maximum likelihood estimator (MLE) of $\boldsymbol{\theta}_1$ and v_x the dimension (i.e., number of parameters) for $\boldsymbol{\theta}_1$. This is due to the fact that, under the considered assumptions, TLE and MLE computed on the subset of N_1 regular units coincide. The quantity in (D.4) is equivalent to the TBIC defined in Neykov et al., 2007 (up to a multiplicative constant), thus, for concluding the proof, it suffices to show that the approximation of the log model evidence for the non-regular units is equal to 0. Start by noticing that:

$$BIC(Y_j) = \log w_j(\mathbf{y}_j|\hat{\boldsymbol{\psi}}_j) \quad j = 1, \ldots, N_0 \tag{D.5}$$

with $\hat{\boldsymbol{\psi}}_j$ MLE for $\boldsymbol{\psi}_j$, since just a single observation contributes to the log maximized likelihood $\forall j, j = 1, \ldots, N_0$, so that the penalization term disappears. Further, since the probability density functions $w_j(\cdot|\boldsymbol{\psi}_j)$ are specific for each contaminated unit $\mathbf{y}_j$, the estimator of $w_j(\cdot|\boldsymbol{\psi}_j)$ is given by the Dirac delta in $\mathbf{y}_j$. Therefore, at the MLE, the likelihood contribution for each contaminated unit $\mathbf{y}_j$ is equal to 1: taking the logarithm nullifies the value of (D.5), proving the initial statement.

□

The reasoning employed in proving Proposition D.1 justifies, in principle, the usage of the Trimmed BIC in selecting model parameters as well as the percentage of contamination in the data. One of the major criticism moved against TBIC is that different subsets of units may be considered when computing it for different models, even in the case of a fixed trimming level: this could be improperly seen as comparing models with BIC calculated on different data points. However, under the specification in (D.1), models comparison is actually always conducted via BIC on the whole dataset, but, as stated in the proof above, the likelihood for each contaminated unit is equal to 1 at the MLE and its contribution in the BIC calculation equals 0, thus retrieving the TBIC definition. Notice that the same reasoning holds even when models with different trimming levels are compared. Notwithstanding, as already mentioned at the end of the previous Section, this is just an initial result developed by the author at the very end of his PhD program, and further studies are hence needed for corroborating the validity of the presented argument. A limit of the result derived in Proposition D.1 lies on the assumption that the trimmed units are precisely the ones not belonging to the regular model. Clearly, the identification of the non-regular units is itself an output of the inferential procedure: considering it deterministic is a strong assumption that needs to be relaxed. Active research on this topic is currently ongoing.

Acknowledgements

At the end of a journey, it often occurs to look back and realize how far you have gone, and how much you have enjoyed the course. Mine has not been a solo trip; several fellow travelers have accompanied me and all of you shall be thanked: I would simply not have concluded this adventure without you.
Francesca, you are the first person I am grateful to. You are a great supervisor, your endless passion and research enthusiasm are contagious. You have guided me since the very beginning of my PhD expedition, never letting me lose the track, while always being proactive in having me explore new routes. You have been a mentor, teaching me knowledge and values that go far beyond statistics, I will never be able to thank you enough.
Brendan, deep thanks go to you too. For having agreed in accepting me at UCD in the first place, and for your constant dedication in co-supervising me while I have been in Dublin. You are a great professor, I have learnt so much from you.
I thank Agustin Mayo-Iscar and Luis Angel García Escudero for both stimulating discussion and advises on how to enforce the eigenvalue-ratio constraints under the different patterned models of Section 2.8.
I am very grateful to Anna Sandionigi, Lorenzo Guzzetti, Maurizio Casiraghi and Massimo Labra for fruitful discussions and domain-knowledge sharing: their valuable feedback deeply enhanced the quality of Section 3.4.
I would like to acknowledge Gunter Ritter for the stimulating discussions and suggestions on how to transpose the ML subset selector approach, originally developed for clustering, to the classification framework in which also the entire family of patterned models is considered.
I thank Ludovic Duponchel for providing relevant context on how the novel methodologies developed in this manuscript may be favorably employed, as well as for supplying the dataset on which the application in Section 4.4 was based on.
I thank Fra, Mattia, Nik, Luca, Luigi and Riccardo for having made the first year of the PhD program unforgettable, it has been a "measurable" (bearable) experience, and not a Vitali one, only due to your almost surely contribution. A special mention goes to Francesco and his Bayesian fundamentalism: you are a great friend and a great researcher, just do not forget to remind this to yourself (i.e., to update your prior belief) every now and then. The morning vocal messages about our research news on my way to work have been an essential part of my PhD routine.
I thank Riccardo (x2), Michael, Silvia, Lida and all the Insight and Killarney folk: I am honored to have shared my Irish excursion with you. Particularly, I am grateful to Riccardo for the pints and for having had the (questionable) pleasure of living together in Dublin 1. Thank you also for having fed me when I have been too lazy to cook: Supervalu soup I owe you too. Michael, you have been such a landmark for me I could easily account you to be a supervisor of mine. You are so good and so passionate about your work it is almost magical talking stats with you. You have spent countless hours correcting my mistakes, listening to my delusions and improving my coding skills, too bad you are still not a tidyverse supporter. You and Silvia are fantastic people and make such a great team, I wish you all the very best for the future.
Anna and Andrea, it was a pleasure to share the office with you during the last hike before the thesis submission, you have made it a truly enjoyable time, despite the approaching deadline.

I thank my parents Ruggero and Gabriella, for having believed in me and having let me find my own route, nevertheless always being my compass whenever I felt too lost to proceed by myself. I owe you everything.
I deeply thank the entire Sanse crew, the thrilling of sailing uncharted waters would be much more terrifying without the certainty of having a safe haven to dock to. You are the secure harbor I can always count on. Despite the difficulty of being friend with a person you seldom see, every time I come home you make me feel like I had never left: I am truly blessed for having such an awesome group of friends, thank you.
Last but not least, to the person who has willingly chosen to love me regardless of all my flaws and insecurities, with my wanderlust syndrome that so many times has separated us. Alice, I had you go through so many "goodbye" and "see you soon" I will never make up for. And yet here you are, first Dublin and now Milan: you are the greatest gift God could have given me and, with His blessing, it will soon be forever. Because the best is yet to come and, with you at my side, the journey has just started.

Bibliography

[1] A. C. Aitken. "A series formula for the roots of algebraic and transcendental equations". In: *Proceedings of the Royal Society of Edinburgh* 45.01 (1926), pp. 14–22.

[2] H. Akaike. "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723.

[3] A. Alfons, C. Croux, and S. Gelper. "Sparse least trimmed squares regression for analyzing high-dimensional large data sets". In: *The Annals of Applied Statistics* 7.1 (2013), pp. 226–248.

[4] C. Alimentarius. "Revised codex standard for honey". In: *Codex stan* 12 (2001), p. 1982.

[5] D. F. Andrews, P. J. Bickel, F. R. Hample, P. J. Huber, W. H. Rogers, and J. W. Tukey. *Robust Estimates of Location: Survey and Advances*. Princeton University Press, 1972.

[6] J. L. Andrews and P. D. McNicholas. "Variable Selection for Clustering and Classification". In: *Journal of Classification* 31.2 (2014), pp. 136–153.

[7] D. Angluin and P. Laird. "Learning From Noisy Examples". In: *Machine Learning* 2.4 (1988), pp. 343–370.

[8] J Baek, G. J. McLachlan, and L. K. Flack. "Mixtures of Factor Analyzers with Common Factor Loadings: Applications to the Clustering and Visualization of High-Dimensional Data". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.7 (2010), pp. 1298–1309.

[9] J. D. Banfield and A. E. Raftery. "Model-based Gaussian and non-Gaussian clustering". In: *Biometrics* 49.3 (1993), p. 803.

[10] R. Barandela and E. Gasca. "Decontamination of training samples for supervised pattern recognition methods". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 1876 LNCS. 2000, pp. 621–630.

[11] V. Barnett and T. Lewis. *Outliers in statistical data*. Wiley, 1974.

[12] R. Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957.

[13] H. Bensmail and G. Celeux. "Regularized Gaussian discriminant analysis through eigenvalue decomposition". In: *Journal of the American Statistical Association* 91.436 (1996), pp. 1743–1748.

[14] C. Biernacki. "Degeneracy in the Maximum Likelihood Estimation of Univariate Gaussian Mixtures for Grouped Data and Behaviour of the EM Algorithm". In: *Scandinavian Journal of Statistics* 34.3 (2007), pp. 569–586.

[15] C. Biernacki, G. Celeux, and G. Govaert. "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models". In: *Computational Statistics & Data Analysis* 41.3-4 (2003), pp. 561–575.

[16] A. L. Blum and P. Langley. "Selection of relevant features and examples in machine learning". In: *Artificial Intelligence* 97.1-2 (1997), pp. 245–271.

[17] D. Böhning, E. Dietz, R. Schaub, P. Schlattmann, and B. G. Lindsay. "The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family". In: *Annals of the Institute of Statistical Mathematics* 46.2 (1994), pp. 373–388.

[18] N. A. Bokulich, J. H. Thorngate, P. M. Richardson, and D. A. Mills. "Microbial biogeography of wine grapes is conditioned by cultivar, vintage, and climate". In: *Proceedings of the National Academy of Sciences* 111.1 (2014), E139–E148.

[19] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang. "Benchmark for filter methods for feature selection in high-dimensional classification data". In: *Computational Statistics & Data Analysis* (2019), p. 106839.

[20] N. Bouguila and D. Ziou. "A powerful finite mixture model based on the generalized Dirichlet distribution: unsupervised learning and applications". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 1. IEEE. 2004, pp. 280–283.

[21] C. Bouveyron. "Adaptive mixture discriminant analysis for supervised learning with unobserved classes". In: *Journal of Classification* 31.1 (2014), pp. 49–84.

[22] C. Bouveyron and C. Brunet-Saumard. "Model-based clustering of high-dimensional data: A review". In: *Computational Statistics and Data Analysis* 71 (2014), pp. 52–78.

[23] C. Bouveyron and S. Girard. "Robust supervised classification with mixture models: Learning from data with uncertain labels". In: *Pattern Recognition* 42.11 (2009), pp. 2649–2658.

[24] C. Bouveyron, G. Celeux, T. B. Murphy, and A. E. Raftery. *Model-Based Clustering and Classification for Data Science: With Applications in R*. Vol. 50. Cambridge University Press, 2019.

[25] J. M. Brenchley, U. Hörchner, and J. H. Kalivas. "Wavelength selection characterization for NIR spectra". In: *Applied Spectroscopy* (1997).

[26] C. E. Brodley and M. A. Friedl. "Identifying Mislabeled Training Data". In: *Journal of Artificial Intelligence Research* 11 (1999), pp. 131–167.

[27] P. J. Brown. "Wavelength selection in multicomponent near-infrared calibration". In: *Journal of Chemometrics* (1992).

[28] R. P. Browne and P. D. McNicholas. "Estimating common principal components in high dimensions". In: *Adv Data Anal Classif* 8 (2014), pp. 217–226.

[29] M. L. Calle. "Statistical Analysis of Metagenomics Data". In: *Genomics & Informatics* 17.1 (2019), e6.

[30] R. B. Cattell. "The scree test for the number of factors". In: *Multivariate Behavioral Research* 1.2 (1966), pp. 245–276.

[31] G. Celeux and G. Govaert. "A classification EM algorithm for clustering and two stochastic versions". In: *Computational Statistics and Data Analysis* 14.3 (1992), pp. 315–332.

[32] G. Celeux and G. Govaert. "Gaussian parsimonious clustering models". In: *Pattern Recognition* 28.5 (1995), pp. 781–793.

[33] G. Celeux, C. Maugis-Rabusseau, and M. Sedki. "Variable selection in model-based clustering and discriminant analysis with a regularization approach". In: *Advances in Data Analysis and Classification* 13.1 (2019), pp. 259–278.

[34] A. Cerioli, L. A. García-Escudero, A. Mayo-Iscar, and M. Riani. "Finding the number of normal groups in model-based clustering via constrained likelihoods". In: *Journal of Computational and Graphical Statistics* 27.2 (2018), pp. 404–416.

[35] A. Cerioli, M. Riani, A. C. Atkinson, and A. Corbellini. "The power of monitoring: how to make the most of a contaminated multivariate sample". In: *Statistical Methods & Applications* 27.4 (2018), pp. 661–666.

[36] A. Cerioli, A. Farcomeni, and M. Riani. "Wild adaptive trimming for robust estimation and cluster analysis". In: *Scandinavian Journal of Statistics* 46.1 (2019), pp. 235–256.

[37] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection". In: *ACM Computing Surveys* 41.3 (2009), pp. 1–58.

[38] W.-C. Chang. "On Using Principal Components Before Separating a Mixture of Two Multivariate Normal Distributions". In: *Applied Statistics* 32.3 (1983), p. 267.

[39] L. H. Chiang and R. J. Pell. "Genetic algorithms combined with discriminant analysis for key variable identification". In: *Journal of Process Control* 14.2 (2004), pp. 143–155.

[40] J. Chiquet, M. Mariadassou, and S. Robin. "Variational inference for probabilistic Poisson PCA". In: *The Annals of Applied Statistics* 12.4 (2018), pp. 2674–2698.

[41] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297.

[42] J. A. Cuesta-Albertos, A. Gordaliza, and C. Matrán. "Trimmed k-means: An attempt to robustify quantizers". In: *Annals of Statistics* 25.2 (1997), pp. 553–576.

[43] M Dash and H Liu. "Feature selection for classification". In: *Intelligent Data Analysis* 1.1-4 (1997), pp. 131–156.

[44] N. E. Day. "Estimating the components of a mixture of normal distributions". In: *Biometrika* 56.3 (1969), pp. 463–474.

[45] N. Dean, T. B. Murphy, and G. Downey. "Using unlabelled data to update classification rules with applications in food authenticity studies". In: *Journal of the Royal Statistical Society. Series C: Applied Statistics* 55.1 (2006), pp. 1–14.

[46] A Dempster, N. Laird, and D Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society* 39.1 (1977), pp. 1–38.

[47] K. A. J. Doherty, R. G. Adams, and N. Davey. "Unsupervised learning with normalised data and non-Euclidean norms". In: *Applied Soft Computing* 7.1 (2007), pp. 203–210.

[48] F. Dotto and A. Farcomeni. "Robust inference for parsimonious model-based clustering". In: *Journal of Statistical Computation and Simulation* 89.3 (2019), pp. 414–442.

[49] F. Dotto, A. Farcomeni, L. A. García-Escudero, and A. Mayo-Iscar. "A reweighting approach to robust clustering". In: *Statistics and Computing* 28.2 (2018), pp. 477–493.

[50] G. Downey. "Authentication of food and food ingredients by near infrared spectroscopy". In: *Journal of Near Infrared Spectroscopy* 4.1 (1996), p. 47.

[51] J. A. Fernández Pierna, P. Volery, R. Besson, V. Baeten, and P. Dardenne. "Classification of Modified Starches by Fourier Transform Infrared Spectroscopy Using Support Vector Machines". In: *Journal of Agricultural and Food Chemistry* 53.17 (2005), pp. 6581–6585.

[52] J. A. Fernández Pierna and P. Dardenne. "Chemometric contest at 'Chimiométrie 2005': A discrimination study". In: *Chemometrics and Intelligent Laboratory Systems* 86.2 (2007), pp. 219–223.

[53] Y. Fong, Y. Huang, P. B. Gilbert, and S. R. Permar. "chngpt: threshold regression model estimation and inference". In: *BMC Bioinformatics* 18.1 (2017), p. 454.

[54] M Forina, C Armanino, M Castino, and M Ubigli. "Multivariate data analysis as a discriminating method of the origin of wines". In: *Vitis* 25.3 (1986), pp. 189–201.

[55] C. Fraley and A. E. Raftery. "Model-based clustering, discriminant analysis, and density estimation". In: *Journal of the American Statistical Association* 97.458 (2002), pp. 611–631.

[56] B. Frénay and M. Verleysen. "Classification in the presence of label noise: A survey". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.5 (2014), pp. 845–869.

[57] B. Frénay, G. Doquire, and M. Verleysen. "Estimating mutual information for feature selection in the presence of label noise". In: *Computational Statistics and Data Analysis* 71 (2014), pp. 832–848.

[58] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[59] J. H. Friedman. "Regularized Discriminant Analysis". In: *Journal of the American Statistical Association* 84.405 (1989), pp. 165–175.

[60] H. Fritz, L. A. García-Escudero, and A. Mayo-Iscar. "tclust : An R Package for a Trimming Approach to Cluster Analysis". In: *Journal of Statistical Software* 47.12 (2012), pp. 1–26.

[61] H. Fritz, L. A. García-Escudero, and A. Mayo-Iscar. "A fast algorithm for robust constrained clustering". In: *Computational Statistics and Data Analysis* 61 (2013), pp. 124–136.

[62] M. T. Gallegos. "Maximum likelihood clustering with outliers". In: *Classification, Clustering, and Data Analysis*. Springer, 2002, pp. 247–255.

[63] M. T. Gallegos and G. Ritter. "A robust method for cluster analysis". In: *The Annals of Statistics* 33.1 (2005), pp. 347–380.

[64] M. T. Gallegos and G. Ritter. "Trimmed ML Estimation of Contaminated Mixtures". In: *Sankhyā: The Indian Journal of Statistics, Series A (2008-)* 71.2 (2009), pp. 164–220.

[65] M. T. Gallegos and G. Ritter. "Using combinatorial optimization in model-based trimmed clustering with cardinality constraints". In: *Computational Statistics & Data Analysis* 54.3 (2010), pp. 637–654.

[66] L. A. García-Escudero, A Gordaliza, C Matrán, and A Mayo-Iscar. "Exploring the number of groups in robust model-based clustering". In: *Statistics and Computing* 21.4 (2011), pp. 585–599.

[67] L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. "Avoiding spurious local maximizers in mixture modeling". In: *Statistics and Computing* 25.3 (2015), pp. 619–633.

[68] L. A. García-Escudero, A. Gordaliza, F. Greselin, S. Ingrassia, and A. Mayo-Iscar. "Robust estimation of mixtures of regressions with random covariates, via trimming and constraints". In: *Statistics and Computing* 27.2 (2017), pp. 377–402.

[69] L. García-Escudero, A. Gordaliza, A. Mayo-Iscar, and R. San Martín. "Robust clusterwise linear regression through trimming". In: *Computational Statistics & Data Analysis* 54.12 (2010), pp. 3057–3069.

[70] L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. "A general trimming approach to robust cluster Analysis". In: *The Annals of Statistics* 36.3 (2008), pp. 1324–1345.

[71] L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. "A review of robust clustering methods". In: *Advances in Data Analysis and Classification* 4.2-3 (2010), pp. 89–109.

[72] L. A. García-Escudero, A. Gordaliza, F. Greselin, S. Ingrassia, and A. Mayo-Iscar. "The joint role of trimming and constraints in robust estimation for mixtures of Gaussian factor analyzers". In: *Computational Statistics & Data Analysis* 99 (2016), pp. 131–147.

[73] L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. "Comments on "The power of monitoring: how to make the most of a contaminated multivariate sample"". In: *Statistical Methods & Applications* 27.4 (2018), pp. 661–666.

[74] L. A. García-Escudero, A. Gordaliza, F. Greselin, S. Ingrassia, and A. Mayo-Iscar. "Eigenvalues and constraints in mixture modeling: geometric and computational issues". In: *Advances in Data Analysis and Classification* 12.2 (2018), pp. 203–233.

[75] L. A. García-Escudero, F. Greselin, and A. M. Iscar. "Robust, fuzzy, and parsimonious clustering, based on mixtures of Factor Analyzers". In: *International Journal of Approximate Reasoning* 94 (2018), pp. 60–75.

[76] A. Gordaliza. "On the breakdown point of multivariate location estimators based on trimming procedures". In: *Statistics & Probability Letters* 11.5 (1991), pp. 387–394.

[77] A. Gordaliza. "Best approximations to random variables based on trimming procedures". In: *Journal of Approximation Theory* 64.2 (1991), pp. 162–180.

[78] L. Greco and C. Agostinelli. "Weighted likelihood mixture modeling and model-based clustering". In: *Statistics and Computing* (2019).

[79] F Greselin and S Ingrassia. "Maximum likelihood estimation in constrained parameter spaces for mixtures of factor analyzers". In: *Statistics and Computing* 25.2 (2015), pp. 215–226.

[80] I. Guyon, C. Aliferis, and Others. "Causal feature selection". In: *Computational methods of feature selection*. Chapman and Hall/CRC, 2007, pp. 79–102.

[81] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics*. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: Wiley, 2005.

[82] T. Hastie and R. Tibshirani. "Discriminant analysis by Gaussian mixtures". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 155–176.

[83] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Vol. 44. Springer Series in Statistics 8. New York, NY: Springer New York, 2009.

[84] R. J. Hathaway. "A Constrained Formulation of Maximum-Likelihood Estimation for Normal Mixture Distributions". In: *The Annals of Statistics* 13.2 (1985), pp. 795–800.

[85] D. M. Hawkins. *Identification of Outliers*. Dordrecht: Springer Netherlands, 1980.

[86] D. M. Hawkins and G. J. McLachlan. "High-breakdown linear discriminant analysis". In: *Journal of the American Statistical Association* 92.437 (1997), p. 136.

[87] X. He and W. K. Fung. "High Breakdown Estimation for Multiple Populations with Applications to Discriminant Analysis". In: *Journal of Multivariate Analysis* 72.2 (2000), pp. 151–162.

[88] C. Hennig. "Breakdown points for maximum likelihood estimators of location-scale mixtures". In: *The Annals of Statistics* 32.4 (2004), pp. 1313–1340.

[89] R. J. Hickey. "Noise modelling and evaluating learning from examples". In: *Artificial Intelligence* 82.1-2 (1996), pp. 157–179.

[90] M. Hubert and K. Van Driessen. "Fast and robust discriminant analysis". In: *Computational Statistics & Data Analysis* 45.2 (2004), pp. 301–320.

[91] M. Hubert, P. J. Rousseeuw, and K. Vanden Branden. "ROBPCA: A New Approach to Robust Principal Component Analysis". In: *Technometrics* 47.1 (2005), pp. 64–79.

[92] M. Hubert, M. Debruyne, and P. J. Rousseeuw. "Minimum covariance determinant and extensions". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 10.3 (2018), pp. 1–11.

[93] U. Indahl and T. Næs. "A variable selection strategy for supervised classification with continuous spectroscopic data". In: *Journal of Chemometrics* 18.2 (2004), pp. 53–61.

[94] S. Ingrassia. "A likelihood-based constrained algorithm for multivariate normal mixture models". In: *Statistical Methods and Applications* 13.2 (2004), pp. 151–166.

[95] S. Ingrassia and R. Rocci. "Degeneracy of the EM algorithm for the MLE of multivariate Gaussian mixtures and dynamic constraints". In: *Computational Statistics & Data Analysis* 55.4 (2011), pp. 1715–1725.

[96] R. S. Jackson. *Wine science: Principles and application*. Academic press, 2008, p. 789.

[97] G. H. John, R. Kohavi, and K. Pfleger. "Irrelevant Features and the Subset Selection Problem". In: *Machine Learning Proceedings 1994*. 1994.

[98] R. A. Johnson, D. W. Wichern, and Others. *Applied multivariate statistical analysis*. Vol. 5. 8. Prentice hall Upper Saddle River, NJ, 2002.

[99] N. Kasabov and Shaoning Pang. "Transductive support vector machines and applications in bioinformatics for promoter recognition". In: *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*. Vol. 1. 2. IEEE, 2003, 1–6 Vol.1.

[100] R. E. Kass. "Bayes Factors in Practice". In: *The Statistician* 42.5 (1993), p. 551.

[101] R. E. Kass and A. E. Raftery. "Bayes Factors". In: *Journal of the American Statistical Association* 90.430 (1995), p. 773.

[102] J. D. Kelly, C. Petisco, and G. Downey. "Application of Fourier transform midinfrared spectroscopy to the discrimination between Irish artisanal honey and such honey adulterated with various sugar syrups". In: *Journal of Agricultural and Food Chemistry* 54.17 (2006), pp. 6166–6171.

[103] R. Kohavi and G. H. John. "Wrappers for feature subset selection". In: *Artificial Intelligence* 97.1-2 (1997), pp. 273–324.

[104] H.-P. Kriegel, E. Schubert, and A. Zimek. "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?" In: *Knowledge and Information Systems* 52.2 (2017), pp. 341–378.

[105] E. Krusińska and J. Liebhart. "Robust Selection of the Most Discriminative Variables in the Dichotomous Problem with Application to Some Respiratory Disease Data". In: *Biometrical Journal* 30.3 (1988), pp. 295–303.

[106] P. Lagacherie and S. Holmes. "Addressing geographical data errors in a classification tree for soil unit prediction". In: *International Journal of Geographical Information Science* 11.2 (1997), pp. 183–198.

[107] S. X. Lee and G. J. McLachlan. "Finite mixtures of canonical fundamental skew t-distributions: The unification of the restricted and unrestricted skew t-mixture models". In: *Statistics and Computing* 26.3 (2016), pp. 573–589.

[108] M. Li, S. Xiang, and W. Yao. "Robust estimation of the number of components for mixtures of linear regression models". In: *Computational Statistics* 31.4 (2016), pp. 1539–1555.

[109] H. Liu and H. Motoda. *Computational methods of feature selection*. CRC Press, 2007.

[110] M Maechler, P Rousseeuw, A Struyf, M Hubert, and K Hornik. *cluster: Cluster Analysis Basics and Extensions*. 2017.

[111] O. Maimon and L. Rokach. "Data mining and knowledge discovery handbook". In: (2005).

[112] R. Maitra. "Initializing partition-optimization algorithms". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 6.1 (2009), pp. 144–157.

[113] A. Malossini, E. Blanzieri, and R. T. Ng. "Detecting potential labeling errors in microarrays by data perturbation". In: *Bioinformatics* 22.17 (2006), pp. 2114–2121.

[114] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press London; New York, 1979, xv, 521 p. :

[115] M. Markou and S. Singh. "Novelty detection: a review—part 1: statistical approaches". In: *Signal Processing* 83.12 (2003), pp. 2481–2497.

[116] R. Maronna and P. M. Jacovkis. "Multivariate clustering procedures with variable metrics". In: *Biometrics* 30.3 (1974), p. 499.

[117] C Maugis, G Celeux, and M. L. Martin-Magniette. "Variable selection in model-based clustering: A general variable role modeling". In: *Computational Statistics & Data Analysis* 53.11 (2009), pp. 3872–3882.

[118] C. Maugis, G. Celeux, and M. L. Martin-Magniette. "Variable selection in model-based discriminant analysis". In: *Journal of Multivariate Analysis* 102.10 (2011), pp. 1374–1387.

[119] C. Maugis, G. Celeux, and M.-L. Martin-Magniette. "Variable Selection for Clustering with Gaussian Mixture Models". In: *Biometrics* 65.3 (2009), pp. 701–709.

[120] G. J. McLachlan, R. W. Bean, and L. Ben-Tovim Jones. "Extension of the mixture of factor analyzers model to incorporate the multivariate t-distribution". In: *Computational Statistics and Data Analysis* 51.11 (2007), pp. 5327–5338.

[121] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004.

[122] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Vol. 544. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1992.

[123] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions, 2E*. Vol. 54. Wiley Series in Probability and Statistics 1. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2008, p. 395.

[124] G. J. McLachlan and D. Peel. "Robust cluster analysis via mixtures of multivariate t-distributions". In: *Advances in Pattern Recognition*. Ed. by A. Amin, D. Dori, P. Pudil, and H. Freeman. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 658–666.

[125] G. J. Mclachlan and S. Rathnayake. "On the number of components in a Gaussian mixture model". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.5 (2014), pp. 341–355.

[126] P. D. McNicholas. *Mixture Model-Based Classification*. Chapman and Hall/CRC, 2016.

[127] P. D. McNicholas, K. R. Jampani, A. F. McDaid, T. B. Murphy, and L. Banks. *pgmm: Parsimonious Gaussian Mixture Models*. 2011.

[128] P. D. McNicholas and T. B. Murphy. "Parsimonious Gaussian mixture models". In: *Statistics and Computing* 18.3 (2008), pp. 285–296.

[129] P. McNicholas, T. Murphy, A. McDaid, and D. Frost. "Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models". In: *Computational Statistics & Data Analysis* 54.3 (2010), pp. 711–723.

[130] G. Menardi. "Density-based Silhouette diagnostics for clustering methods". In: *Statistics and Computing* 21.3 (2011), pp. 295–308.

[131] V. Mezzasalma, A. Sandionigi, I. Bruni, A. Bruno, G. Lovicu, M. Casiraghi, and M. Labra. "Grape microbiome as a reliable and persistent signature of field origin and environmental conditions in Cannonau wine production". In: *PLOS ONE* 12.9 (2017). Ed. by M. Scali, e0184615.

[132] V. Mezzasalma, A. Sandionigi, L. Guzzetti, A. Galimberti, M. S. Grando, J. Tardaguila, and M. Labra. "Geographical and Cultivar Features Differentiate Grape Microbiota in Northern Italy and Spain Vineyards". In: *Frontiers in Microbiology* 9.MAY (2018), pp. 1–13.

[133] D. Miller and J. Browning. "A mixture model framework for class discovery and outlier detection in mixed labeled/unlabeled data sets". In: *2003 IEEE XIII Workshop on Neural Networks for Signal Processing (IEEE Cat. No.03TH8718)*. Vol. 2003-Janua. IEEE, 2003, pp. 489–498.

[134] T. M. Mitchell. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[135] T. B. Murphy, N. Dean, and A. E. Raftery. "Variable selection and updating in model-based discriminant analysis for high dimensional data with food authenticity applications". In: *The Annals of Applied Statistics* 4.1 (2010), pp. 396–421.

[136] S. Newcomb. "A Generalized Theory of the Combination of Observations so as to Obtain the Best Result". In: *American Journal of Mathematics* 8.4 (1886), pp. 343–366.

[137] N. M. Neykov, P. Filzmoser, R. I. Dimova, and P. N. Neytchev. "Robust fitting of mixtures using the trimmed likelihood estimator". In: *Computational Statistics & Data Analysis* 52.1 (2007), pp. 299–308.

[138] J. Pacheco, S. Casado, L. Núñez, and O. Gómez. "Analysis of new variable selection methods for discriminant analysis". In: *Computational Statistics and Data Analysis* 51.3 (2006), pp. 1463–1478.

[139] K. Pearson. "Contributions to the mathematical theory of evolution". In: *Philosophical Transactions of the Royal Society of London. A* 185 (1894), pp. 71–110.

[140] D Peel and G. J. McLachlan. "Robust mixture modelling using the t distribution". In: *Statistics and Computing* 10.4 (2000), pp. 339–348.

[141] R. C. Prati, J. Luengo, and F. Herrera. "Emerging topics and challenges of learning from noisy data in nonstandard classification: a survey beyond binary class noise". In: *Knowledge and Information Systems* 60.1 (2019), pp. 63–97.

[142] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[143] R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2018.

[144] A. Raftery, J. Hoeting, C. Volinsky, I. Painter, and K. Y. Yeung. *BMA: Bayesian Model Averaging*. 2018.

[145] A. E. Raftery and N. Dean. "Variable selection for model-based clustering". In: *Journal of the American Statistical Association* 101.473 (2006), pp. 168–178.

[146] W. M. Rand. "Objective criteria for the evaluation of clustering methods". In: *Journal of the American Statistical Association* 66.336 (1971), p. 846.

[147] L. M. Reid, C. P. O'Donnell, and G. Downey. *Recent technological advances for the determination of food authenticity*. 2006.

[148] G. Ritter. *Robust Cluster Analysis and Variable Selection*. Chapman and Hall/CRC, 2014.

[149] S. J. Roberts, R. Everson, and I. Rezek. "Maximum certainty data partitioning". In: *Pattern Recognition* 33.5 (2000), pp. 833–839.

[150] P. J. Rousseeuw. "Least Median of Squares Regression". In: *Journal of the American Statistical Association* 79.388 (1984), pp. 871–880.

[151] P. J. Rousseeuw and K. V. Driessen. "A fast algorithm for the minimum covariance determinant estimator". In: *Technometrics* 41.3 (1999), pp. 212–223.

[152] N. Russell, L. Cribbin, and T. B. Murphy. "upclass: An R Package for updating model-based classification rules". In: *Cran. R-Project. Org.* (2014).

[153] Y. Saeys, I. Inza, and P. Larranaga. "A review of feature selection techniques in bioinformatics". In: *Bioinformatics* 23.19 (2007), pp. 2507–2517.

[154] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. "Support vector method for novelty detection". In: *Advances In Neural Information Processing Systems* 12 (2000), pp. 582–588.

[155] G. Schwarz. "Estimating the dimension of a model". In: *The Annals of Statistics* 6.2 (1978), pp. 461–464.

[156] L. Scrucca and A. E. Raftery. "Improved initialisation of model-based clustering using Gaussian hierarchical partitions". In: *Advances in Data Analysis and Classification* 9.4 (2015), pp. 447–460.

[157] L. Scrucca, M. Fop, B. Murphy, T., and E. Raftery, Adrian. "mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models". In: *The R Journal* 8.1 (2016), p. 289.

[158] Shaoning Pang and N. Kasabov. "Inductive vs transductive inference, global vs local models: SVM, TSVM, and SVMT for gene expression classification problems". In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. Vol. 2. IEEE, 2004, pp. 1197–1202.

[159] T. Shi and S. Horvath. "Unsupervised learning with random forest predictors". In: *Journal of Computational and Graphical Statistics* 15.1 (2006), pp. 118–138.

[160] A. Strehl and J. Ghosh. "Cluster ensembles - A knowledge reuse framework for combining multiple partitions". In: *Journal of Machine Learning Research*. 2003.

[161] D. M. J. Tax and R. P. W. Duin. "Outlier detection using classifier instability". In: *Advances in Pattern Recognition*. Ed. by A. Amin, D. Dori, P. Pudil, and H. Freeman. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 593–601.

[162] G. Thomson. "The factorial analysis of human ability". In: *British Journal of Educational Psychology* 9.2 (1939), pp. 188–195.

[163] V. Todorov. "Robust selection of variables in linear discriminant analysis". In: *Statistical Methods and Applications* 15.3 (2007), pp. 395–407.

[164] V. Todorov and P. Filzmoser. "An Object-Oriented Framework for Robust Multivariate Analysis". In: *Journal of Statistical Software* 32.3 (2009), pp. 1–47.

[165] K. Vanden Branden and M. Hubert. "Robust classification in high dimensions based on the SIMCA Method". In: *Chemometrics and Intelligent Laboratory Systems* 79.1-2 (2005), pp. 10–21.

[166] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Vol. 3. September. New York, NY: Springer New York, 2000.

[167] T. Vatanen, M. Kuusela, E. Malmi, T. Raiko, T. Aaltonen, and Y. Nagai. "Semi-supervised detection of collective anomalies with an application in high energy particle physics". In: *Proceedings of the International Joint Conference on Neural Networks*. 2012.

[168] L. Waldron. "Data and Statistical Methods To Analyze the Human Microbiome". In: *mSystems* 3.2 (2018), pp. 1–4.

[169] M. A. Wolters. "A Genetic Algorithm for Selection of Fixed-Size Subsets with Application to Design Problems". In: *Journal of Statistical Software* 68.Code Snippet 1 (2015).

[170] X. Wu. *Knowledge acquisition from databases*. Westport, CT, USA: Intellect books, 1995.

[171] L. Yu. "Feature selection for genomic data analysis". In: *Computational methods of feature selection* (2008), pp. 337–353.

[172] L. Yu and H. Liu. "Efficient feature selection via analysis of relevance and redundancy". In: *Journal of Machine Learning Research* 5 (2004), pp. 1205–1224.

[173] W. Zhang, R. Rekaya, and K. Bertrand. "A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: Application to human breast cancer". In: *Bioinformatics* 22.3 (2006), pp. 317–325.

[174] X. Zhu and X. Wu. "Class noise vs. attribute noise: A quantitative study". In: *Artificial Intelligence Review* 22.3 (2004), pp. 177–210.