A Critical Look at the Music Classification Experiment Pipeline: Using Interventions to Detect and Account for Confounding Effects

by

Francisco RODRÍGUEZ-ALGARRA

Submitted in partial fulfillment of the requirements
of the Degree of Doctor of Philosophy

London, UK

June 2020

AUTHOR'S DECLARATION

I, Francisco Rodríguez-Algarra, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author.

Francisco Rodríguez-Algarra
16th November, 2019

Collaborations and publications

The following publications directly relate to the research reported in this dissertation:

RODRÍGUEZ-ALGARRA, F., B. L. Sturm, and H. Maruri-Aguilar (2016). "Analysing Scattering-Based Music Classification Systems: Where's the Music?" In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 344–350

RODRÍGUEZ-ALGARRA, F., B. L. Sturm, and S. Dixon (2019). "Characterising Confounding Effects in Music Classification Experiments through Interventions". *Transactions of the International Society for Music Information Retrieval*, 2(1), pp. 52–66

Details about the contents of each publication and the contributions of each collaborator are provided in Sec. 1.3.

ABSTRACT

This dissertation focuses on the problem of confounding in the design and analysis of music classification experiments. Classification experiments dominate evaluation of music content analysis systems and methods, but achieving high performance on such experiments does not guarantee systems properly address the intended problem. The research presented here proposes and illustrates modifications to the conventional experimental pipeline, which aim at improving the understanding of the evaluated systems and methods, facilitating valid conclusions on their suitability for the target problem.

Firstly, multiple analyses are conducted to determine which cues scattering-based systems use to predict the annotations of the *GTZAN* music genre collection. In-depth system analysis informs empirical approaches that alter the experimental pipeline. In particular, deflation manipulations and targeted interventions on the partitioning strategy, the learning algorithm and the frequency content of the data reveal that systems using scattering-based features exploit faults in *GTZAN* and previously unknown information at inaudible frequencies.

Secondly, the use of interventions on the experimental pipeline is extended and systematised to a procedure for characterising effects of confounding information in the results of classification experiments. Regulated bootstrap, a novel resampling strategy, is proposed to address challenges associated with interventions dealing with partitioning. The procedure is demonstrated on *GTZAN*, analysing the effect of artist replication and infrasonic information on performance measurements using a wide range of system-construction methods.

Finally, mathematical models relating measurements from classification experiments and potentially contributing factors are proposed and discussed. Such models enable decomposing measurements into contributions of interest, which may differ depending on the goals of the study, including those from pipeline interventions. The adequacy for classification experiments of some conventional assumptions underlying such models is also examined.

The reported research highlights the need for evaluation procedures that go beyond performance maximisation. Accounting for the effects of confounding information using procedures grounded on the principles of experimental design promises to facilitate the development of systems that generalise beyond the restricted experimental settings.

ACKNOWLEDGEMENTS

This dissertation would have been impossible without the help, support and patience of countless people. First and foremost, I would like to thank Dr Bob L. Sturm, who initiated this project and inspired most of my research, and Prof. Simon Dixon, who managed to guide me to the finish line with relentless encouragement and wonderful advice. Their passion and commitment to rigorous research has been a constant source of inspiration for me during these past years. I will forever be grateful for their ideas, insights, and trust in what I could achieve. Their example has been invaluable. They showed me how to overcome my fears and the path to becoming a scientist. I cannot express how much I appreciate having had the opportunity to work with them.

I would also like to thank Dr Hugo Maruri-Aguilar for helping me understand what at the time was a completely new discipline for me, in both lectures and meetings. He remained positive even when I was clueless, always providing thoughtful and useful advice that has undoubtedly improved my work. Dr Joakim Andén and Dr Vincent Lostanlen also deserve my thanks for helping me comprehend their research, and Prof. Geraint Wiggins for his constructive and always illuminating feedback.

One of the most enjoyable and rewarding experiences of these last few years in Queen Mary has been demonstrating. Along with Dr Sturm and Prof. Dixon, I would like to thank Dr Nikos Tzevelekos for giving me the opportunity to help in labs and lectures, as well as Dr Paula Fonseca for trusting me so often to assist her.

I would have been unable to get where I am right now without the amazing friends I made in C4DM, and those that have been around whenever I needed to cheer up. The laughter during our game nights or a simple chat in front of a drink have been a bright light in my life even during the darker moments. I am extremely grateful for having crossed paths with Veronica, Maria, Mi, Daniel, Beici, Manos, Saumitra, Bhusan, Adán, Chris, Chunyang, Yading, and Juan, among others. Whatever happens in the future, whichever paths each of us follows from now on, I will always cherish the joy they brought to my life.

I have been fortunate enough to be given the chance to join the Blizard Institute to conduct fascinating research even before finishing my PhD. My most sincere gratitude to Prof. Vardhman Rakyan and Dr Robert Lowe for believing in me and being so understanding since I joined. I would also like to thank my incredible current and former lab mates Sarah, Amy, Rob, Ama, Zak and Selin, who have already taught me so much and make each day more and more exciting.

Last but not least, I could not be more grateful for the wonderful family that I have, for the unconditional love and support of those that remain and those that sadly left. There are no words to express how lucky I am to have the kind, caring, and trusting parents that I have, how much I owe them for all they have done for me, how much I love them, how much I miss them every day. I would also like to thank Siying, who has travelled by my side for most of the path. She has believed in me much more than I ever could. Her optimism and care when I felt overwhelmed, her encouragement and advice when I saw no way out, will always remain in my heart, no matter what.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Acronyms: Academic Disciplines and Conferences

AI	Artificial Intelligence
DoE	Design of Experiments
ICML	International Conference on Machine Learning
IR	(Text) Information Retrieval
IRT	Item Response Theory
ISMIR	International Society for Music Information Retrieval (formerly International Symposium for Music Information Retrieval)
MCA	Music Content Analysis
MIR	Music Information Retrieval
MIREX	Music Information Retrieval Evaluation eXchange
ML	Machine Learning
TREC	Text REtrieval Conference

Acronyms: Music Informatics and Signal Processing

DC	Direct Current
FB	FilterBank
FFT	Fast Fourier Transform
GFCC	Gammatone-Frequency Cepstral Coefficients
HPCP	Harmonic Pitch Class Profile
IIR	Infinite Impulse Response (filterbank)
MER	Music Emotion Recognition
MFCC	Mel-Frequency Cepstral Coefficients
MGR	Music Genre Recognition
NPR	Near-Perfect Reconstruction (filterbank)

Acronyms: Machine Learning

ABDT	AdaBoosted Decision Trees
BDT	Binary Decision Trees
CNN	Convolutional Neural Network
CV	Cross-Validation
DT	Decision Trees
ER	Error Rate
K-NN	K-Nearest Neighbours
MLP	Multi-Layer Perceptron
NB	Naive Bayes
RF	Random Forest
SVM	Support Vector Machine

Acronyms: Statistics and Experimental Design

ANOVA	ANalysis Of VAriance
BFD	Blocked Factorial Design
CBD	Complete Block Design
CRD	Completely Randomised Design
CSS	Crude Sum of Squares
df	Degrees of Freedom
FD	Factorial Design
GBD	Generalised Blocked Design
GBFD	Generalised Blocked Factorial Design
HSD	(Tukey's) Honestly Significant Differences
MS	Mean Squares
SE	Standard Error
SS	Sum of Squares
VR	Variance Ratio

Abbreviations: Function Names

$acc(\cdot)$	Classification accuracy
$dim(\cdot)$	Dimensionality

$err(\cdot)$	Error rate
$log(\cdot)$	Natural logarithm
$prec(\cdot)$	Precision
$rec(\cdot)$	Recall

Abbreviations: Feature Representations

`Mel Sc.`	Mel-frequency spectrogram features
`1-L Sc.`	First-order time-scattering features
`1&2-L Sc.`	First- and second-order time-scattering features
`TF Sc.`	First- and second-order time-frequency scattering features
`TF Adap. Sc.`	First- and second-order time-frequency-adaptive scattering features
`1,2&3-L Sc.`	First-, second-, and third-order time-scattering features
`Des. 1-L Sc.`	Descriptive statistics of first-order time-scattering features
`Tim+Dyn`	Features related with timbre and dynamics from `Essentia`

Abbreviations: Evaluation Conditions

`pr`	Pruned collection
`filt`	High-pass filtered recordings

Mathematical Symbols: Statistics and Experimental Design

$\mathcal{F}, \mathcal{G}, \mathcal{H}$	Generic factor variables
$\mathscr{F}$	A set of non-equivalent factors
$V_{\mathcal{F}}, W_{\mathcal{F}}$	Subspaces associated with factor $\mathcal{F}$
$\mathbf{R}_{\mathcal{F}}, \mathbf{P}_{\mathcal{F}}$	Respectively, relation and projection matrices associated with factor $\mathcal{F}$
$\mathcal{F} \wedge \mathcal{G}$	Infimum of factors $\mathcal{F}$ and $\mathcal{G}$
$\mathcal{F} \vee \mathcal{G}$	Supremum of factors $\mathcal{F}$ and $\mathcal{G}$
$\perp$	Vector orthogonality
$\perp\!\!\!\perp$	Factor orthogonality
$\oplus$	Orthogonal direct sum of vector spaces
Ω	Set of all observational units (or "plots") ω in an experiment
$\mathscr{T}$	Set of all treatments in an experiment
$\mathcal{T}$	A treatment factor
τ	The fixed effect of a treatment factor

$\mathcal{B}$	A blocking factor
β	The random effect of a blocking factor
ε	Residual of a structural model
$\mathcal{E}$	The equality factor
μ	Mean; Benchmark parameter of a structural model
$\mathcal{U}$	The universal factor
σ	Standard deviation
$\mathcal{N}(\mu, \sigma^2)$	A Gaussian distribution with mean μ and variance σ^2
χ^2_{df}	A Chi-squared distribution with df degrees of freedom

Mathematical Symbols: Music Content Analysis Systems and Experimental Pipeline

Θ	A music universe, with elements θ
$\mathcal{R}_\Theta$	A music recording universe, with elements r_θ
$\mathcal{U}_{\mathcal{V},A}$	A semantic universe, with vocabulary of tokens $\mathcal{V}$ and semantic rules A
$\mathcal{U}_{\mathbb{F},A'}$	A semantic feature universe, with feature space $\mathbb{F}$ and semantic rules A'
e	A feature extractor
p	A predictor (classifier or regressor)
s	A prediction system
ℓ	A learning algorithm
m	A system-construction method
r	A raw data instance (a recording)
$\boldsymbol{R}$	A collection of raw data instances
a	An annotation
$\mathtt{a}$	A particular value of an annotation a
$\boldsymbol{A}$	A collection of annotations
f	A feature representation
$\boldsymbol{F}$	A collection of feature representations
$\phi(\cdot)$	A performance metric function
y	A performance value
$\mathcal{Y}$	Distribution of performance values
$\psi(\cdot)$	An assignment function
k	Assignment index

$\boldsymbol{C}$	An annotated music collection of N instances $c_n = (r_n, a_n)$
$\boldsymbol{D}$	A dataset of N instances $d_n = (f_n, a_n)$
$\mathcal{X}$	Feature extraction algorithm factor variable, with X levels
$\mathcal{L}$	Learning algorithm factor variable, with L levels
$\mathcal{M}$	Method factor variable, with M levels
$\mathcal{S}$	System factor variable, with J levels
$\mathcal{K}$	Sample factor variable, with K levels
$\mathcal{C}$	Collection factor variable, with C levels
$t(\cdot)$	A data transformation function
n_r	A threshold value ("number of recordings")
η_r	A relative threshold value
$\lfloor \cdot \rfloor$	Floor function
z, w	Potential confounders
$\mathtt{z}$	A particular value of a potential confounder z
$\mathcal{Z}$, $\mathcal{W}$	Potential confounder factor variables, with Z and W levels, respectively
$\mathcal{Q}$	Deflation step factor variable
α, κ	Respectively, slope and intersect of a linear model associating performance measurements under different conditions
Δ_R, Δ_A	Respectively, "real" and "accumulated" variation from two interventions
u_i	Loss of prediction i
$\pi(i)$	Probability that loss u_i equals 1
$\{\cdot\}_t$	Version of any variable associated with training
$\{\cdot\}_p$	Version of any variable associated with testing ("prediction")
$\{\cdot\}_h$	Version of any variable associated with held-out data
$\{\hat{\cdot}\}$	Predicted or estimated version of any variable
$\{\cdot\}'$	Intervened version of any variable

CHAPTER 1

INTRODUCTION

1.1 Motivation and Goals

Artificial Intelligence (AI) is taking the world by storm. News of formidable achievements by artificial systems breaks on an almost daily basis. From defeating human champions in complex games[1] to self-driving cars,[2] from writing poetry[3] to the discovery of novel scientific theories,[4] no field seems too challenging. Music is no exception. Not only do streaming services increasingly rely on AI to tailor recommendations for their users based on their taste and listening habits,[5] but some companies also offer automatically composed pieces on demand.[6] These are all undoubtedly extraordinary feats, and extraordinary feats demand extraordinary proof.

Much of the progress witnessed in recent times can be attributed to the widespread adoption of data-driven approaches instead of the more traditional rule-based agents. Algorithms that learn directly from data uncover complex patterns far beyond any human's capability. As a result, however, the internal models such algorithms create are similarly complex for humans to comprehend, so the assessment of their success is often limited to

[1] https://deepmind.com/research/case-studies/alphago-the-story-so-far

[2] https://waymo.com/

[3] https://www.theguardian.com/technology/2016/may/17/googles-ai-write-poetry-stark-dramatic-vogons

[4] https://now.tufts.edu/news-releases/planarian-regeneration-model-discovered-artificial-intelligence

[5] https://www.androidpit.com/music-streaming-and-artificial-intelligence

[6] https://futurism.com/a-new-ai-can-write-music-as-well-as-a-human-composer

whether they generate expected outputs given particular inputs. One either assumes that the means to achieve such outputs, although obscure, must match what an intelligent agent would have performed, or accepts that any means, no matter how alien, is equally valid. A common argument for this latter position defends that engineers managed to make planes fly only when they stopped trying to imitate the way birds fly — alternative means may yield desired outcomes.

Nevertheless, an increasing number of authors have realised that data-driven models may appear successful by exploiting cues that have little to do with the intended problem, but are incidentally associated with the target outcomes within the evaluation environment. Learning algorithms take advantage of any pattern they uncover in the data, irrespective of whether such patterns would appear in real-life scenarios. Presumably irrelevant changes in the input data might cause apparently successful trained models to change their outputs in unexpected ways. Articles in high-impact journals, such as the one Heaven (2019) recently published in Nature, demonstrate an increasing concern about the consequences of this phenomenon.

Music data poses particularly interesting challenges for artificial systems and their evaluation. Since music is an inherently human construct, creating algorithms capable of describing the contents of an audio file using musical concepts similarly to a human suggests a sophisticated intelligence. Grasping musical concepts not only often requires dedicated education, but may also depend on cultural and social cues external to the sound itself. Similar to other fields, some believe that generating outcomes indistinguishable from those from humans evidences sufficient understanding of such musical concepts. Others, such as Wiggins (2009) and Widmer (2016), defend that only methods that adhere to specific cognitive processes and musical knowledge may lead to musical intelligence. In practice, the answer may lie in between, with some use cases requiring different approaches. Conventional evaluation practices, however, fail to provide the information necessary for developers, researchers and users to assess whether systems function in a manner suitable for their intended use case.

The present dissertation discusses and addresses some of the most pressing issues in the evaluation of music analysis systems and methods. Suitable evaluation is fundamental for any discipline, enabling to judge the success of proposed solutions, highlighting

promising paths and contrasting them with alternatives, as well as to keep track of how the discipline progresses. Unsuitable or insufficient evaluation, on the other hand, may lead to completely misguided research paths or dead ends. Identifying the main limitations of the conventional evaluation practices and proposing alternatives that overcome such limitations is paramount. In particular, the research presented in this dissertation targets the following goals:

1. To develop and showcase a systematic evaluation methodology that enables to uncover the reasons behind the performance of music analysis systems, providing valid and relevant information to assess their suitability to specific use cases. Such methodology will require in-depth analyses and domain knowledge to inform suitably targeted experiments, grounded on the formal principles of experimental design. Moreover, the methodology should allow comparisons between alternative systems and methods on their reliance upon particular sources of information. This is essential to distinguish actual solutions from those that only appear to work within the confined experimental setting.
2. To bridge the gap between the language and tools of the statistical Design of Experiments and the evaluation machinery employed to assess systems and methods for the automatic analysis of music data. Despite being vital to reach valid conclusions, music analysis research largely lacks a formal experimental design framework. The present dissertation thus intends to facilitate the translation between experimental practices in the discipline and the concepts and mathematical formalism of statistical Design of Experiments. This includes the identification of contributing factors and the formulation of models that relate them for the analysis of experimental measurements.

Addressing these goals should provide a solid foundation for future empirical studies conducted in the discipline and other applied Machine Learning fields, and, hopefully, facilitate the development of truly impactful solutions.

1.2 Structure of this Dissertation

The remainder of this dissertation is grouped into two main parts plus an overall conclusion. The first part reviews the background knowledge related to the topic of the present

dissertation. It comprises the following chapters:

Chapter 2 reviews the goals of Music Content Analysis research and describes the main evaluation practices employed in the discipline, highlighting their drawbacks and presenting solutions proposed in the literature.

Chapter 3 introduces the fundamental principles and tools of statistical Design of Experiments, presents a particular approach known as the Calculus of Factors, and reviews common means to express and analyse the results of experiments in the evaluation of learning algorithms. Additionally, **Appendix A** includes concrete examples of the Calculus of Factors approach.

The second part of this dissertation reports the specific contributions achieved through original research. It comprises the following chapters:

Chapter 4 describes multiple analyses conducted to determine which cues are used by systems based on a particular feature representation (called the scattering transform) to predict the annotations of the *GTZAN* music genre collection. In-depth system analysis informs empirical approaches that alter the experimental pipeline in two forms: deflation manipulations and targeted interventions. These reveal that such systems exploit faults in the collection and previously unknown information at inaudible frequencies.

Chapter 5 extends and systematises the use of interventions on the experimental pipeline to a procedure for characterising effects of confounding information in the results of classification experiments. Regulated bootstrap, a novel resampling strategy, is proposed to address challenges associated with interventions dealing with partitioning. The procedure is demonstrated on *GTZAN*, analysing the effect of artist replication and infrasonic information on performance measurements using a wide range of system-construction methods.

Chapter 6 proposes mathematical models that relate measurements from classification experiments to potentially contributing factors. Such models enable the decomposition of measurements into contributions of interest, which may differ depending on the goals of the study, including those from pipeline interventions. The suitability for classification experiments of some conventional assumptions underlying

such models is also examined. Additionally, **Appendix B** develops an illustrative example analysis.

Finally, **Chapter 7** provides concluding remarks and suggests future research paths following the work presented in this dissertation.

1.3 Contributions and Collaborations

Much of the research reported in this dissertation has been previously published in peer-reviewed venues, and is the result of joint efforts with collaborators. The following lists such publications and their corresponding contributions, and details who conducted each part of the work.

- **RODRÍGUEZ-ALGARRA, F.**, B. L. Sturm, and H. Maruri-Aguilar (2016). "Analysing Scattering-Based Music Classification Systems: Where's the Music?" In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 344–350

  This article largely corresponds with the study reported in Ch. 4, with the exception of the deflation analyses, which were not included in the paper. The main contributions of the study are the following:

  (i) An in-depth analysis of a state-of-the-art approach for automatic Music Genre Recognition (MGR).

  (ii) An illustration of the use of non-conventional evaluation procedures, namely deflation and intervention experiments, to illuminate the reasons behind performance of prediction systems.

  (iii) Evidence that MGR systems based on the state-of-the-art approach exploit faults of the evaluation collection to perform predictions, including the previously-unknown presence of information at inaudible frequencies.

  I (Francisco Rodríguez-Algarra) performed the system analysis, assisted by Dr Bob L. Sturm. I designed, implemented and conducted the experiments, analysed the results, and wrote the article. Dr Bob L. Sturm supervised and edited the writing. Dr Hugo Maruri-Aguilar provided advice and proofread the text.

- **RODRÍGUEZ-ALGARRA, F.**, B. L. Sturm, and S. Dixon (2019). "Characterising Confounding Effects in Music Classification Experiments through Interventions".

Transactions of the International Society for Music Information Retrieval, 2(1), pp. 52–66

This article closely matches the research reported in Ch. 5, which includes the following contributions:

(i) A discussion regarding the concept of confounding in applied Machine Learning scenarios, and a proposal about its usage.

(ii) A systematic procedure for the analysis of the effects of confounding in the evaluation of applied Machine Learning systems and methods.

(iii) A resampling strategy specifically designed to address certain types of confounding factors.

(iv) An analysis of the effects of confounding factors and their interactions on evaluations using a widely employed benchmarking collection for MGR.

I developed the procedure, defined the regulated bootstrap algorithm, designed, implemented and conducted the case study, analysed the results, and wrote the article. Dr Bob L. Sturm and Prof. Simon Dixon supervised and edited the writing.

Finally, the research reported in Ch. 6 remains unpublished at the time of writing. The main contributions of this study are the following:

(i) A translation of the language and tools of statistical Design of Experiments to the analysis of measurements from applied Machine Learning classification experiments, including illustrations of procedures from the so-called Calculus of Factors approach to experimental design.

(ii) A thorough discussion of the suitability of different structural models for the analysis of measurements from classification experiments in diverse scenarios.

(iii) An assessment of how introducing interventions in classification experiments impacts the analysis of their measurements.

(iv) A proposal for the replacement of linear with logistic structural models to better suit the data obtained from classification experiments.

I proposed the models, conducted the analysis and wrote the text. Dr Bob L. Sturm and Prof. Simon Dixon supervised and edited the writing. Dr Hugo Maruri-Aguilar provided advice and proofread the text.

Part I

Background

CHAPTER 2

EVALUATION IN MUSIC CONTENT ANALYSIS RESEARCH

This chapter reviews the goals of Music Content Analysis (MCA) research and the evaluation practices often employed in the field. Evaluation of any kind will only be appropriate as long as it provides information aligned with the purposes of the study in particular and the discipline in general. No discussion about evaluation methodologies can thus succeed without first understanding which goals the objects to be evaluated aim to achieve. It is paramount to determine which are such purposes in the field of interest to assess whether current evaluation methodologies provide sufficiently valid and reliable information, and which improvements might be necessary otherwise. Since the ultimate goals of each particular study might widely differ, Sec. 2.1 first considers which ones appear most important and delimits the kinds of studies this dissertation covers. Sec. 2.2 then describes evaluation paradigms used in such studies, of which the most widely accepted one is detailed in Sec. 2.3. Sec. 2.4 reports the evaluation of a family of MCA systems from the literature, upon which a case study is built later in this dissertation. Sec. 2.5 reviews the main drawbacks of conventional evaluation practices and some ways forward proposed in the literature. Finally, Sec. 2.6 briefly summarises this review and discusses

its main implications for the research reported in this dissertation.

2.1 Brief Overview of Music Content Analysis Research

This section describes the goals of MCA research, which are first contextualised by briefly reviewing the aims of the broader Music Information Retrieval (MIR) discipline. The particularities of MCA often highlighted in the literature lead to a formal representation of the systems developed in the discipline that is later used throughout this dissertation.

2.1.1 Music Information Retrieval

Broadly speaking, MIR research aims to develop technologies that connect users (listeners, composers, scholars, etc.) with the music, and information about music, that satisfies their particular needs (Casey et al., 2008). MIR is a relatively young multidisciplinary field, usually considered to have its starting point in the International Symposium for Music Information Retrieval (ISMIR) held in 2000. One may trace its roots further back to computer music research (Roads, 1996), but its consolidation as a discipline seems to be linked to the vast amount of musical data that the Internet made widely available. This demands tools that facilitate managing such data.

The data on which MIR focuses comes from various sources, which are usually grouped in two categories (Schedl et al., 2014): music *content* and music *context*. Music *content* largely refers to information that can be extracted directly from the audio signal. Some authors, however, also include symbolic sources such as scores as belonging to the music content space (Casey et al., 2008). Musical facets such as melody, harmony, rhythm, timbre, and so on, naturally fit in this category. Conversely, music *context* comprises aspects related to the music item not intrinsically included in the audio signal or its notated representation. This includes the biography of the artist, the cover art of the album, and the position of the track in the charts.

According to Serra et al. (2013), MIR centers on developing methodologies for *processing*, *representing* and *understanding* music digital data. They seem to agree with Herrera et al. (2009) in considering pure "retrieval" applications secondary in the MIR literature despite the conventional name of the discipline. In this sense, they propose using the alternative denomination Music Information Research.

Schedl et al. (2014), on the other hand, consider that the broad areas of concern of MIR research are (i) the extraction and inference of *meaningful features* from music, (ii) the *indexing* of music using these features, and (iii) the development of *search and retrieval* schemes. They thus place final user applications of the methodologies developed within the community as core targets. In this sense, Schedl et al. (2013) argue that *user properties* (characteristics of the listener, such as their musical preferences) and *user context* (dynamic aspects of the current listening process, such as the listener's location), deserve as much attention as music content and context information.

The extraction of "meaningful features" that Schedl et al. (2014) mentions is key to automatic music analysis (or "description" (Sturm et al., 2014)). This aligns with the development of "representations" to facilitate "understanding" that Serra et al. (2013) describe. Much of MIR addresses this goal, especially from audio data.

2.1.2 Music Content Analysis

Music Content Analysis (MCA) is the branch of MIR concerned with the extraction of information solely from the audio recordings of musical pieces. Other denominations for the same body of research include Music Content Description (Schedl et al., 2014) or Content-Based MIR (Casey et al., 2008). The study of audio data comprises a large, if not the largest, part of the research conducted by the MIR community (Serra et al., 2013), as a quick review of the works presented at ISMIR clearly reveals.

MCA studies target a range of musical, cultural and cognitive concepts, whose representations might be the ultimate goals themselves or used as intermediate annotations for various applications. These applications include music recommendation (Celma, 2010), collection organisation (Stober, 2013), and music practice enhancement tools, such as tutoring systems (Dittmar et al., 2012).

MCA systems generate representations of target concepts automatically, as opposed to the manual annotation adopted by some services, such as Pandora.[1] Hiring human labour is expensive, which makes manual annotation only feasible when dealing with limited music collections. The number of users and size of music collections would often lead

[1] `http://www.pandora.com`

to prohibitive costs both in monetary and temporal terms if one relied solely on human effort.

The challenge in creating MCA systems lies to a large extent on the lack of unambiguous links between what is measurable from the audio samples and how humans experience music. The former, usually referred to as low-level features or descriptors, can be obtained through the application of more or less simple mathematical formulae, often developed within the context of Signal Processing research (Müller et al., 2011). This generates deterministic representations that computers exploit effortlessly but humans struggle to interpret (Serra et al., 2013).

The lower levels of abstraction provide the building blocks for descriptors of both mid- and high-level concepts. The boundary between these is somehow arbitrary, with some concepts belonging to one or the other depending on the particular application considered. Nevertheless, researchers often agree on regarding note pitches and onset locations as mid-level features, placing concepts human employ for casual conversation about music, such as genre (Sturm, 2014d) or emotion (Yang and Chen, 2012), at the top level.

Mapping lower level information becomes increasingly complex as the target concepts approach human understanding levels, with many only feasibly addressed through Machine Learning approaches. Many studies focus on assessing combinations of low-level features and Machine Learning algorithms for their suitability to capture and represent high-level concepts. This is the case of Andén and Mallat (2014), who propose and evaluate Support Vector Machine (SVM) models trained on features derived from the Scattering Transform (Mallat, 2012) for addressing music genre recognition. The case study described in later chapters builds upon this work.

2.1.3 Formalising Music Content Analysis Problems and Systems

This dissertation largely focuses on methodologies to evaluate MCA systems that generate descriptors of high-level concepts from audio signals of music pieces. Expressing formally how such systems work helps disambiguate the problem they address. Sturm et al. (2014) propose one such formalisation, which is represented schematically in Fig. 2.1 and described here.[2]

[2]Some symbols introduced here differ from those in previous publications (i.e., Rodríguez-Algarra et al., 2019; Sturm et al., 2014) to avoid clashes with well-established notation in the DoE literature.

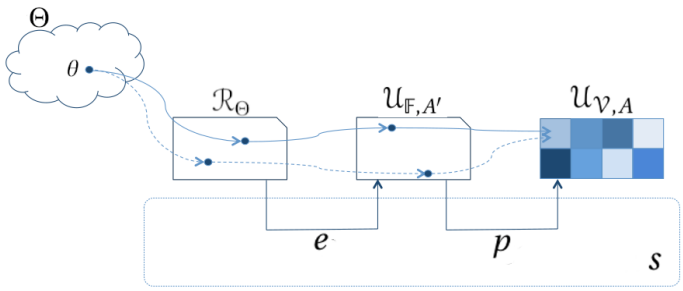


Figure 2.1: Schematic representation of a Music Content Analysis (MCA) system, adapted from Rodríguez-Algarra et al. (2016). Θ is a music universe, with θ an element of that universe; $\mathcal{R}_\Theta$ is a recording universe; $\mathcal{U}_{\mathbb{F},A'}$ is a semantic feature universe, defined by a vocabulary of features $\mathbb{F}$ and semantic rules A'; $\mathcal{U}_{\mathcal{V},A}$ is a semantic universe, defined by a vocabulary of tokens $\mathcal{V}$ and semantic rules A. s is an MCA system, composed of a feature extractor e and a predictor p.

The main goal of an MCA problem is to build a system s that addresses the *use case* specified by a music universe Θ, a tangible music recording universe $\mathcal{R}_\Theta$, a semantic universe $\mathcal{U}_{\mathcal{V},A}$, and a set of success criteria. Θ comprises all music pieces of interest for the problem, such as Viennese music compositions from the late 18th Century; $\mathcal{R}_\Theta$ comprises concrete realisations of the elements of Θ that follow particular specifications, such as mono audio recordings in `wav` format. As Fig. 2.1 shows, each element in Θ may relate to multiple elements in $\mathcal{R}_\Theta$. This is the case if $\mathcal{R}_\Theta$ includes recordings of various performances of the same piece, multiple excerpts sliced from the same recording, or even the same excerpt exposed to diverse signal processing transformations.

The semantic universe $\mathcal{U}_{\mathcal{V},A}$ determines the possible descriptions of the elements of $\mathcal{R}_\Theta$ with respect to the concept (or concepts) of interest. It can be defined as:

$$\mathcal{U}_{\mathcal{V},A} := \{v \in \mathcal{V}^n \mid n \in \mathbb{N} \wedge A(v)\} \tag{2.1}$$

this is, those sequences v of tokens in the vocabulary $\mathcal{V}$ that satisfy a particular semantic rule of the form $A : \mathcal{V}^n \rightarrow \{\mathtt{T},\mathtt{F}\}$. To put it simply, $\mathcal{U}_{\mathcal{V},A}$ represents all acceptable descriptions, which may involve joining multiple tokens for some problems. This is clearly the case in autotagging (Bertin-Mahieux et al., 2008), where multiple labels of distinct nature associate with each instance, but may also apply in other less evident situations. For instance, if the tokens in $\mathcal{V}$ correspond to artist names, then acceptable descriptions should account for possible collaborations between them and permit multiple tokens per

instance. Regarding the hypothetical example of Viennese music during late 18th Century introduced above, collaborations were extremely rare but not unheard of — W. A. Mozart apparently collaborated with J. M. Haydn and A. C. Adlgasser to compose "Die Schuldigkeit des ersten Gebots".[3] For simplicity of illustration, however, we consider in what follows descriptions formed by single tokens, such as "W. A. Mozart" and "F. J. Haydn".

An MCA system s is a map from $\mathcal{R}_\Theta$ to $\mathcal{U}_{\mathcal{V},A}$, i.e., $s : \mathcal{R}_\Theta \rightarrow \mathcal{U}_{\mathcal{V},A}$, such that $s(r_\theta) = v$, with $r_\theta \in \mathcal{R}_\Theta$ a recording, and v an acceptable description according to the rules A defines. In other words, s receives as input a music recording and outputs a semantic label that describes it according to certain criteria. For instance, from recordings in $\mathcal{R}_\Theta$ as in the example above, MCA systems may attempt to identify whether each piece was composed by "W. A. Mozart", "F. J. Haydn" or another member of $\mathcal{U}_{\mathcal{V},A}$ solely from the audio signal.

We can decompose the map s into two maps, an *extractor* $e : \mathcal{R}_\Theta \rightarrow \mathcal{U}_{\mathbb{F},A'}$ and a *predictor* $p : \mathcal{U}_{\mathbb{F},A'} \rightarrow \mathcal{U}_{\mathcal{V},A}$, where $\mathcal{U}_{\mathbb{F},A'}$ indicates the semantic feature universe defined by a vocabulary of features $\mathbb{F}$ and semantic rules $A' : \mathbb{F} \rightarrow \{\mathtt{T},\mathtt{F}\}$. Describing the contents of a music recording with respect to a concept of interest thus involves selecting and computing which signal-level information e should obtain, as well as determining how to integrate such information through the map defined by p — either a "classifier" or "regressor" — to the conceptual domain defined by $\mathcal{U}_{\mathcal{V},A}$. For instance, an s may attempt to estimate the harmonic content of recordings in $\mathcal{R}_\Theta$ using an e that generates Harmonic Pitch Class Profile (HPCP) representations (Gómez, 2006), and then feeding such representations to a p that links particular harmonic patterns with the composer tokens in $\mathcal{U}_{\mathcal{V},A}$.

This formalisation does not force any specific way of constructing systems, but it is hereinafter assumed that p comes from the training of a supervised Machine Learning algorithm, since implementations of this kind dominate the literature. Particularly when targeting mid-level concepts, p can be defined instead as a set of explicit rules derived from expert knowledge. As the level of abstraction increases, though, researchers tend to rely on data-driven approaches. This means many solutions model representations of a particular concept through fitting a learning algorithm ℓ on a sample of annotated instances $\mathfrak{L} \subset \mathcal{U}_{\mathcal{V},A} \times \mathcal{U}_{\mathbb{F},A'}$. p thus originates from a learning process, i.e., $p(\cdot) = \ell(\cdot \mid \mathfrak{L})$. The

[3] https://en.wikipedia.org/wiki/Die_Schuldigkeit_des_ersten_Gebots

literature in the discipline explores a wide range of learning algorithms to this end, from Nearest Neighbours and Gaussian Mixture Models (e.g., Tzanetakis and Cook, 2002) to Support Vector Machines (e.g., Andén and Mallat, 2014) and Neural Networks (e.g., Costa et al., 2017), among others.

Conventionally, predictors receive as input representations derived from explicitly engineered (or "handcrafted") audio features, but it is becoming increasingly popular to rely on end-to-end deep learning architectures where e is implicit (Humphrey et al., 2013). Handcrafted features may come from a standard set used across domains (e.g., Bogdanov et al., 2013; McFee et al., 2015b; Peeters, 2004) or be tailored for the specific problem (e.g., Gómez, 2006). End-to-end architectures remove the need for selecting or designing feature representations, and appear to achieve comparable or superior performance to conventional approaches in some domains (e.g., Korzeniowski and Widmer, 2017; Pons et al., 2018; Sigtia and Dixon, 2014).

2.2 Evaluation Paradigms in Music Content Analysis Research

Conducting proper evaluation is essential to determine whether developed MCA systems successfully address their intended problems (Sturm, 2016b). Designing such systems involves a myriad of choices that researchers and developers face, and need to be properly accounted for during evaluation. Proper evaluation practices enable comparison of alternative approaches, such as different implementations of e and p, and against the state of the art, tracking progress in the discipline and avoiding dead ends (Sturm, 2016a). Establishing such practices is far from a trivial endeavour, with some authors considering it one of the grand challenges of the discipline (Serra et al., 2013). This section reviews evaluation approaches for MCA studies, first introducing those that explicitly embrace the subjective nature of music and later focusing on those that aim to objectively assess success.

2.2.1 Embracing Subjectivity: Human Inspection and Judgment

Since music is a fundamentally human construct, assessing whether an artificial system successfully addresses a music analysis problem is far from straightforward. Some evaluation strategies address this issue by acknowledging the subjective nature of music and

introducing humans in the loop as judges. Although some members of the community regard such strategies as "unscientific" (Downie, 2003b), others advocate recuperating human feedback within a broader evaluation framework (Schedl et al., 2013; Urbano et al., 2013).

The most basic subjective strategy is what Hernández-Orallo (2017) calls "white box" evaluation. This strategy basically relies on inspecting the components of a system to determine whether they suit the target problem. For instance, given a rule-based agent or the rules derived from a Decision Tree model, a panel of experts in the subject may judge whether such rules capture the nature of the concept under study. The inherent biases this strategy risks introducing, as well as the increasing complexity of the systems the community develops, makes white box evaluation seem weaker than behaviour-based "black box" analyses (Cohen, 1995). A recent push for interpretable models, however, suggests inspection may suitably complement other kinds of evaluation approaches (Doshi-Velez and Kim, 2017; Lipton, 2016; Mishra et al., 2017).

An alternative to inspecting systems is to ask humans to judge the outcomes of such systems. This so-called "perceptual evaluation" — since the evaluation relies on some judges' perception — is based on listening tests: given the outcomes of one or more systems on the same $\mathcal{R}_\Theta$, people are asked to rate or rank the quality of each after listening to the input the systems receive (Gupta et al., 2018; Jillings et al., 2016; Wierstorf et al., 2017). Analyses of this kind dominated the literature in the early years of the discipline, but their conclusions were soon deemed unreliable since the different judges involved made comparisons across studies impossible (Downie, 2003b). Some authors argue, however, that abandoning perceptual evaluation ignores user satisfaction, the ultimate goal of any deployed system, as success criterion (Schedl et al., 2013).

Aside from outcome quality, other factors may influence user satisfaction, such as the speed and ease of use of the final system's interface. Published studies, however, rarely assess whole systems under near-working conditions despite some authors' recommendations (Serra et al., 2013; Urbano et al., 2013). Holistic user-experience evaluation has been called the "Grand Challenge" in MIR evaluation, and implemented alongside the more conventional "objective" tasks in recent editions of the MIREX[4] evaluation exchange (Hu

[4] http://www.music-ir.org/mirex/wiki/MIREX_HOME

et al., 2017; Lee et al., 2015).

2.2.2 Seeking Objectivity: Cranfield Paradigm and Classification Experiments

The first editions of the ISMIR conference highlighted an urgent need to standardise evaluation procedures, with the aim to objectively compare systems (Downie, 2003b). The research community then organised three workshops especially devoted to discussing and agreeing on an evaluation paradigm suitable for the discipline, whose main insights were gathered by Downie (2003a). The experimental methodologies developed by Text-IR researchers (Jones, 1981; Tague-Sutcliffe, 1992), exemplified by the competitions held in the Text REtrieval Conference (TREC)[5] (Vorhees, 2007), served as a clear reference for the MIR community during the early years of consolidation of the discipline (Downie, 2004). What follows reviews how the MIR community has adapted the Cranfield paradigm, the most common evaluation framework in Text-IR, to develop empirical practices not unlike those adopted in Artificial Intelligence and related disciplines (Cohen, 1995; Hernández-Orallo, 2017). Such practices assess the suitability of solutions to MCA problems without human intervention beyond collecting and annotating some recordings.

Evaluation in Text-IR has long relied on what is known as the Cranfield paradigm (Cleverdon, 1991), where the role of humans shifts from judges to annotators. Instead of appraising the outcomes of the systems to arbitrary inputs, in the Cranfield paradigm someone selects a set of documents — an *evaluation collection* — associated with a problem — or *domain* — and annotates them on their expected relevance to specific topic categories — or *queries*; the annotations are the *ground truth* of the collection. Once an evaluation collection has been created, multiple studies can use them to assess different systems, which reduces both biases in their comparison and evaluation costs from, e.g., recruiting judges.

A classic Cranfield experiment assesses systems using a selection of queries, for which systems generate lists of documents from the collection that each deems relevant. The lists are then compared against the ground truth for their corresponding query, considering which documents appear in the lists and, occasionally, the order in which they appear

[5] `https://trec.nist.gov/`

as well. This procedure matches the *Retrieve* strategy that Sturm (2014b) identifies in some MCA studies.

Inspired by the TREC competitions, the MIR community established the Music Information Retrieval Evaluation EXchange (MIREX), whose tasks exemplify the commonly accepted evaluation paradigm in the discipline. The annual MIREX campaigns started in 2005 alongside the ISMIR conference, with a pre-MIREX evaluation campaign held one year earlier (Cano et al., 2006), and have since become a common forum for the assessment of MCA systems and methods. They are considered to offer an objective comparison between systems, and are often used to test novel evaluation procedures (Downie et al., 2010). The positive impact of MIREX in MIR research is undeniable (Cunningham et al., 2012). Some authors, however, have recently raised concerns about the sustainability of these campaigns in their current format (McFee et al., 2016), proposing alternatives in terms of the logistics involved.

The evaluation strategies that the MIREX tasks illustrate, despite their original TREC inspiration, arguably differ from the Cranfield paradigm. Instead of asking systems to retrieve a list of relevant documents, evaluation in much MCA research is based on systems attempting to predict the ground truth annotations of a given set of recordings. Sturm (2014b) calls this strategy *Classify*, which prevails in several conventional MCA problems, such as Music Genre Recognition (MGR) (Sturm, 2014d), and Music Emotion Recognition (MER) (Sturm, 2013b). In this strategy, success relates to the proportion of ground truth annotations that a system manages to reproduce.

McFee et al. (2016) summarise the evaluation procedure they consider standard in MIR research, which essentially corresponds with the *Classify* strategy:

1. A human annotator observes an input, r_i, generating a reference output, a_i;
2. An MCA system receives the same input r_i and estimates the expected output, $\hat{a}_i$;
3. a_i and $\hat{a}_i$ are compared;
4. Steps 1-3 are repeated over all input items from the collection;
5. Summary statistics (the overall performance metrics of the system) are computed.

In practice, the human annotation of all r_i occurs previous to any system's evaluation, sometimes obtained as a consensus between multiple individuals instead of a single one as the procedure above suggests. Urbano et al. (2013) consider that such a procedure

implicitly reflects the query-answer nature of Cranfield-style Text-IR experiments, with queries matching classes and the answers being the system's predictions. In other words, they regard *Retrieve* and *Classify* as two sides of the same coin.

The widespread adoption of the *Classify* strategy in MCA studies motivates considering "classification experiments" (or, maybe more generally, "prediction experiments") as the *de facto* evaluation apparatus in the discipline, so the discussion is hereinafter focused on this strategy. Sec. 2.3 reviews in more detail the elements that form the pipeline of a music classification experiment, and Sec. 2.4 describes an example from the literature upon which the illustrative examples in later chapters are built. Music classification experiments implicitly assume that the capacity to reproduce the annotations of a curated collection of recordings entails an underlying ability to capture and model the intricacies of a musical concept. At face value, this interpretation seems sound; Sec. 2.5 argues this is not necessarily the case.

2.3 The Music Classification Experiment Pipeline

Studies that rely on a *Classify* evaluation strategy follow a similar pipeline to assess and compare systems and the methods that constructed them. This section first overviews the pipeline of a generic music classification experiment, which Fig. 2.2 represents schematically, and later details its main components. The concepts and notation introduced here appear throughout the rest of the dissertation.

2.3.1 Overview

Classification experiments compare systems on their ability to predict the annotations of a *collection*[6] $\boldsymbol{C} = \big(c_1, \ldots, c_N \;:\; c_n = (r_n, a_n)\big)$, where $r_n \in \mathcal{R}_\Theta$ is a *recording* or *realisation* (the raw data) and $a_n \in \mathcal{U}_{\mathcal{V},A}$ its *annotation* (the associated class label). Unless stated otherwise, it is assumed here that an experiment deals with a single collection. The link between recordings and labels in $\boldsymbol{C}$ intends to exemplify some concept of interest over the abstract population of music instances Θ. Sec. 2.3.2 discusses common characteristics of the collections used in MCA evaluation and some challenges that arise in their creation.

[6] This dissertation uses *collection* instead of the more common term *dataset* when referring to raw data instances. For consistency with the Machine Learning literature, the latter is used only in Sec. 3.3 to refer to tabulated data structures of extracted features ready to be used as inputs for learning algorithms.

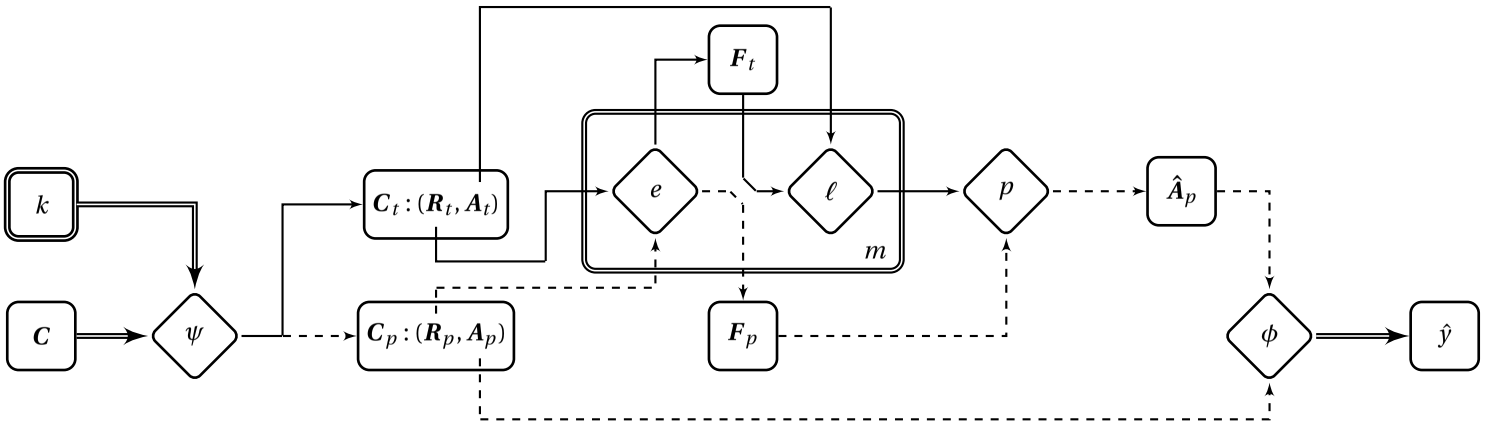


Figure 2.2: Pipeline of a single iteration k of a classification experiment evaluating a system construction method m (combination of feature extraction e and learning algorithm ℓ) in a collection $\boldsymbol{C}$. Square-shaped nodes represent data structures; diamond-shape nodes represent processes. A double border indicates a factor with fixed level. Solid lines indicate training flow; dashed lines indicate prediction flow. ψ is a data assignment/partitioning function. $\boldsymbol{C}_t$ is the training collection, with $\boldsymbol{R}_t$ its raw data and $\boldsymbol{A}_t$ the corresponding annotations; $\boldsymbol{C}_p$ is the testing collection, with $\boldsymbol{R}_p$ its raw data and $\boldsymbol{A}_p$ the corresponding annotations. $\boldsymbol{F}_t$ and $\boldsymbol{F}_p$ are the features extracted from the training and testing collections. p is the trained predictor, $\hat{\boldsymbol{A}}_p$ the predicted annotations, ϕ the performance metric function, and $\hat{y}$ the performance estimate.

The systems to be evaluated are assumed to rely on training a supervised learning algorithm ℓ to build a classifier (or regressor) p on a series of annotated instances from $\boldsymbol{C}$. A method m is the composition of a feature extractor e and a learning algorithm ℓ; a system s is a fixed actualisation of m through training. Most published MCA studies assess system construction methods instead of final systems, with many focusing on feature representations for particular description problems (Schedl et al., 2014). In MIREX, the organisation has recently grouped together several tasks that follow a similar pipeline and focus on the system construction methods as "Audio Classification (Train/Test)" tasks,[7] even though they deal with entirely distinct underlying concepts.

When learning algorithms are involved, it is commonplace to ensure that systems do not predict the annotations of recordings that were used in their training; otherwise, one risks *overfitting* — systems learning and exploiting the particularities of specific instances, instead of general patterns across the collection (Hastie et al., 2009). To avoid overlaps between recordings used for training and prediction, classification experiments rely on some *partitioning* (or *assignment*) function ψ. Section 2.3.3 describes common partitioning strategies, including some that rely on *resampling* — iteratively constructing multiple splits $(\boldsymbol{C}_t, \boldsymbol{C}_p)_k$ from $\boldsymbol{C}$. A classification experiment on a collection $\boldsymbol{C}$ thus comprises multiple iterations, each defined by a combination of a method m and an assignment variable k. The training data $\boldsymbol{C}_t$ for each iteration is then used to construct s with m, i.e., $s = m(\boldsymbol{C}_{t,k})$, and obtain predictions on $\boldsymbol{C}_{p,k}$, i.e., $\hat{\boldsymbol{A}}_p = s(\boldsymbol{C}_{p,k})$.

It is standard practice to use a further split of the training collection $\boldsymbol{C}_t$ for optimising hyperparameters of the learning algorithms. This process essentially follows the same pipeline in Fig. 2.2 but at a smaller scale. For simplicity of exposition, however, it is assumed that each optimisation round is a classification experiment in itself, with the method m in each iteration representing a particular combination of hyperparameters for a specific algorithm.

Given the predictions obtained in each iteration of a classification experiment, one estimates the suitability of each system or method through computing one or more *performance metrics* ϕ. These are almost always exclusively based on how closely predictions $\hat{\boldsymbol{A}}_p$ match the ground truth annotations $\boldsymbol{A}_p$, but other success criteria, such as compu-

[7] `http://www.music-ir.org/mirex/wiki/2017:Audio_Classification_(Train/Test)_Tasks`

tation speed, could also influence the conclusions. Sec. 2.3.4 describes the most widely used performance metrics in classification experiments. These metrics may suggest some solutions outperform others. Statistical tests may then help distinguish between real and spurious differences in performance. Some strategies for conducting these tests are reviewed later in Ch. 3 once the underlying statistical concepts have been introduced.

2.3.2 Evaluation Collections

The conventional evaluation paradigm in MCA studies requires annotated data collections to train and obtain predictions from the systems and methods to assess. Constructing such collections, however, is far from trivial, and arguably to a larger extent than other disciplines that adhere to a similar paradigm (Downie, 2003b). Challenges such as the availability of suitable data and the often complex annotation process strongly impact the quality and reproducibility of the research conducted in the discipline.

Data gathering Music data is costly to acquire and often restricted to share. Any modern computer with Internet connection suffices to obtain an overwhelming number of freely available text or image documents, to which anyone with a word processor or digital camera can easily contribute; gathering music data is much more complex. Creating music pieces is an extremely specialised skill that requires years of training and devoted equipment, and existing data often requires some payment to obtain and is subject to copyright regulations that severely limit the researchers' ability to share them with the community. As a consequence, many studies rely on data gathered from the researchers' personal collections, which do not necessarily represent the breadth of the intended scope of their analysis. This process is called *convenience sampling* (Urbano et al., 2013), as opposed to the random sampling that proper representation of the target population would require. Moreover, such collections are often kept private and poorly described after the publication of the study, which hampers reproducibility and progress in the discipline (Peeters and Port, 2012). Sturm (2012a), for instance, finds that in MGR almost 60% of published studies use private collections, with 75% of those studies relying solely on such collections.

Despite the difficulties in gathering and sharing music data, some evaluation collections are made public. This often leads to the specific task defined by such a collection

implicitly replacing the broader problem it intended to represent as research target for part of the community (Schedl et al., 2014). Although standardised collections facilitate comparisons across studies, their repeated use leads to what could be called "collection overfitting": solutions becoming increasingly more tailored to the specifics of the collection as opposed to the underlying problem (Drummond and Japkowicz, 2010). As discussed in Sec. 2.4.1, this is the case in MGR, where *GTZAN* (Tzanetakis and Cook, 2002) dominates among the publicly available collections despite its flaws. *GTZAN* includes audio recordings, but defies copyright regulations by providing only short excerpts of each song; others rely on data with explicit sharing permission, such as the Jamendo collection often used for the evaluation of Singing Voice Detection systems (Ramona et al., 2008).

Two main alternatives exist when sharing audio data is not feasible: providing precomputed feature representations or synthesising artificial audio. The collections that adhere to the first option tend to be much larger than their audio counterparts. This is the case, for instance, of the Million Song Dataset (Bertin-Mahieux et al., 2011) or the recent AcousticBrainz genre collection used within the MediaEval competitions (Bogdanov et al., 2018), which contains data from over two million recordings. As a comparison, *GTZAN* includes 1000 excerpts. Lacking audio, however, seriously limits which kinds of studies can benefit from those collections, since one can only evaluate learning algorithms in this manner. Synthetic data, despite being common in some problems such as frequency content estimation (Klapuri, 2009), is unsuitable for many problems of interest. In addition, even in those cases where synthetic data is feasible, its use for evaluation might lead to results that do not correspond with those one would get from real data (Niedermayer et al., 2011). Sturm and Collins (2014), however, argue that creating data that strictly follows a series of explicit rules might serve to evaluate algorithms in their ability to uncover such rules, avoiding at the same time the usual challenges in data gathering and annotation.

Annotation Annotating music collections is a complex and time-consuming process, the quality of which strongly impacts the development and evaluation of MCA systems (Schedl et al., 2014; Urbano et al., 2013). Research teams often rely on their own expertise to annotate private collections, which may introduce biases in the annotations that translate into the results they obtain. If such collections later become public, as in the case of

GTZAN, the original biases quickly spread across the community. When a collection is intended to be publicly released, however, it is common practice to gather annotations from multiple sources, such as a panel of experts, regular listeners, or a combination of both.

Various reasons may cause annotators to disagree in their perception (Flexer and Grill, 2016), such as differences in expertise, their listening environment, failure to phrase the requested task unambiguously, or simply the subjective nature of music itself. Such disagreements could be accounted for in the annotations as multiple weighted labels, but are often resolved through a majority vote (Craft et al., 2007). Other practical issues, such as the number of possible labels, also impact the quality of the annotations.

The research community has explored various strategies to facilitate and reduce the costs of annotation beyond recruiting students and colleagues from their departments. Some of those strategies aim at increasing the engagement of the annotators. Online platforms that reward users for their effort, such as Amazon's Mechanical Turk, offer an inexpensive and efficient way to collect a large number of annotations that appear largely consistent with other more controlled approaches (Lee, 2010). Games with a purpose have also proven useful to engage the crowds to obtain annotations, with several of such games being presented simultaneously at ISMIR'07 (Law et al., 2007; Mandel and Ellis, 2007; Turnbull et al., 2007). Some have also developed interactive tools to collect annotations from live audiences (Page et al., 2015).

Instead of attempting to maximise the number of annotations on a budget, McFee et al. (2016) propose the exact opposite: reduce the cost by annotating as little as possible. Their approach, which they call *incremental annotation*, assumes that only disagreements between systems serve to compare their performances. They thus suggest limiting the initial annotation to a subset of instances in the collection that will then be used to train the systems, and choosing another subset for testing from which annotators will be requested to annotate those that the systems predict differently. Apart from yielding estimates of performance differences between systems, these newly annotated instances can then be incorporated into the training collection for potential future evaluations. This procedure seems particularly fitted for regular competitions such as MIREX, but can only rank systems and not produce estimates of performance. It is also unclear whether it may introduce undesirable biases in the systems by extending the training material from a par-

ticular subset of instances that may have been distinctly predicted for a reason (e.g., their higher "difficulty").

2.3.3 Partitioning Strategies

Machine Learning-based systems are prone to overfit. It is thus standard practice to avoid assessing their performance using instances that were previously used in their training through partitioning collections into separate training and testing materials. This reduces the possibility of obtaining overoptimistic performance estimates. The most common partitioning strategies are the hold-out set, K-fold Cross-Validation, and bootstrap sampling, which are described briefly below. The reader is referred to textbooks such as those by Hastie et al. (2009), Alpaydin (2014) and Weihs et al. (2017) for more thorough reviews.

Any partitioning strategy may be modified to ensure *stratification*, so that the derived collections maintain the same class distribution as the original one. In practice, this may be accomplished by partitioning the instances of each class separately using the chosen strategy and joining them at the end.

Hold-out The most basic partitioning strategy is to create what is usually known as a hold-out set, which involves selecting a certain portion of the instances in the collection and avoiding using those for training. The selection is often random, but curation other than stratification might be necessary in some cases. No strict rule exists for the relative sizes of derived collections, although studies often leave aside around 20-30% of the instances for testing.

In this strategy, the partitioning is performed only once, which can be problematic. One obtains a single performance estimate, so there is no way to distinguish between the performance of the trained system and that of the method that was used to construct it, which is usually the real target of the evaluation. Even if the ultimate goal is to estimate the performance of a fixed system, this strategy does not utilise the whole collection to train such system, so the measured performance may substantially differ from what a system would have obtained had it been constructed using all instances in the collection. Moreover, a single measurement cannot suffice to determine if observed differences in perfor-

mance between systems are due to actual superiority or an artifact of the split. To address these issues, researchers often resort to resampling methods such as the ones below.

K-fold Cross-Validation Given a collection $\boldsymbol{C}$ of size N, K-fold Cross-Validation (K-CV) creates $K \leq N$ pairs $(\boldsymbol{C}_t, \boldsymbol{C}_p)$ using the following procedure: generate K "folds" randomly from $\boldsymbol{C}$, each containing N/K instances; then, iterate K times, each time selecting a different fold K_i and leaving it aside for testing, i.e., $\boldsymbol{C}_{p,i}$; combine the remaining $K-1$ folds for that iteration to create the corresponding training collection $\boldsymbol{C}_{t,i}$; train a system on each $\boldsymbol{C}_{t,i}$ and obtain the corresponding performance estimate on $\boldsymbol{C}_{p,i}$. This procedure thus yields K performance estimates per method to evaluate. These estimates are not independent, however, since their respective training collections overlap (for $K > 2$).

The preferred value of K is somewhat arbitrary, but $K = 10$ appears often. The MIREX classification tasks, however, use $K = 3$. Flach (2012) recommends adjusting K so that the number of instances in each fold is at least 30. An extreme alternative is to set $K = N$, which is called Leave-One-Out Cross-Validation. In this case, the procedure yields N performance estimates, each either a success or failure on a single instance of the collection. This alternative, however, is costly and does not permit stratification. To avoid this issue, some studies increase the number of measurements instead, conducting K-CV multiple times and averaging the results, with K being 10 or lower, as performed by Tzanetakis and Cook (2002). Dietterich (1998) recommends computing 5 rounds of 2-CV but, to the best of our knowledge, this option has not been implemented in MCA studies.

Bootstrap The statistical learning literature often encourages the use of bootstrap sampling over K-CV to obtain multiple train/test pairs (Hastie et al., 2009; Hothorn et al., 2005). Bootstrap sampling is based on the bootstrap estimation technique by Efron (1977). According to Alpaydin (2014), it constitutes the best resampling strategy for small collections. Nevertheless, the bootstrap has been virtually ignored by the MIR community to this end with very few exceptions, such as Skowronek et al. (2007). In a bootstrap sampling procedure, one draws uniformly N training instances with replacement from a collection $\boldsymbol{C}$ of size N. A training collection $\boldsymbol{C}_{t,i}$ derived in this way comes from the empirical distribution function of $\boldsymbol{C}$, which makes the elements of $\boldsymbol{C}_{t,i}$ independent and identically

distributed. Note $\boldsymbol{C}_t$ can contain repeated elements.[8] In fact, one expects $\boldsymbol{C}_{t,i}$ to contain around 63.2% of the instances in $\boldsymbol{C}$, so different $\boldsymbol{C}_{t,i}$ from the same $\boldsymbol{C}$ are likely to overlap. To obtain the corresponding testing collection $\boldsymbol{C}_{p,i}$, one could also use sampling with replacement, but it is preferable to derive it as $\boldsymbol{C}_{p,i} = \boldsymbol{C} \setminus \boldsymbol{C}_{t,i}$ (i.e., the part of $\boldsymbol{C}$ not included in $\boldsymbol{C}_{t,i}$) to ensure no overlaps occur between paired training and testing collections. Repeating this process an arbitrary number of times yields estimates with arguably improved statistical properties over K-CV, such as a reduced variance (Efron, 1983; Efron and Tibshirani, 1997).

2.3.4 Performance Metrics

Although there exist many different ways of estimating performance from the predictions in a classification experiment, only those that commonly appear in multiple MCA problems are reported here. The reader is referred to the textbook by Japkowicz and Shah (2011) for a thorough review of performance metrics in the context of the evaluation of learning algorithms; in addition, Raffel et al. (2014) describe several problem-specific metrics in MCA.

The most common performance metrics are derived directly from a confusion matrix, which records in each cell (x, y) the number of instances in the test collection with ground truth A_x that have been predicted as A_y, with both $A_x, A_y \in \mathcal{V}$, the vocabulary of classes in the collection. Let a_i be the ground truth annotation of instance i in a test collection $\boldsymbol{C}_p$, $\hat{a}_i$ its predicted annotation by a system, and $N_p = |\boldsymbol{C}_p|$ the size of the test collection. Then the *accuracy* of the system in $\boldsymbol{C}_p$ is:

$$acc = \frac{\sum_{i=1}^{N_p} I(a_i = \hat{a}_i)}{N_p} \tag{2.2}$$

where $I(\cdot)$ is an indicator function that yields 1 only if its predicate input is true, and 0 otherwise. This metric thus captures the proportion of correct predictions over the whole testing collection, and corresponds to the size of the diagonal in the confusion matrix relative to the sum of all entries. The complementary value to the accuracy is the error rate: $err = 1 - acc$.

[8] Since "set" implies no repeated elements, the use of the alternative term "collection" when referring to the training and testing materials is thus preferred here to encompass those created from bootstrap sampling.

Accuracy provides an estimate of the probability that an arbitrary instance from the collection will be predicted correctly. This, however, does not take into account the class distribution in the collection, which may strongly affect how representative the measurements are as estimates of performance. For instance, if a collection is highly imbalanced (the instances of a single class substantially outnumber the rest), accuracy can reach values close to 1 if systems predict all instances as belonging to the majority class. This issue motivates IR-related disciplines to prefer *precision* and *recall* over accuracy as performance metrics. These two metrics are defined at class level. In particular, for an arbitrary class $A_x \in \mathcal{V}$:

$$prec(A_x) = \frac{\sum_{i=1}^{N_p} I(a_i = \hat{a}_i = A_x)}{\sum_{i=1}^{N_p} I(\hat{a}_i = A_x)} \tag{2.3}$$

$$rec(A_x) = \frac{\sum_{i=1}^{N_p} I(a_i = \hat{a}_i = A_x)}{\sum_{i=1}^{N_p} I(a_i = A_x)} \tag{2.4}$$

Precision thus reflects the proportion of instances predicted to belong to a class that actually belong to that class, whereas recall reflects the proportion of instances of a class that are predicted to belong to that class. In terms of the confusion matrix, precision is basically a marginalisation over columns and recall over rows.

The so-called *F-score* or *F-measure* summarises the precision and recall of a class using a harmonic mean to penalise values close to zero in either of the two metrics:

$$F(A_x) = 2 \cdot \frac{prec \cdot rec}{prec + rec} \tag{2.5}$$

One can then obtain a summary of any of these metrics by averaging over all classes in the collection. The average recall, sometimes used as a sort of *normalised accuracy*, would then simply be:

$$rec = \frac{\sum_{\forall A_x \in \mathcal{V}} rec(A_x)}{|\mathcal{V}|} \tag{2.6}$$

with $|\mathcal{V}|$ being the number of classes in the collection. This value coincides with the accuracy in the particular case of a completely balanced collection.

2.4 Evaluation of Scattering-based Music Genre Recognition Systems

Having discussed the main characteristics of conventional MCA evaluation practices, this section now explores how a particular study from the literature implements such practices and the conclusions that its authors extract. The examined study was conducted by Andén and Mallat (2014) and assesses the benefits of a signal processing technique called the Scattering Transform in various audio classification problems. The focus lies here on the experiments they perform on the MGR benchmark collection *GTZAN*, which is described first.

2.4.1 Music Genre Recognition Evaluation using *GTZAN*

A widely studied problem in MCA is the automatic identification of the musical genre of a piece solely from the acoustic information in an audio recording. Music Genre Recognition (MGR), as it is often called, is a complex (and sometimes troubling) problem, since the concept of genre is largely a human construct and thus extremely challenging to define in objective terms. This has not stopped researchers proposing and publishing a plethora of approaches for MGR, the vast majority of which relying on Machine Learning techniques.

A particular music collection, known as *GTZAN*, appears in over a hundred publications (Sturm, 2014d), and remains a benchmark collection in recent studies (e.g., Choi et al., 2017). The article by Tzanetakis and Cook (2002) that introduced *GTZAN* is often considered as the seminal work of MGR; Sturm (2012b), however, finds various papers on the topic that precede it. The success of *GTZAN* is more likely due to it being the first MGR collection publicly available for download.[9]

GTZAN consists of 1,000 30-second 22050 Hz mono 16-bit audio excerpts in `au` format, each annotated with one of 10 music genre labels: `blues`, `classical`, `country`, `disco`, `hiphop`, `jazz`, `metal`, `pop`, `reggae`, and `rock`. Sturm (2014d) provides a thorough analysis of the contents of *GTZAN*, reporting repetitions, distortions and mislabellings, and highlighting the replication of artists in many classes. The collection was originally released without providing any information about its excerpts, and only recently the community has started identifying to which music recordings each belongs (Sturm, 2013c).

[9] `http://opihi.cs.uvic.ca/sound/genres.tar.gz`

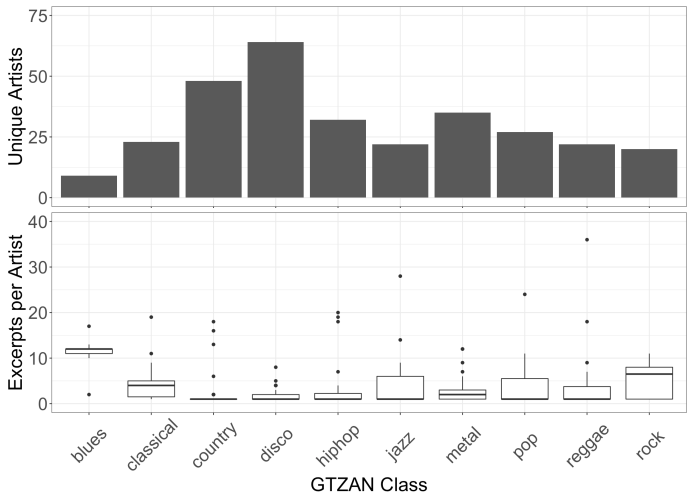


Figure 2.3: Artist distribution across classes in *GTZAN*, showing the number of unique artists (Top) and the quantiles of the number of excerpts per artist (Bottom) in each class. Dots indicate outliers.

At the moment of writing, all but 23 of the 1000 recordings in *GTZAN* have been identified.[10] Figure 2.3 summarises the artist distribution for each class in *GTZAN*, assuming all artists from still unidentified excerpts are unique. Queen is the only artist known to appear across classes in the collection (`rock` and `metal`). `blues` remains the class with highest artist replication, with all but one artist appearing in more than 10 excerpts. In `reggae`, a single artist (Bob Marley) appears in more than a third of the excerpts.

MGR performance obtained on *GTZAN* has substantially increased since the collection was first introduced. Tzanetakis and Cook (2002) reported accuracies just over 60%; less than a decade later, Guaus (2009) reached almost 100%. Sturm (2013c), however, suggests that accounting for the faults in the collection leads to an "ideal" classification accuracy no higher than 94.5%. Systems evaluated on *GTZAN* that achieve accuracies higher than this threshold might actually perform "worse" than some that appear inferior. Recent studies, such as the one Choi et al. (2017) conducted, use this "ideal" threshold as their target.

2.4.2 Systems based on the Scattering Transform

The Scattering Transform is a signal processing technique able to obtain feature representations of signals from diverse origins, such as image (Bruna and Mallat, 2013) or speech

[10] `http://www.eecs.qmul.ac.uk/~sturm/research/GTZANindex.txt`

(Andén et al., 2015). This technique generates audio features from a signal by means of a cascade of wavelet transforms, in a structure that resembles that of a Convolutional Neural Network (CNN), but generating output in every layer instead of only in the deepest one. In each layer, it applies a complex modulus to the wavelet transform of the input received from the previous one. As Mallat (2012) demonstrates, this modulus captures invariances to some perturbations, such as global time-shifts and local deformations like time-warping, if applied on the time axis. Unlike other hierarchical representations, such as CNNs, these invariances do not need to be learned from data but are inherent to the feature representations. This comes at the cost of losing some high frequency information in each layer, similarly to what happens in the computation of Mel-Frequency Cepstral Coefficients (MFCCs); unlike MFCCs, however, adding further layers to the cascade can recover the lost information.

Andén and Mallat (2011, 2014) proposed leveraging representations derived from the Scattering Transform to construct MGR systems. They claim such representations have perceptual relevance. Scattering-based representations relate to modulation features (Chi et al., 2005), which are potentially useful for timbre-related music classification problems, such as instrument recognition (Siedenburg et al., 2016), or MGR (Lee et al., 2009), since genres are often characterised by a particular ensemble of musical instruments. For that purpose, they train Support Vector Machines (SVM) on *GTZAN* with different variants of the features, mainly by changing the number of layers and whether such layers apply over time or frequency dimensions.

In terms of the formalism introduced above, the MCA systems in Andén and Mallat (2014) are as follows. $\mathcal{R}_\Theta$ consists of time-domain signals of duration about 30 seconds uniformly sampled at $F_s = 22050$ Hz (the sampling rate of *GTZAN*). $\mathcal{U}_{\mathcal{V},A}$ is the set of the 10 *GTZAN* labels. $\mathcal{U}_{\mathbb{F},A'}$ is a space consisting of sequences of 80 elements of a vector vocabulary $\mathbb{F}$. All systems trained by Andén and Mallat (2014) use the same $\mathcal{U}_{\mathcal{V},A}$ and Gaussian-kernel SVM as learning algorithm ℓ, but extractor e with different $\mathcal{U}_{\mathbb{F},A'}$. The excerpts are split into 80 half-overlapping time frames, and the trained systems predict annotations for each such a frame independently; the final prediction for each excerpt is obtained from majority voting over the individual frame-wise predictions.

Extractor	Short Description	Accuracy
`Mel Sc.`	Δ-MFCC (T=740 ms)	82.0 ± 4.2
`1-L Sc.`	Time Scattering, $l = 1$	80.9 ± 4.5
`1&2-L Sc.`	Time Scattering, $l = 2$	89.3 ± 3.1
`TF Sc.`	Time & Frequency Scattering, $l = 2$	90.7 ± 2.4
`TF Adap. Sc.`	Time & Frequency Scattering, $l = 2$, Adaptive Q_1	91.4 ± 2.2
`1,2&3-L Sc.`	Time Scattering, $l = 3$	89.4 ± 2.5

Table 2.1: Classification accuracies (in % ± standard deviation) on *GTZAN* reported for the scattering-based MGR systems by Andén and Mallat (2014)

Andén and Mallat (2014) describe the six feature extractors that they compare as follows, with the term "order" indicating the number of layers used in the Scattering Transform cascade:

`Mel Sc.:` Mel-frequency spectrogram (84 coefficients, 740-ms frames, 50% overlap), concatenated with first- and second-order time derivatives over the sequence of feature vectors, for a total of 252 feature dimensions.[11]

`1-L Sc.:` First-order ($l = 1$) time-scattering features (effective sampling rate 2.7 Hz), for a total of 85 feature dimensions.

`1&2-L Sc.:` First- and second-order ($l = 2$) time-scattering features (effective sampling rate 2.7 Hz), for a total of 747 feature dimensions.

`TF Sc.:` First- and second-order ($l = 2$) time-frequency scattering features, for a total of 1574 feature dimensions.

`TF Adap. Sc.:` First- and second-order ($l = 2$) time-frequency-adaptive scattering features, for a total of 1907 feature dimensions.

`1&2-L Sc.:` First-, second-, and third-order ($l = 3$) time-scattering features (effective sampling rate 2.7 Hz), for a total of 2769 feature dimensions.

To compare the different system-construction methods they consider, the authors perform 10-fold Cross-Validation (10-CV) on *GTZAN*. The article does not specify whether the 10-CV is stratified, so we assume each of the 10 folds contains a unique but likely imbalanced selection of recordings from the collection. The authors report as performance metric the average error rates over the 10 folds plus/minus standard deviation, which correspond to the classification accuracies shown in Tab. 2.1. Although they do not compute

[11] In reality, the analysis conducted for the study reported in Ch. 4 reveals that the implementation by Andén and Mallat (2014) does not actually compute Δ- and Δ-Δ-MFCCs, but instead cyclically time-shifts the sequence of MFCCs ahead and behind by one frame, so that the predictor has flexibility in learning a transformation.

any statistical test, the authors conclude that systems generating time-frequency-adaptive scattering representations (i.e., `TF Adap. Sc.`) outperform the others they implement. Moreover, they also claim that such systems achieve higher performance than the ones proposed by Lee et al. (2009), which they consider the state-of-the-art at the moment of publication with an accuracy of $90.6\% \pm 3.1$. According to what Sturm (2013c) finds in his systematic review, however, this claim seems questionable.

2.5 Critical Analysis of Conventional Evaluation Practices

The standardisation of evaluation practices that the MIR community undertook in the early 2000s arguably improved over the previously inconsistent situation. The approach most adopted, however, is not flawless. Researchers in the community realised that the evaluation paradigm exemplified by the MIREX campaigns required some improvements soon after it became widely accepted. This section summarises the most relevant criticisms and ways forward proposed in the literature. First, however, the concepts of experimental validity and reliability are briefly reviewed, since they are fundamental for judging any empirical practice but have largely been ignored by the MIR community.

2.5.1 Experimental Validity and Reliability

To the best of our knowledge, only Urbano et al. (2013) have previously attempted to translate the concepts of experimental validity and reliability to an MIR setting. Others, such as Gouyon et al. (2014), later provide formal definitions of validity for some specific problems, but take the concept translations Urbano et al. (2013) make at face value. Some such translations, however, are arguably not entirely accurate from an MCA perspective, so what follows revisits this topic and provides alternative views where necessary.

2.5.1.1 Validity vs Reliability

Validity and reliability are both desirable properties of any empirical methodology, and MCA evaluation is no exception. Roughly speaking, validity concerns how close to the actual truth one can get from the results of an experiment, whereas reliability concerns how close to each other multiple measurements are when conducted under similar conditions. Inspired by Trochim and Donnelly (2007), Urbano et al. (2013) use a bullseye as

a metaphor to clarify the distinction between validity and reliability. Suppose the centre of the bullseye represents an underlying truth, and shooting arrows represents experiments aimed at uncovering such truth. An experimental methodology leading to valid conclusions would then appear as arrows hitting the target surrounding the centre of the bullseye, on average, regardless of how scattered they end; a methodology leading to reliable conclusions, on the other hand, would appear as arrows hitting the target clustered around the same spot, regardless of how far such a spot is from the centre. In statistical terms, validity relates to bias and reliability to variance.

Ideally, one expects empirical methodologies to shoot the bullseye right in the centre — i.e., to produce both valid and reliable conclusions. Unfortunately, this is rarely the case. Methodologies that yield both invalid and unreliable conclusions are obviously harmful to any discipline. However, validity and reliability are often considered to require a trade-off, since addressing one might harm the other, so one might wonder what to prioritise when assessing and improving evaluation practices. The bullseye serves well for this. Shooting arrows in a tight cluster far away from the centre of the bullseye seems to require a straightforward adjustment: either move the shooter or the bullseye so that the arrows hit right in the centre. One rarely knows where the actual truth is, such as how successful a solution is in modelling a musical concept or even how well a particular system performs in deployment conditions. Many studies thus implicitly place the bullseye where the arrows hit, changing the question that the experiments answer after the fact to something along the lines of "how many ground truth annotations does a system reproduce." "Shooting more arrows" often overcomes the dangers of a lack of reliability, since a conclusion becomes more trustworthy as more evidence packs around it; unfortunately, no number of arrows can fix invalid conclusions. Compared to other disciplines, the artificial nature of the experiments in MCA makes them relatively cheap and adjustable. Evaluation practices should arguably leverage these upsides to help locate the intended goal instead of blindly shooting over and over towards a misplaced bullseye. Attempting to make measurements more reliable without regarding their validity, or faking validity by adjusting the question and not the mechanism to answer it, harms the progress of the discipline.

Another distinction that needs to be addressed is that between reliability and reproducibility. In most academic disciplines, such a distinction does not exist: a result is reliable if it is reproducible. Those disciplines mostly deal with experiments in the physical world for which some conditions are beyond the control of the experimenter. Reproducibility is thus defined as "closeness of the agreement between the results of measurements of the same measurand carried out under *changed* conditions of measurement" (Taylor and Kuyatt, 1994) (emphasis not in the original). In other words, reproducibility in that sense associates with similar, but different, measurement conditions, as does the definition of reliability above. In computer experiments, however, results tend to be deterministic — the same conditions that yielded a specific result can be replicated to achieve that same result. For MIR and associated disciplines, therefore, reproducibility involves whether a third party has all resources available (and whether they are easy to use) to generate an *exact* copy of the results of an experiment (Peng, 2011; Six et al., 2018). This is sometimes referred to as repeatability (Bartlett and Frost, 2008), even in some MIR publications such as by Page et al. (2013), or replicability (Drummond, 2009). Both senses are undeniably important, but this dissertation assumes that reliability and reproducibility mean different concepts.

2.5.1.2 Validity Threats and Campbellian Typology

Validity threats involve any circumstance that might lead to a false conclusion being accepted by producing an apparent but false effect, or obscuring a real one (Trochim and Donnelly, 2007). These threats might work at different levels and require different tools to reduce their potential impact, so it is useful to identify such levels and their particular vulnerabilities. The most common way of categorising validity is usually called the Campbellian validity typology, first introduced by Campbell (1957) more than half a century ago and presented in its current form by Shadish et al. (2002). It consists of four types of validity: (statistical) conclusion, internal, construct, and external validity. These categories are organised hierarchically, in the sense that no inference might be regarded as internally valid if threats to statistical conclusion validity have not been properly tackled, and so on. Despite some criticism and a few attempts to propose alternatives, such as by Reichardt

(2011), the distinction between these four categories remains the default approach, especially in the social sciences.

Urbano (2011), and later Urbano et al. (2013), provide a thorough discussion of the Campbellian validity typology applied to MIR research evaluation, but almost entirely from a IR perspective. In their opinion, the truth to uncover is how satisfied a user would be with a particular system, instead of whether such a system successfully captures a musical concept. Each of the four categories is here revisited, including a Machine Learning (ML) perspective missing in previous analyses. Sec. 2.5.2 links some pitfalls of conventional evaluation practices identified in the MCA literature with such categories, emphasising the underlying musical concept targeted instead of a hypothetical user.

(Statistical) Conclusion Validity The base of the validity staircase concerns whether one can infer a covariation between independent and dependent variables in a study (Shadish et al., 2002). One often intends to detect whether a differential effect exists on the response when measurements are performed under different conditions. Assuming the independent variable is the system (or the method to build a system) and the response variable is a performance metric over a testing collection, in a classification experiment a researcher might infer that a difference in the performance metric exists between systems. This conclusion is statistically valid if sufficient statistical rigour has been employed to reach it.

Typical threats to statistical conclusion validity involve the improper, or lack of, use of statistical machinery suitable for the type of data one gathers from the measurements. Lack of statistical power when the sample is too small also threatens this type of validity. ML and IR researchers have long encouraged fellows to perform statistical tests over the results of their studies before publishing (Jones, 1981; Langley, 1988; Tague-Sutcliffe, 1992), similar to what some advocate in MIR (Flexer, 2006). Nevertheless, while inference has become standard practice in the former disciplines (Japkowicz and Shah, 2011), even to an excessive degree in some authors' opinion (Drummond, 2006), the same cannot be said of MIR research beyond evaluation forums such as MIREX (Downie et al., 2010; Urbano et al., 2013).

Internal Validity An inference is internally valid if, *within the study*, an observed covariation arises due to a causal relationship between the independent and dependent vari-

ables (Shadish et al., 2002). In other words, one wants to ensure that observed differential effects in the response appear because conditions have been purposely changed, and not due to uncontrolled circumstances. Lack of control facilitates *confounding*, which leads to invalid conclusions about causal relationships (Pearl, 2009). Two variables potentially influencing measurements are confounded if the experimental design cannot disentangle their effects (Cobb, 1998). Many experimental and quasi-experimental designs alleviate confounding by controlling extraneous variables other than the target of the study — explicitly setting or accounting for their values in the different experimental conditions — to avoid them impacting the measurements (Montgomery, 2013; Shadish et al., 2002).

Simple experimental design choices overcome the most obvious risks of confounding in classification experiments (Langley, 1988). For instance, if one measures the performance of multiple systems each on different instances, the influence of such systems (the outcome of interest) becomes confounded with the selection of instances (an extraneous variable). This is easily avoided by comparing measurements on the same instances, a standard evaluation practice.

Construct Validity In a study, a *construct* is the ultimate concept about which one wants to make conclusions, and an *operationalisation* is how one targets such a construct within the study, i.e., what is measured. Construct validity thus concerns whether the conclusions one makes about the operationalisations in a study generalise to the intended constructs (Shadish et al., 2002). In other words, construct validity concerns whether a study measures what one intends to measure. In the context of classification experiments and related practices, this often involves whether the metrics employed reflect the true performance of the evaluated algorithms (Hand, 2012; Jamain and Hand, 2008; Law, 2008), or whether such metrics capture user satisfaction (Carterette, 2011).

Some argue most forms of confounding arise due to breaches of construct validity (Coolican, 2017): a variable becomes a confounder because of how a construct and its operationalisation relate, or differ. This seems to be the sense some authors in ML-related disciplines employ (e.g., Charalambous and Bharath, 2016). Understood in this manner, confounding in classification experiments may occur when the explicit success criteria — e.g., achieving a high value on a performance metric — can happen for reasons other

than meeting the implicit success criteria — e.g., capturing the defining characteristics of an underlying concept. Researchers in fields such as Computer Vision have realised the risks that this kind of confounding provokes, leading to systems that appear successful on benchmark collections but fail when exposed to minor perturbations in their input (Nguyen et al., 2015). Since this interpretation of confounding is fundamental for the purposes of this dissertation, subsequent chapters develop it further.

External Validity The upmost level of the validity ladder concerns how generalisable the conclusions of a study are (Trochim and Donnelly, 2007). A conclusion is generalisable if it applies to a broader population of interest than the specific settings of the study intend to represent, or even beyond to further populations and settings (Shadish et al., 2002). Breaches of external validity thus involve conclusions that fail to generalise as claimed, or experimental settings that do not provide enough support to such claims.

Literature related with classification experiments defines generalisation restrictively as performance estimates remaining stable when calculated on instances unseen during model training (Hastie et al., 2009). In this sense, overfitting becomes the major threat to external validity. This interpretation, however, assumes that the collection defines the problem or, in the best case scenario, that all its instances are uniformly drawn from a specific data generation process of interest. Although some industry environments might face situations where evaluation and deployment data come from the same source, this is arguably not the case in most published research. If one aims to evaluate a learning algorithm in a domain-agnostic manner[12] or a method on its ability to capture the characteristics of an underlying concept, then generalisation must concern data from populations yet to be considered. The term generalisation is used hereinafter in this sense. The selection bias that convenience sampling often introduces thus becomes a major threat to external validity in classification experiments, along with any breach in statistical rigour, control and construct operationalisation that might compromise the lower validity levels.

[12] Domain-agnostic superiority is theoretically impossible according to the popular No Free Lunch theorems (Wolpert and Macready, 1997). In practice, one often aspires to superiority for some realistic data distributions within a restricted family of domains (Japkowicz and Shah, 2011).

2.5.2 Pitfalls of Conventional Evaluation Identified in the Literature

Over the years, researchers have raised concerns about certain aspects of the conventional evaluation practices in MCA, showcasing potential validity threats that the community faces. Many of those concerns appear within the context of music similarity analysis, which we consider an umbrella term for many MCA problems that deal with, or at least are used as proxies for, modelling resemblances between recordings according to some explicit or implicit criteria. Studies often address these problems from a classification perspective, assuming that if a system predicts that two recordings belong to the same category, then such system considers those recordings closer than those that end in different categories. Regardless of whether a study addresses a problem such as MGR as a goal in itself or as a proxy for music similarity, the evaluation machinery remains unchanged, thus facing similar validity threats.

The main threat to statistical conclusion validity, as stated above, is neglecting or misusing statistical machinery when analysing measurements. Flexer (2006) claimed most published MIR research at the time of writing missed statistical inferential analysis; several years later, Urbano et al. (2013) still found that was the case. The size of the music collections used for evaluation also poses a threat to statistical conclusion validity, since it relates to the strength of the inferences one can derive from data. Urbano and Schedl (2013) discuss this issue and propose a method to reach sufficient inferential power through collection size at minimal cost. Whether the blind use of common inference tools, such as frequentist statistical tests, leads to valid conclusions of practical importance is however questionable (Urbano et al., 2012), especially in a scientific climate that seems increasingly wary of some of those tools (Ioannidis, 2005; Wasserstein and Lazar, 2016).

As in other ML-related disciplines, threats to construct validity often relate to discrepancies between performance metrics and the implicit success criteria they intend to represent. When multiple different metrics exist, it may be unclear which one, if any, best operationalises success (e.g., Serrà, 2007). Even if a metric dominates the evaluation of a particular problem, multiple implementations can exist, which may also hamper the relationship between operationalisation and construct (Raffel et al., 2014). Metrics rarely reflect human perception of success (Davies and Böck, 2014; Hu and Kando, 2012; Seyerlehner et al., 2010), thus some authors suggest incorporating user-specific information

into the measurements (Hu and Liu, 2010; Schedl et al., 2013). Similarly, Widmer (2016) argues that systems achieving high performance in conventional metrics often demonstrate a complete lack of basic musical knowledge, such as the inherent temporal nature of music. Metrics thus fail to operationalise the degree of musical knowledge systems acquire.

Operationalisations other than performance metrics also compromise construct validity in classification experiments. These include, for instance, which $\mathcal{R}_\Theta$ and $\mathcal{U}_{\mathcal{V},A}$ one selects to reflect the Θ of interest and its intended descriptions. Craft et al. (2007) discuss unreliability in the annotations of music collections, an issue that others later echo (Flexer and Grill, 2016; Pálmason et al., 2017; Wiering, 2009). Wiggins (2009) goes one step further and claims that the concept of ground truth in music is misleading, since music is a product of culture and the minds of those who interact with it, and thus intrinsically dynamic. Following the suggestion by Aucouturier and Pachet (2003) that MGR is ill-defined, McKay and Fujinaga (2006) discuss the suitability of MGR tasks as a testbed for MCA systems, encouraging researchers to rethink how they state, address and evaluate the problem. According to Sturm (2014d), little has changed recently: MGR is still the most widely addressed problem in the literature, using virtually the same evaluation strategy and collection as when it became popular almost two decades ago.

The faults of *GTZAN* mentioned in Sec. 2.4.1 highlight a further issue in the conventional evaluation approach: both systems and the collections used in their evaluation are often treated as pure black boxes. Until Sturm (2013c) independently created a partial index of its content, at least a hundred studies had already used *GTZAN* in their evaluation completely disregarding which recordings appeared. Sturm (2013a) shows that analysing the usually overlooked causes of success (or failure) provides information useful for improving the proposed systems, an impossible endeavour if one ignores the contents of the collection. Relying solely on the number of ground truth labels a system reproduces, he argues, no matter how sophisticated the reported performance metrics are, does not guarantee the validity of the experimental results. He conducts similar analyses for mood classification problems, reaching the same conclusions (Sturm, 2013b, 2014c).

The partitioning of collections into training and testing materials affects validity in classification experiments, as the MIR community has long acknowledged. For instance,

the presence of the same artists or albums in both training and testing recordings artificially inflates performance estimates; this is known as artist or album effects, respectively (Flexer and Schnitzer, 2010; Pampalk et al., 2005). In recent years, Sturm has repeatedly identified systems that appear successful on benchmark collections but exploit information extrinsic to the problem at hand (Sturm, 2014a, 2016b; Sturm et al., 2015). He refers to such systems as "horses" as a homage to Clever Hans — a horse that appeared able to solve mathematical problems but was instead relying on unintentional gestural cues by its questioners (Pfungst et al., 1911). The Clever Hans metaphor is gaining increasing attention in disciplines beyond MIR (e.g., Hernández-Orallo, 2019; Lapuschkin et al., 2019), with authors such as Hand (2018) acknowledging the trust issues that "horses" cause in data-driven research.

A major consequence of the issues above is that experimental results fail to generalise to deployment scenarios. The usual convenience sampling procedure employed in both constructing collections and selecting them for evaluation further reinforces this common failure. Bogdanov et al. (2016), for instance, shows that systems trained on recordings from one collection often substantially underperform when applied on different, but related, collections. Moreover, the discrepancy between evaluation and deployment settings, along with the black box nature of conventional evaluation, contributes to the fact that disciplines that could benefit from insights obtained from MCA research largely ignore its methods and outcomes (Aucouturier and Bigand, 2013; Siedenburg et al., 2016).

2.5.3 Improvements to Evaluation Practices

Researchers in the MIR community have proposed several modifications to the conventional evaluation practices to address the issues summarised above. Some such proposals have already been mentioned above, but those most relevant for the goals of this dissertation are reviewed in more detail here. Despite their merit, user-centric proposals are left aside, since they also require a solid underlying framework that current approaches fail to provide. Proposals focused on implementation and logistic details (e.g., McFee et al., 2016; Raffel et al., 2014) are also not covered.

Sturm (2016a) identifies and discusses what he considers the most commonly proposed solutions, which largely correspond to improvements to each of the main compo-

nents of the classification experiment pipeline reviewed in Sec. 2.3. Apart from the need for transparent tool implementations that Raffel et al. (2014) suggest, he mentions increasing the size of evaluation collections, generalising the use of resampling strategies such as cross-validation, tailoring performance metrics and ensuring studies conduct formal statistical testing. He argues these improvements are less pressing than ensuring experiments answer the questions one actually intends to ask to avoid making errors "of the third kind" — getting the correct answers to the wrong questions (Hand, 1994). In other words, no matter how much data and how many folds you employ, and how sophisticated are the performance metrics and tests you compute, the validity of any conclusion you make will always be compromised unless the experimental procedure itself is properly revised. Sturm advocates that the community should prioritise the principles of statistical experimental design (Fisher, 1935; Montgomery, 2013), which he believes the literature in the discipline largely ignores.

Sturm (2016a) mention filtered partitioning, irrelevant transformations, and interpretable explanations as specific evaluation practices aimed at ensuring that experimental results actually address the intended research question of a study. These are reviewed next, as is Item Response Theory, a promising evaluation approach that has been recently explored in other ML-related disciplines but not in MIR research at the time of writing.

Filtered Partitioning Pampalk et al. (2005) introduced artist "filters" to counteract artist effects in music similarity experiments — performance estimates becoming higher when recordings of the same artist appear during both training and prediction than when they do not. Their approach is referred to as "filtered partitioning" hereinafter, since it may be applied to information other than the artist, such as the album to which each recording belongs (Flexer and Schnitzer, 2010). The principle is simple: instead of randomly assigning recordings to either the training or testing collections, one groups in one such collection all those that share a particular piece of information that may cause problems (e.g., their artist). Similar to stratification, this process introduces a regulation in the partitioning process, in this case ensuring that systems cannot exploit a particular source of information to predict annotations on the "regulated" testing collection.

Comparing regulated results from filtered partitioning with those from a conventional

random partitioning enables assessing the impact of leaving a factor unregulated. Using this approach, some studies show not only that unregulated collections might bias performance estimates, but also that the magnitude of such bias varies across feature representations and learning algorithms (Flexer, 2007; Sturm, 2014d). More commonly, studies use filtered partitioning to alleviate overoptimistic performance estimates, such as in MIREX. This increases the chances that the experimental results actually address the question of interest — whether systems are able to capture the intricacies of the target concept (e.g., genre) instead of exploiting auxiliary information (e.g., artist characteristics).

Irrelevant Transformations Data augmentation techniques artificially increase the amount of data available from a collection, either by transforming its original instances — data warping — or creating new instances within the feature space — synthetic oversampling (Wong et al., 2016). Researchers from the MIR community have largely focused on augmenting training data in a variety of problems, such as MGR (Li and Chan, 2011), chord recognition (Humphrey and Bello, 2012) and singing voice detection (Schlüter and Grill, 2015). Data augmentation of this kind intends to generate more generalisable systems by providing a wider variety of inputs than the real data permits, and software tools such as the MUDA architecture developed by McFee et al. (2015a) facilitate this process.

Adversarial training (Goodfellow et al., 2015; Gu and Rigazio, 2014) is a particular type of data augmentation where the incorporated instances, called "adversarial examples", are obtained through perturbations identified by maximising prediction error — i.e., minimal modifications of the input data that produce the largest increase in the system's error rate. Incorporating adversarial examples in their training makes systems more robust to small perturbations that should not affect their predictions. Recent publications suggest that this approach is gaining traction within the MIR community. Stoller et al. (2018), for instance, apply adversarial training to singing voice extraction, whereas Kim and Bello (2019) employ it to address music transcription.

Augmenting test data enables evaluating the effects of data perturbations. Adversarial attacks employ adversarial examples created as mentioned above to determine the vulnerability of trained systems to small perturbations in their input (Goodfellow et al., 2015; Kereliuk et al., 2015; Szegedy et al., 2014). Within the MIR community, the Audio Degrada-

tion Toolbox by Mauch and Ewert (2013) provides several transformations mainly aimed at assessing how robust prediction systems are to suboptimal acoustic conditions. The evaluation paradigm first introduced by Sturm (2014a) goes a step further, employing "irrelevant transformations" on the testing data to uncover reasons behind performance.

The procedure that Sturm (2014a) proposes involves transforming input signals in a manner strongly linked to a specific cue that systems could exploit, and observing whether performance changes in relation to such a transformation. If the transformation is supposedly irrelevant for the problem of interest, either because of the nature of the underlying concept or because human listeners retain their judgement of the annotations despite the transformation, but performance estimates change as a consequence, systems are likely exploiting the information being manipulated

A possible implementation of this idea is a *deflation* process, which attempts to force a system to behave as if it was randomly assigning labels by iteratively modifying the recordings that it previously labels correctly. Given a system s and a collection $\boldsymbol{C}$, a deflation process involves the following steps:

1. Find the recordings in $\boldsymbol{C}$ that s maps "correctly";
2. Create a transformation $t(\cdot)$;
3. Apply $t(\cdot)$ to all recordings found in step (1);
4. Have s map transformed recordings;
5. Find the transformed recordings that s maps "incorrectly";
6. For each recording in (1) that s now maps "incorrectly" in (5), replace it in $\boldsymbol{C}$ with its transformed version;
7. Return to (1); repeat until the performance estimate of s reaches a random baseline, or a maximum number of iterations is reached.

Sturm (2014a) also proposes the reverse, an *inflation* process that attempts to reach perfect performance estimates by means of irrelevant transformations.

Combining irrelevant transformations with system analysis has revealed the reasons behind the success of several systems on a variety of problems. Sturm (2016b), for instance, shows that the systems proposed by Pikrakis (2013), which are apparently successful in identifying rhythm patterns from audio, actually rely on the tempo of the recordings (their "speed") to predict correctly the annotations of the *BALLROOM* collection (Dixon

et al., 2004). The relationship between tempo and annotations appears mainly because dance competitions heavily regulate how fast or slow each dance style can be.

Interpretable Explanations Most Machine Learning algorithms yield trained models with such an inherent complexity that is virtually impossible to discern how they make predictions, leading to studies often relying on speculation to justify apparent success (Lipton and Steinhardt, 2018). Realising stakeholders in some fields, such as finance or healthcare, often mistrust predictions made by black boxes, researchers in ML-related disciplines have recently increased efforts to make models and their decisions more easily "interpretable" (Doshi-Velez and Kim, 2017). These efforts have led to scientific meetings devoted to the topic along with prestigious venues such as the ICML conference.[13]

Some researchers have proposed techniques to facilitate the interpretability of MCA systems, such as for the auralisation (Choi et al., 2016) and visualisation (Schlüter, 2016) of features that contribute to the predictions systems make. Mishra et al. (2017) propose *SLIME*, a method to interpret the behaviour of MCA systems based on the post-hoc model-agnostic local analysis developed by Ribeiro et al. (2016). Given a classifier built using any learning algorithm, this technique aims to highlight which parts of a spectrotemporal representation of an input signal contribute most towards a decision. As an example, Mishra et al. (2017) analyse singing voice detection systems (Lehner et al., 2013), revealing that some highly performing ones appear to use parts of the input that contain no voices to correctly predict recordings as "vocal". More recently, Mishra et al. (2018a,b) improve the contiguity and efficiency of the explanations using feature inversion, a technique to derive plausible representations in the form of the input from the intermediate representations that the layers of deep neural networks generate. Also based on Ribeiro et al. (2016), Haunschmid et al. (2019) propose a method to explain the predictions of Music Emotion Recognition systems. These works showcase an increasing interest in understanding what MCA systems actually learn from data.

Item Response Theory In psychometrics, researchers attempt to devise better ways to measure psychological traits and aptitudes. Hernández-Orallo and collaborators have recently noticed the similarity between this goal and that of the evaluation of Machine

[13] https://sites.google.com/view/whi2018

Learning and, more generally, Artificial Intelligence systems (Hernández-Orallo, 2017; Martínez-Plumed et al., 2019). In particular, they propose leveraging the so-called Item Response Theory (IRT) (Baker and Kim, 2017; de Ayala, 2009). This, in essence, entails decomposing each individual measurement — i.e., the response to a test question — into "latent" variables that capture characteristics of both the subject (or *respondent*), such as their *ability*, and the question (or *item*) itself, such as its *difficulty* or *discrimination* capability. IRT thus aims to uncover reasons behind performance, gauging the extent to which observed results arise from the actual capabilities of the evaluated entity and how much they are influenced by the contents of the test itself. This information can then help make ability judgements relieved from test artefacts.

In a Machine Learning context, Martínez-Plumed et al. (2016, 2019) illustrate the use of an evaluation methodology inspired by IRT to compare classifiers. Classifiers correspond to respondents in the IRT paradigm, with individual instances in test collections as items on which one measures — i.e., the responses to be decomposed are single class predictions. Martínez-Plumed et al. (2016, 2019) showcase how one can interpret the estimates of item-level parameters they consider (difficulty, discrimination, and guessing — the probability of chance-like success) to gain insights lacking in conventional benchmarking, and explore which classifiers should be included in the analysis to obtain suitable estimates. Lalor et al. (2016) instead propose using human raters to estimate IRT item-level parameters, which can then be used to adjust performance measurements of Natural Language Processing systems. More recently, Lalor et al. (2019) suggest replacing humans with "artificial crowds" simulated via Deep Neural Networks. Regardless of how parameter estimates are obtained, these works highlight how decomposing performance measurements into latent contributions can provide insights on the behaviour of assessed systems as well as uncover issues with the evaluation collections employed.

2.6 Summary and Forward Look

Research in Music Content Analysis (MCA) aims at developing systems that automatically describe audio recordings according to some musically-related concept. The most widely used evaluation strategy to assess the suitability of MCA systems is the classification experiment, which relies on the predictions such systems make on some previously anno-

tated collection of recordings. This evaluation approach is subject to a variety of validity threats and limitations. A major concern is the inability to discern whether a system actually addresses the problem for which it was designed or suffers from confounding instead. Proposed improvements to the classification experiment pipeline, such as introducing filtered partitioning or irrelevant transformations, illuminate the reasons behind apparent success (or failure) and thus provide information necessary to improve developed systems. Later chapters build upon these ideas to propose a systematic evaluation methodology that jointly assesses MCA systems and some possible confounding effects that might impact their performance. Together with techniques aimed at providing interpretable explanations to the predictions MCA systems make, the proposed methodology suggests a falsificationist perspective to MCA evaluation: instead of attempting to confirm a system's success, which might be impossible, one should try as thoroughly as possible to disprove it; each failed attempt then provides further evidence of success. First, however, Ch. 3 introduces the fundamental principles and tools of statistical Design of Experiments (DoE) on which the proposed methodology relies.

CHAPTER 3

STATISTICAL DESIGN AND ANALYSIS OF EXPERIMENTS

Any discipline that aspires to be regarded as scientific must pursue the highest standards of evidence in its studies. Design of Experiments (DoE) is a branch of statistics that aims at precisely this. From the seminal work of Fisher (1935), DoE researchers develop methodologies to plan experiments in a way that ensures appropriate data collection and analysis, so that they may provide valid evidence to answer the research questions one targets.

This chapter first introduces the fundamental principles and tools of DoE in Sec. 3.1, including widely accepted methods to express and analyse experimental measurements. Sec. 3.2 then reviews a particular approach to DoE based on what is known as the Calculus of Factors. Sec. 3.3 finally reviews common mechanisms to express and analyse the results of classification experiments.

The explanations in this chapter largely follow the traditional terminology and conventions of frequentist statistics used in much DoE research. The core ideas presented, however, would easily adapt to other approaches gaining traction in the community, such as Bayesian inference (Benavoli et al., 2017), as a response to the limitations of Null-Hypothesis Significance Testing (Berrar and Dubitzky, 2018).

3.1 Experimental Design Fundamentals

This section introduces the fundamental concepts of DoE, as well as the expression of empirical measurements as structural models, upon which the following sections build. The many excellent textbooks from the DoE literature, such as those by Montgomery (2013), Cobb (1998) and Mason et al. (2003), discuss more extensively these matters; Hinkelmann (2015) overviews the main contributions to DoE from a historical perspective.

3.1.1 Terminology

In a DoE context, an *experiment* is an empirical study that deliberately changes one or more independent variables to observe how the change affects dependent variables; the deliberate change is an *intervention*. The set of conditions to compare are the *treatments* of the study, and the outcome variable to observe is the *response*. Each *run* (iteration) of the experiment yields a single measurement of the response. An *effect* is a difference in response between treatments.

In a broad sense, the subjects of the study are called *units* or, for historical reasons, *plots*. In many studies, each subject receives an individual treatment and is measured once; however, this is not always the case. Consider, for instance, students who attend the same lectures, but are examined individually — treatments (the lectures) and measurements (the exams) target subjects at different levels of aggregation. Subjects that receive treatments are *experimental units*, whereas those on which one performs measurements are *observational units*.

Both treatments and units in a study might display *structure*: characteristics common across multiple elements of those sets. Treatments may be combinations of basic components, such as chemical and dosage in drug testing; units may form homogeneous groups, such as sex or assigned physician in human patients. A characteristic that groups units is a *blocking* variable, with each of the resulting groups being a *block*.

One jointly calls treatment and blocking variables the explanatory variables, or *covariates*, of the experiment. This dissertation assumes that covariates are categorical — a categorical, or *factor*, variable takes one of a limited number of values, called *levels*. Each observation i in a study takes a particular level $\mathcal{F}(i)$ of each covariate factor $\mathcal{F}$.

3.1.2 Principles of Experimental Design

An experimental *design* is a particular plan to assign treatments to units, so that one can isolate the effects of interest from nuisance variability introduced by auxiliary factors. Following the tradition of Fisher (1935), most experimental designs rely on the principles of *replication*, *randomisation*, and *blocking*. Some also include *factorisation* as a fundamental principle of experimental design (e.g., Cobb, 1998). According to Mason et al. (2003), adopting these principles mainly intends to eliminate known sources of bias and reduce the impact of unknown ones on the inferences derived from experimental results.

Replication For each individual treatment or combination of treatment factors, one should obtain multiple observations to capture and account for variability other than effect differences. This is directly related to the power calculations in statistical inference tests. Replication should not be confused with repetition, however; a measurement is *repeated* if it is obtained more than once over the same observational unit. Due to natural variations and possible measurement errors, one might encounter variation in responses measured repeatedly on the same unit; these, however are *false replications*, since the effect of the treatment and that of the particular unit cannot be disentangled. In this dissertation, replicated measurements or *replicates* always refer to measurements on different observational units with the same treatment.

Randomisation Decisions such as the assignment of treatments to units or the ordering of measurements should be made randomly when possible. Random does not mean haphazard; true randomness requires the aid of specific devices, since one can never guarantee that haphazard decisions are devoid of unconscious biases. Randomising thus promotes the planned chance-like variability most statistical tests assume, as opposed to unknown systematic variability that might bias results. In computer-based experiments, however, randomisation appears less relevant. Digital environments can perfectly duplicate all units, and subject each duplicate to a different treatment without risking side effects such as spillovers — the application of a treatment affecting measurements other than the intended one. Ordering effects are also virtually non-existent.

Blocking When one identifies closely similar units according to some variable, it is preferable to group them into *blocks* and apply randomisation within each block separately. This accounts for the systematic variability that each group introduces, thus removing it from the effects of interest. Blocking hence *controls* the impact of a variable on the measurements. Deciding which possible sources of homogeneity deserve being controlled, however, is far from trivial. For any given experiment, widely different conclusions could be reached depending on which blocks are considered for its design and analysis.

Factorisation If a study involves multiple treatment variables (i.e., multiple sets of conditions, each of a different nature), one should consider, if feasible, all possible combinations of the values such variables take as individual treatments to assign and compare in a single experiment. The rationale behind this is twofold. First, factorisation makes studies more efficient, since designing and conducting a single experiment targeting multiple treatments simultaneously is likely cheaper and faster than multiple separate experiments. Second, including all possible combinations of values might illuminate *interactions* between variables; two variables interact if the value one takes impacts the effect the other has on the response.

These principles inform a large portion of the experimental designs that studies implement. Of them, replication is probably the most important; no general conclusion can arise from a single observation. Randomisation is beneficial in most settings, but largely irrelevant in those of most interest here. Finally, both blocking and factorisation are optional but highly recommended when feasible. Structural models, which are discussed next, reflect the choices one makes in this regard.

3.1.3 Structural Models

Estimating effects of interest requires relating the responses to the explanatory variables in some form. *Structural models* pose particular presumed relationships (Horton, 1978). Authors often assume that the observed responses can be approximated as a linear combination of the effects of relevant covariates. This provides a simple and well-established paradigm for analysing experimental results.

The structural models considered here are *mixed-effects* models, which means they include parameters for both *fixed* and *random* effects. A fixed effect assumes all units with the same level of the related factor contribute equally to the response; a random effect allows the contribution to differ across units, coming from a random variable with parameters that depend on the level of the factor. DoE usually associates fixed effects with treatment factors, and random effects with blocking factors. This assumes the treatments in the study include all one aims to compare, whereas the blocks only include a (random) selection among all possible groups.

A *linear* mixed-effects structural model decomposes the response y_i observed in each unit i as the sum of (a) a value μ constant for all units, which Cobb (1998) calls the *benchmark* parameter, (b) the fixed-effects parameters $\tau_{\mathcal{F}(i)}$, (c) the random-effects parameters $\beta_{\mathcal{G}(i)}$, and (d) a *residual* ε_i. Each factor contributes to the expected response adding (or subtracting) an amount that depends on its level. The residual captures the difference between expected and actual response for a particular unit.[1]

In the literature, the benchmark parameter μ might represent two different quantities: a *baseline* (e.g., Eugster, 2011) or a *global mean* (e.g., Bailey, 2008). The interpretation of the factor effects changes depending on which quantity μ represents, so researchers must choose carefully and report clearly which one applies in their study. Unless stated otherwise, this dissertation assumes μ in structural models represents a baseline. Estimated effects should then be interpreted as differences against the expected response of some reference combination of factor levels.

There are a few models based on some simple assumptions that appear often. When one ignores all structure in both units and treatments, and these are assigned to the units at random, one adopts a *Completely Randomised Design* (CRD). Its corresponding structural model only includes the fixed effect of a treatment factor $\mathcal{F}$:

$$y_i = \mu + \tau_{\mathcal{F}(i)} + \varepsilon_i. \tag{3.1}$$

For instance, a clinical study might aim to compare the effects of several drugs on the recovery of patients. In that case, the levels of $\mathcal{F}$ correspond to each such drug, with μ

[1] Traditional interpretation of ε_i usually associates this term with measurement error; in reality, it also reflects the effects of factors missing in the model and the natural variation among units. Some authors call this term *residual* instead of *error* to avoid misinterpretations (Montgomery, 2013).

possibly modelling the mean response measured in patients taking either a placebo or a "gold-standard" drug.

If the treatment factor $\mathcal{F}$ reflects combinations of levels of various additional factors, say $\mathcal{G}$ and $\mathcal{H}$, the measurements can be expressed as a *Factorial Design* (FD), with model:

$$y_i = \mu + \tau_{\mathcal{G}(i)} + \tau_{\mathcal{H}(i)} + \tau_{\mathcal{G}\mathcal{H}(i)} + \varepsilon_i \tag{3.2}$$

where $\mathcal{G}\mathcal{H}$ represents the *interaction* between $\mathcal{G}$ and $\mathcal{H}$. For instance, in the example above, $\mathcal{G}$ and $\mathcal{H}$ might represent chemical compound and dosage, respectively.

Both Eqn (3.1) and (3.2) only include parameters for fixed effects. The opposite, models that only include random effects, have little practical use since one usually assumes distributions with zero mean for such effects. Combinations of fixed and random effects, on the other hand, are pervasive. If units group into blocks $\mathcal{B}$, with at least one observation per level of the treatment $\mathcal{F}$ in each block, one has a *Complete Block Design* (CBD):

$$y_i = \mu + \tau_{\mathcal{F}(i)} + \beta_{\mathcal{B}(i)} + \varepsilon_i . \tag{3.3}$$

For instance, the levels of $\mathcal{B}$ might correspond to the hospitals the patients attend while participating in the clinical study.

Well-established tools, such as least squares, provide estimates of the effects considered in such models from observed measurements. These estimates facilitate conclusions about the particular effects one aims to evaluate, isolating them from possible nuisance variables if accounted for through a suitable structural model. Although linear additive models, such as those presented above, dominate much of the literature, more complex relationships may also be considered. In particular, the family of models that Nelder and Wedderburn (1972) called Generalised Linear Models (GLMs) extend ordinary linear additive models for non-normal responses. In GLMs, a *link function* transforms the response variable in a manner depending on its assumed distribution, and this transformed response is then related with a linear additive combination of explanatory variables. For the particular case of binomial data, GLMs that use the *logit* link function are often called *logistic models*, although other link functions, such as the so-called *probit* are also suitable. GLMs for responses with a Poisson distribution, on the other hand, are called *loglinear models* and use a logarithm as link function. GLMs in which parameters for both fixed

and random effects appear are called Generalised Linear Mixed-Effects Models (GLMMs). Unless stated otherwise, however, this dissertation follows the usual convention of using linear additive structural models for both theoretical explanations and concrete analyses.

3.1.4 Statistical Inference for the Hypothesis of Equality of Means

Researchers usually design and conduct experiments in order to determine whether various conditions yield different outcomes. In practice, this often boils down to comparing the mean responses from observations grouped according to their treatment factor level. In the presence of uncertainty, however, observed differences in such means may arise as a product of chance. Statistical inference techniques aim at gauging whether a given set of observations provide enough evidence for a particular claim. For instance, a common approach is to assume the observed responses come from populations with the same mean — i.e., define a null hypothesis H_0 of equality of means — and perform a test to determine whether the measurements sufficiently support the assumption.

Most tests rely on calculating a specific *test statistic* from the measurements and compare its value with a probability distribution of a particular family. To determine whether a difference is *statistically significant*, then, one first fixes a target probability α, called *significance level*. Many studies use $\alpha = 0.05$, but this value is largely arbitrary. One then obtains from the probability distribution of the test statistic the value that corresponds to the α set and compares it with the calculated test statistic. If the test statistic exceeds the threshold value from the distribution, one rejects the null hypothesis H_0 of equality of means.

Due to chance, however, it is still possible to reject the null hypothesis when it is actually true, or not reject it even if it is actually false. The former is called a *Type I error*, and the latter a *Type II error*. The *power* or sensitivity of a test is the probability of correctly rejecting H_0 when the alternative hypothesis H_1 is true. Increasing the number of replications in each group raises the power of the test.

The remainder of this section introduces tests conventionally conducted to assess the equality of means between groups. These tests are *parametric*, which means they rely on strong assumptions about the underlying population distribution. Some data types of interest may not satisfy those assumptions, so Sec. 3.3 includes some alternatives. In any

case, the following tests (and the frequentist inferential approach in general) are presented not necessarily to advocate their use, but because they help understand the decomposition of measurements into contributions.

Two groups: t-test For the particular case of the equality of means between two groups, it is common to use a t-test, which involves comparing a test statistic with a Student's t distribution. Under the null hypothesis $H_0 : \mu_1 - \mu_2 = 0$, the statistic for a t-test is:

$$t = \frac{\bar{y}_1 - \bar{y}_2}{S_{\bar{y}_1 - \bar{y}_2}} \tag{3.4}$$

where $\bar{y}_1$ and $\bar{y}_2$ are the sample means (average of all measurements belonging to each group), and $S_{\bar{y}_1 - \bar{y}_2}$ the sample standard error of the difference, obtained from:

$$S_{\bar{y}_1 - \bar{y}_2} = \sqrt{S^2\left(\frac{1}{N_1} + \frac{1}{N_2}\right)} \tag{3.5}$$

with N_1 and N_2 the number of observations in each group, and S^2 a pooled estimate of the unknown population variance σ^2:

$$S^2 = \frac{\sum (y_{i1} - \bar{y}_1)^2 + \sum (y_{i2} - \bar{y}_2)^2}{N_1 + N_2 - 1} = \frac{SS_1 + SS_2}{N_1 + N_2 - 1}. \tag{3.6}$$

SS refers to the *sum of squares* of a sample (or, more strictly of the deviations from the mean of a sample), which captures its variability. The Student's t distribution against which one compares the statistic has $N_1 + N_2 - 1$ degrees of freedom.

The procedure above, however, assumes that the population variances of the two groups match. If this assumption is violated, and especially when the group sizes differ, the test may yield invalid conclusions. To assess whether the data violates the assumption of homogeneity of variance, one can test the null hypothesis $H_0 : \sigma_1 = \sigma_2$ through the ratio between sample variances; equivalently, $H_0 : \sigma_1 / \sigma_2 = 1$. The sample variance for each of the groups is defined as the quotient between its sum of squares and the number of elements in that group, $S_j^2 = SS_j / (N_j - 1)$. The test statistic is thus defined as:

$$F = \frac{S_1^2}{S_2^2} \tag{3.7}$$

which follows a Fisher's F distribution with $N_1 - 1$ and $N_2 - 1$ degrees of freedom under the null hypothesis. If the F-statistic cannot reject the null hypothesis, then the formula of the sample standard error in Eqn (3.5) changes to:

$$S_{\bar{y}_1 - \bar{y}_2} = \sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}} \tag{3.8}$$

and the degrees of freedom of the t distribution to:

$$df = \frac{\left(\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}\right)}{\left(\frac{S_1^2}{N_1}\right)^2 / (N_1 - 1) + \left(\frac{S_2^2}{N_2}\right)^2 / (N_2 - 1)}. \tag{3.9}$$

Multiple groups: ANOVA For the general case of $g \geq 2$ groups, one could perform all pairwise comparisons as above, but would need to introduce some correction to compensate for the increased probability of error that this causes. Instead, the DoE literature favours the Analysis of Variance (ANOVA) to test differences between the mean responses of multiple groups. Multiple tests of the ANOVA family exist but, to facilitate the explanation of the underlying principles, only the simplest of them (the so-called One Way Unrelated ANOVA) will be considered here.

Put very simply, ANOVA compares the *variance between groups* with the *variance within groups*, since one expects observations in the same group to be more homogeneous among themselves than with the others if each group affects the response differently. If all observations within each group matched, then all the variation in the response would be attributable only to the independent variable that defines the groups. In reality, however, observations from the same group tend to vary due to natural differences between units and other unavoidable circumstances affecting the measurements; these differences are called *residual variation*.

Each observation deviates from the *overall mean* by some amount. Each individual deviation can be split into two quantities: the amount by which the measurement deviates from its group mean, and the amount by which the group mean differs from the overall mean. ANOVA relies on estimating the variance of each of these three components — *total* variance, *between groups* variance, and *within groups* (or *residual*) variance.

Given N observations y_{ij} from g groups, each of size N_j, with $\bar{y}$ the overall sample mean ($\bar{y} = \sum_{j=1}^{g}\sum_{i=1}^{N_j} y_{ij}/N$) and $\bar{y}_j$ the sample mean of group j ($\bar{y}_j = \sum_{i=1}^{N_j} y_{ij}/N_j$), then the *total sum of squares* (SS_{tot}), which captures the overall variability in the data, is:

$$\begin{aligned} SS_{tot} &= \sum_{j=1}^{g}\sum_{i=1}^{N_j}(y_{ij}-\bar{y})^2 = \sum_{j=1}^{g}\sum_{i=1}^{N_j}[(\bar{y}_j-\bar{y})+(y_{ij}-\bar{y}_j)]^2 \\ &= \sum_{j=1}^{g}\sum_{i=1}^{N_j}(\bar{y}_j-\bar{y})^2 + \sum_{j=1}^{g}\sum_{i=1}^{N_j}(y_{ij}-\bar{y}_j)^2 + 2\sum_{j=1}^{g}\sum_{i=1}^{N_j}(\bar{y}_j-\bar{y})(y_{ij}-\bar{y}_j) \\ &= \sum_{j=1}^{g}\sum_{i=1}^{N_j}(\bar{y}_j-\bar{y})^2 + \sum_{j=1}^{g}\sum_{i=1}^{N_j}(y_{ij}-\bar{y}_j)^2 + 2\sum_{j=1}^{g}(\bar{y}_j-\bar{y})\sum_{i=1}^{N_j}(y_{ij}-\bar{y}_j) \\ &= \sum_{j=1}^{g}\sum_{i=1}^{N_j}(\bar{y}_j-\bar{y})^2 + \sum_{j=1}^{g}\sum_{i=1}^{N_j}(y_{ij}-\bar{y}_j)^2 \end{aligned} \tag{3.10}$$

because the cross-product term is 0:

$$\sum_{i=1}^{N_j}(y_{ij}-\bar{y}_j) = \sum_{i=1}^{N_j} y_{ij} - N_j\bar{y}_j = \sum_{i=1}^{N_j} y_{ij} - N_j\sum_{i=1}^{N_j}\frac{y_{ij}}{N_j} = \sum_{i=1}^{N_j} y_{ij} - \sum_{i=1}^{N_j} y_{ij} = 0.$$

Therefore, SS_{tot} can be expressed as the sum of two components, SS_{bet} and SS_{res}. These correspond to the *between groups* and *within groups* variability mentioned above. The quotient of SS_{tot} and the *total degrees of freedom* of the data ($df_{tot} = N-1$) is the *total sample variance* $S^2_{tot} = SS_{tot}/df_{tot}$.

Since there are g groups, the variability between groups has $df_{bet} = g-1$ degrees of freedom. Dividing SS_{bet} by df_{bet} yields the *mean squares* differences between groups:

$$MS_{bet} = \frac{SS_{bet}}{df_{bet}} = \frac{\sum_{j=1}^{g}\sum_{i=1}^{N_i}(\bar{y}_j-\bar{y})^2}{g-1}. \tag{3.11}$$

The degrees of freedom of the residual are given by the difference between df_{tot} and df_{bet}: $df_{res} = (N-1) - (g-1) = (N-g)$. The *residual mean squares* are, thus:

$$MS_{res} = \frac{SS_{res}}{df_{res}} = \frac{\sum_{j=1}^{g}\sum_{i=1}^{N_j}(y_{ij}-\bar{y}_j)^2}{N-g}. \tag{3.12}$$

Both mean squares follow the structure of a sample variance. In fact, the expectation of MS_{res} is precisely σ^2 under the assumption that the residual variation is normally distributed with 0 mean and variance σ^2. Moreover, if no actual difference exists between groups, i.e., all observations come from a single population, the expectation of MS_{bet} is also σ^2. This suggests comparing mean squares to test the null hypothesis H_0 of no differ-

ence between group means. More precisely, the *variance ratio*:

$$F_0 = VR = \frac{MS_{res}}{MS_{bet}} \tag{3.13}$$

follows under H_0 an F distribution with $g-1$ and $N-g$ degrees of freedom. This value can be used as test statistic for H_0, similar to the ones presented in Eqn (3.4) and (3.7) but in this case testing for all comparisons simultaneously. Although rejecting H_0 suggests that at least one group differs from the rest, this method alone does not inform which.

The One Way Unrelated ANOVA approach presented implicitly presumes a CRD model such as the one in Eqn (3.1), assuming no structure relating units and a single treatment factor defining all groups. Generalisations of ANOVA to more complex structures exist, but standard statistics textbooks present each as a separate method. The next section describes a unified framework that permits generalising ANOVA-like analyses to structures of arbitrary complexity subject to certain conditions.

3.2 The Calculus of Factors Approach to Experimental Design

The Calculus of Factors is a mathematical approach for the analysis of experimental data assuming that both units and treatments can be expressed as factor variables. This approach leverages the theory of vector spaces and matrix operations, with which researchers in disciplines related with Signal Processing and Machine Learning may be familiar. Since the concepts related with this approach might appear obscure, Appendix A includes some concrete numerical examples. Bailey (2008, 2015) and Cheng (2014) provide more detailed introductions to this topic.

3.2.1 Factors and their Relationships

Let $\mathcal{F}$ be a factor variable associated with an element of an experiment, where such an element can be either a plot[2] or a treatment. $\mathcal{F}(x)$ refers to the level of factor $\mathcal{F}$ on element x. $\mathcal{F}[x]$ refers to the *class* (or *part*) of $\mathcal{F}$ containing element x, formed by the set of elements that share the same level of factor $\mathcal{F}$ with x; $\mathcal{F}[x]$ is called the $\mathcal{F}$-*class* of x.

[2] The term "plot" is a historical name for "observational unit", which is used here for brevity.

The number of parts in which a factor $\mathcal{F}$ splits the set of all plots Ω or treatments $\mathscr{T}$ is denoted $N_{\mathcal{F}}$. This matches the number of levels of $\mathcal{F}$ if and only if all those levels occur in the set. $\mathcal{F}$ is *uniform* if all $N_{\mathcal{F}}$ $\mathcal{F}$-classes include the same number of observations. All classes in this case have equal size $K_{\mathcal{F}}$, so $N_{\mathcal{F}} K_{\mathcal{F}} = N = |\Omega|$.

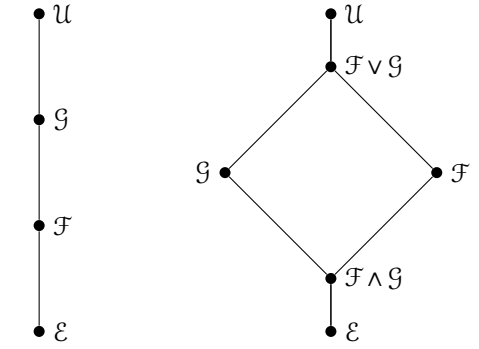
Let $\mathcal{G}$ be another factor variable. $\mathcal{F}$ and $\mathcal{G}$ are *equivalent*, or *aliased*, when *all* $\mathcal{F}$-classes are also $\mathcal{G}$-classes (and vice versa); writing $\mathcal{F} \equiv \mathcal{G}$ in this case. Two equivalent factors are essentially the same, regardless of whether they are labelled differently. Conversely, if $\mathcal{F}$ and $\mathcal{G}$ are not equivalent, all $\mathcal{F}$-classes might also be $\mathcal{G}$-classes, but not the other way around. In this case, $\mathcal{F}$ is *finer* than $\mathcal{G}$, which is denoted $\mathcal{F} \prec \mathcal{G}$, and $\mathcal{G}$ is *coarser* than $\mathcal{F}$, which is denoted $\mathcal{G} \succ \mathcal{F}$. If can be finer or equivalent to $\mathcal{G}$, one writes $\mathcal{F} \preccurlyeq \mathcal{G}$ or, likewise, $\mathcal{G} \succcurlyeq \mathcal{F}$.

The concepts of finer and coarser factors lead to a pair of fundamental factors. The *universal factor* $\mathcal{U}$ consists of a single class that includes every plot in the experiment; $\mathcal{U}$ thus contains the classes of all other factors, so $\mathcal{F} \preccurlyeq \mathcal{U}$ for any factor $\mathcal{F}$. Conversely, the *equality factor* $\mathcal{E}$ consists of one class for each plot in the experiment; $\mathcal{E}$ is thus contained in the classes of all other factors, so $\mathcal{E} \preccurlyeq \mathcal{F}$ for any factor $\mathcal{F}$. In general, then, for any factor $\mathcal{F}$ we have $\mathcal{E} \preccurlyeq \mathcal{F} \preccurlyeq \mathcal{U}$.

The *infimum* of two factors $\mathcal{F}$ and $\mathcal{G}$ is a factor $\mathcal{I} = \mathcal{F} \wedge \mathcal{G}$ whose classes are the (non-empty) intersections between $\mathcal{F}$-classes and $\mathcal{G}$-classes. Hence, $\mathcal{I} \preccurlyeq \mathcal{F}$ and $\mathcal{I} \preccurlyeq \mathcal{G}$, since all classes of $\mathcal{I}$ are contained both in $\mathcal{F}$ and $\mathcal{G}$, but not all classes of $\mathcal{F}$ and $\mathcal{G}$ need to be contained in $\mathcal{I}$. In addition, if another factor $\mathcal{H} \preccurlyeq \mathcal{F}$ and $\mathcal{H} \preccurlyeq \mathcal{G}$, then $\mathcal{H} \preccurlyeq \mathcal{I}$. The dual concept of *supremum* is a factor $\mathcal{S} = \mathcal{F} \vee \mathcal{G}$ such that $\mathcal{F} \preccurlyeq \mathcal{S}$ and $\mathcal{G} \preccurlyeq \mathcal{S}$, and $\mathcal{S} \preccurlyeq \mathcal{H}$ for every factor $\mathcal{H}$ that satisfies both $\mathcal{F} \preccurlyeq \mathcal{H}$ and $\mathcal{G} \preccurlyeq \mathcal{H}$. The classes of $\mathcal{S}$ are the smallest subsets of plots that join all units appearing in the same class either in $\mathcal{F}$ or $\mathcal{G}$. The infimum resembles an intersection between factors, and the supremum their union.

Graphical representations called *Hasse diagrams* can be used to express relationships between factors. To this end, each factor in a set corresponds to a dot in the diagram, whose connections capture their mutual coarseness or fineness, such as in Fig. 3.1. If $\mathcal{F} \prec \mathcal{G}$ then the dot for $\mathcal{G}$ is drawn above the dot for $\mathcal{F}$, joining both dots with a line. Since the universal factor $\mathcal{U}$ is coarser than any other factor, its dot is drawn at the top of the diagram; similarly, the dot for the equality factor $\mathcal{E}$ is drawn at the bottom. If neither $\mathcal{F}$ nor $\mathcal{G}$ are finer than the other, one includes a dot for their supremum ($\mathcal{F} \vee \mathcal{G}$) above and

connected to both of them. A dot for their infimum ($\mathcal{F} \wedge \mathcal{G}$) is also included below and connected to both of them (unless $\mathcal{F}$ and $\mathcal{G}$ are treatment factors and one knows that they do not interact).



(a) $\mathcal{F} \prec \mathcal{G}$ (b) $\mathcal{F} \not\prec \mathcal{G}$

Figure 3.1: Hasse diagrams showing possible relationships between factors.

3.2.2 Subspaces defined by Factors

Let $\mathbf{y}$ be an N-dimensional real-valued vector containing the observations from an experiment. This vector belongs to the $\mathbb{R}^{|\Omega|}$ vector space, which contains subspaces for each factor $\mathcal{F}$ defined in the experiment. $V_{\mathcal{F}}$ is the vector subspace of $\mathbb{R}^{|\Omega|}$ comprising vectors with values constant within each $\mathcal{F}$-class. The dimension of each such subspace equals the number of partitions $N_{\mathcal{F}}$ of the corresponding factor $\mathcal{F}$.

The two fundamental factors introduced in Sec. 3.2.1 also have associated vector subspaces. The subspace $V_{\mathcal{E}}$ corresponding to the equality factor $\mathcal{E}$ coincides with the entire vector space $\mathbb{R}^{|\Omega|}$, hence $V_{\mathcal{E}}$ can be constructed as the span of the set of standard vectors in $\mathbb{R}^N$. The subspace $V_{\mathcal{U}}$ of the universal factor $\mathcal{U}$, on the other hand, is a one-dimensional space of constant vectors — i.e., all vectors of size N with identical values in their coordinates. $V_{\mathcal{U}}$ is the span of $\mathbf{1}_N$ (a vector of N ones).

Relationships between factors correspond to relationships between their vector spaces. In particular, if $\mathcal{F} \preccurlyeq \mathcal{G}$ then $V_{\mathcal{G}}$ is a subspace of $V_{\mathcal{F}}$ — i.e., $V_{\mathcal{F}}$ contains $V_{\mathcal{G}}$ ($V_{\mathcal{G}} \subseteq V_{\mathcal{F}}$). Moreover, a vector is constant on each $\mathcal{F}$-class and $\mathcal{G}$-class if and only if it is constant on each class of their supremum $\mathcal{F} \vee \mathcal{G}$; the subspace $V_{\mathcal{F} \vee \mathcal{G}}$ thus equals $V_{\mathcal{F}} \cap V_{\mathcal{G}}$.

For any factor $\mathcal{F}$ in an experiment, two matrices of size $N \times N$ relate pairs of plots. The *relation* matrix $\mathbf{R}_\mathcal{F}$ contains ones only in cells that correspond to pairs of plots in the same $\mathcal{F}$-class; all other cells are zero. The *projection* (or *averaging*) matrix $\mathbf{P}_\mathcal{F}$ is defined in exactly the same way as $\mathbf{R}_\mathcal{F}$, but containing $1/|\mathcal{F}[\omega]|$ instead of ones in the non-zero cells. This matrix projects orthogonally any vector $\mathbf{v} \in \mathbb{R}^{|\Omega|}$ onto $V_\mathcal{F}$.[3] This property permits decomposing data into different sources, which forms the basis of some statistical analyses such as ANOVA. If $\mathcal{F}$ is uniform, converting between relation and projection matrices becomes trivial: $\mathbf{R}_\mathcal{F} = K_\mathcal{F}\mathbf{P}_\mathcal{F}$.

The projection matrices associated with the special factors $\mathcal{E}$ and $\mathcal{U}$ appear in the analysis of all experiments. Since $\mathcal{E}$ creates a class for every individual plot, and is thus uniform with $K_\mathcal{E} = 1$, its projection matrix is $\mathbf{P}_\mathcal{E} = \mathbf{I}_N$ (the identity matrix of size $N \times N$). $\mathcal{U}$, on the other hand, creates a single class that includes all plots, thus $K_\mathcal{U} = N$. Its projection matrix is then $\mathbf{P}_\mathcal{U} = \mathbf{J}_N/N$, where $\mathbf{J}_N$ indicates a matrix of size $N \times N$ containing ones in all cells.

3.2.3 Factor Orthogonality

The analysis of experimental data becomes much easier when all factors in an experiment are mutually orthogonal, as is the case in most conventional experimental designs. The concept of orthogonal factors is formally defined next from two distinct perspectives. This provides methods to check whether the factors in an experiment satisfy this property, either using their projection matrices or the relative sizes of their classes.

Definition from Subspaces Two factors $\mathcal{F}$ and $\mathcal{G}$ are mutually orthogonal (written $\mathcal{F} \perp\!\!\!\perp \mathcal{G}$) if the subspace $V_\mathcal{F} \cap V_{\mathcal{F}\vee\mathcal{G}}^\perp$ is orthogonal to the subspace $V_\mathcal{G} \cap V_{\mathcal{F}\vee\mathcal{G}}^\perp$:

$$V_\mathcal{F} \cap V_{\mathcal{F}\vee\mathcal{G}}^\perp \perp V_\mathcal{G} \cap V_{\mathcal{F}\vee\mathcal{G}}^\perp \implies \mathcal{F} \perp\!\!\!\perp \mathcal{G}. \tag{3.14}$$

Equivalent definitions state that $\mathcal{F}$ and $\mathcal{G}$ are orthogonal if the subspace $V_\mathcal{F}$ is orthogonal to $(V_\mathcal{G} \cap V_{\mathcal{F}\vee\mathcal{G}}^\perp)$, or $V_\mathcal{G}$ is orthogonal to $(V_\mathcal{F} \cap V_{\mathcal{F}\vee\mathcal{G}}^\perp)$. These definitions, however, are largely impractical to check the orthogonality of factors in an experiment. Alternatively, one can exploit that two factors $\mathcal{F}$ and $\mathcal{G}$ are mutually orthogonal *if and only if* the product of their respective projection matrices is commutative:

[3] Bailey (2008) expresses the projection matrix as $\mathbf{P}_{V_\mathcal{F}}$, making explicit the subspace to which it corresponds. As Bailey (2015) later does, however, we use this notation only when it is necessary to disambiguate matrices related to diverse subspaces.

$$\mathcal{F} \perp\!\!\!\perp \mathcal{G} \iff \mathbf{P}_{\mathcal{F}}\mathbf{P}_{\mathcal{G}} = \mathbf{P}_{\mathcal{G}}\mathbf{P}_{\mathcal{F}}. \tag{3.15}$$

In addition, if two factors are orthogonal, then the product of their projection matrices equals the projection matrix of their supremum:

$$\mathcal{F} \perp\!\!\!\perp \mathcal{G} \implies \mathbf{P}_{\mathcal{F}}\mathbf{P}_{\mathcal{G}} = \mathbf{P}_{\mathcal{G}}\mathbf{P}_{\mathcal{F}} = \mathbf{P}_{\mathcal{F}\vee\mathcal{G}}. \tag{3.16}$$

A sequence of factors $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n$ forms a *chain* if such factors all satisfy $\mathcal{F}_1 \prec \mathcal{F}_2 \prec \cdots \prec \mathcal{F}_n$. Factors in a chain are all mutually orthogonal. Factors not forming a chain can also be mutually orthogonal, but one would need to check specifically.

The equivalence in Eqn (3.15) leads to a somewhat surprising result: since the product of $\mathbf{P}_{\mathcal{F}}$ by itself is trivially commutative, every factor is orthogonal to itself. This result seems to defy the usual interpretation of orthogonality. The definition of orthogonal factors in Eqn (3.14) helps clarify this apparent contradiction. If $\mathcal{F} \equiv \mathcal{G}$, their supremum is the factor itself, so $V_{\mathcal{F}\vee\mathcal{G}} = V_{\mathcal{F}} = V_{\mathcal{G}}$. Eqn (3.14) then becomes:

$$(V_{\mathcal{F}} \cap V_{\mathcal{F}}^{\perp}) \perp (V_{\mathcal{F}} \cap V_{\mathcal{F}}^{\perp}) \implies \mathcal{F} \perp\!\!\!\perp \mathcal{F}.$$

This may seem like a dead end, since it requires a subspace (i.e., $V_{\mathcal{F}} \cap V_{\mathcal{F}}^{\perp}$) to be orthogonal to itself, which is generally impossible. There is one exception, though. The only self-orthogonal vector space that can be constructed is $\{\mathbf{0}\}$: the vector space consisting of a single vector filled with zeroes. It is trivial to check that $\{\mathbf{0}\}$ is orthogonal to itself, since the scalar product $\langle \mathbf{0}, \mathbf{0} \rangle$ is obviously 0. This is exactly what happens here. The intersection between any subspace (e.g., $V_{\mathcal{F}}$) and its orthogonal complement (e.g., $V_{\mathcal{F}}^{\perp}$) is limited to $\{\mathbf{0}\}$. The subspace $V_{\mathcal{F}} \cap V_{\mathcal{F}}^{\perp} = \{\mathbf{0}\}$ is thus orthogonal to itself, which implies that factor $\mathcal{F}$ is also orthogonal to itself.

Definition from Classes A further equivalent definition of factor orthogonality may be useful in some circumstances. Two factors are orthogonal if, within each class of their supremum $\mathcal{F}\vee\mathcal{G}$, the size of the class of $\mathcal{F}\wedge\mathcal{G}$ that contains ω is proportional to the product of the sizes of the $\mathcal{F}$- and $\mathcal{G}$-classes that also contain ω. Formally, this is written:

$$|\mathcal{F}[\omega]| \times |\mathcal{G}[\omega]| = c_j \times |(\mathcal{F}\wedge\mathcal{G})[\omega]| \times |(\mathcal{F}\vee\mathcal{G})[\omega]| \tag{3.17}$$

with $c_j \in \mathbb{R}$ constant for all plots belonging to the same class j of the supremum. The formulation in Eqn (3.17) differs slightly from the one Bailey (2015) reports. In particular, the version here includes c_j to further emphasise that both sides of the equation should

keep the same proportionality ratio within each class of the supremum, which may differ from 1. This point remains unclear in Bailey's formulation.

Non-Orthogonality Including non-orthogonal factors in an experiment hampers the analysis of its measurements. Re-arranging and/or adding experimental material before actually conducting the experiment might suffice to make problematic factors become orthogonal to the rest. For instance, the boundaries of factor levels discretised from continuous variables can often be redefined to generate groups orthogonal to those from other variables. Alternatively, problematic factors can be ignored if expert knowledge suggests that they are not relevant for the response. Otherwise, the Calculus of Factors approach might need to be replaced by a less restrictive alternative, such as the method of least squares, as Hinkelmann (2015) notes.

3.2.4 Orthogonal Decomposition

Orthogonal factors permit decomposing measurements into contributions from such factors using projections into derived subspaces. The general concepts of this procedure are introduced in what follows.

***W*-subspaces** The V subspaces capture all information from the measurements related to the levels of a factor, both originated from itself and carried over from others to which it is chained. $W_{\mathcal{F}}$ is the subspace of $V_{\mathcal{F}}$ containing the information that the factor $\mathcal{F}$ alone contributes, removing any that other factors embed in its levels. Denote $\mathscr{F}$ a set of non-equivalent factors including $\mathcal{F}$. The $W_{\mathcal{F}}$ subspace is then defined as:

$$W_{\mathcal{F}} = V_{\mathcal{F}} \cap (\sum_{\mathcal{G} > \mathcal{F}} V_{\mathcal{G}})^{\perp} = V_{\mathcal{F}} \cap \bigcap_{\mathcal{G} > \mathcal{F}} V_{\mathcal{G}}^{\perp} \tag{3.18}$$

where $\sum_{\mathcal{G} > \mathcal{F}} V_{\mathcal{G}}$ indicates the span of the union of the V-subspaces associated with all factors in $\mathscr{F}$ coarser than $\mathcal{F}$, while $\bigcap_{\mathcal{G} > \mathcal{F}} V_{\mathcal{G}}^{\perp}$ refers to the intersection of all subspaces orthogonal to those same $V_{\mathcal{G}}$.

Decomposing the whole data space into orthogonal pieces requires all W-subspaces from non-equivalent factors to be mutually orthogonal. If all $(\mathcal{F}, \mathcal{G}) \in \mathscr{F}$ satisfy (a) $\mathcal{F} \perp\!\!\!\perp \mathcal{G}$,

and (b) $(\mathcal{F} \vee \mathcal{G}) \in \mathscr{F}$, then (i) $W_\mathcal{F}$ is orthogonal to $W_\mathcal{G}$, and (ii) $V_\mathcal{F}$ is the orthogonal direct sum of all $W_\mathcal{G}$ for $\mathcal{G} \succcurlyeq \mathcal{F}$.

Effects Given an arbitrary data vector $\mathbf{y}$, the *effect* of factor $\mathcal{F}$ on $\mathbf{y}$ is defined as the projection of the data vector onto the W-subspace of $\mathcal{F}$ (i.e., $\mathbf{P}_{W_\mathcal{F}}\mathbf{y}$). Since all W-subspaces are mutually orthogonal, the projection of the data vector onto $W_\mathcal{F}$ is orthogonal to the projection onto the W-subspace associated with any other factor in the set. The projection of $\mathbf{y}$ onto the V-subspace, on the other hand, accumulates the individual effects of all factors coarser or equivalent:

$$\mathbf{P}_{V_\mathcal{F}}\mathbf{y} = \sum_{\mathcal{G} \succcurlyeq \mathcal{F}} \mathbf{P}_{W_\mathcal{G}}\mathbf{y}. \tag{3.19}$$

Unlike V-subspaces, no shortcut method exists to obtain projection matrices onto W-subspaces, so it is necessary to rely on the general method for constructing a projection matrix from the vector basis of a subspace. In particular, given a matrix $\mathbf{W}$ containing as columns a vector basis for the subspace W, one obtains its associated projection matrix by computing:

$$\mathbf{P}_W = \mathbf{W}(\mathbf{W}^\mathrm{T}\mathbf{W})^{-1}\mathbf{W}^\mathrm{T}. \tag{3.20}$$

Connection with ANOVA The decomposition of observed data into a set of orthogonal contributions directly links to the procedure for ANOVA described in Sec. 3.1.4. ANOVA splits the variability of the data into different sources, and uses their ratio to test whether such sources are distinct. The simplified version presented before only considers two possible sources of variability: the groups or conditions to compare and the residual. The Calculus of Factors provides a more general approach that considers multiple factors as potential sources of variability.

Similar to the procedure in Sec. 3.1.4, the variability attributable to each factor can be estimated by dividing its sum of squares by its degrees of freedom. These values directly relate with the W subspaces. Given the subspace $W_\mathcal{F}$ associated with factor $\mathcal{F}$:

- the degrees of freedom $d_\mathcal{F}$ coincide with the dimensionality of $W_\mathcal{F}$

$$d_{\mathcal{F}} = dim(W_{\mathcal{F}}); \tag{3.21}$$

- the sum of squares $SS_{\mathcal{F}}$ is the square norm of the projection of the data $\mathbf{y}$ onto $W_{\mathcal{F}}$

$$SS_{\mathcal{F}} = \left\| \mathbf{P}_{W_{\mathcal{F}}} \mathbf{y} \right\|^2. \tag{3.22}$$

The mean squares associated with factor $\mathcal{F}$ are, then, $MS_{\mathcal{F}} = SS_{\mathcal{F}} / d_{\mathcal{F}}$, and the variance ratio is $VR_{\mathcal{F}} = MS_{\mathcal{F}} / MS_{\mathcal{E}}$, since $W_{\mathcal{E}}$ captures the residual variability not associated with any other factor in the experiment. The steps followed to calculate degrees of freedom and sums of squares, however, are far from immediate, requiring to obtain subspaces and their projection matrices. The procedure described next streamlines such calculations.

3.2.5 Calculations on the Hasse diagram

The relationships between factors that Hasse diagrams capture facilitate calculations of degrees of freedom and sum of squares of such factors. This requires two temporary distinct Hasse diagrams, one including only factors related with the plot structure and another including only factors related with the treatment structure. To make it easier to distinguish both diagrams, it is common to use filled (black) dots to indicate plot factors and empty (white) dots to indicate treatment factors, as Fig. 3.2 shows.

(a) Plots (b) Treatments

Figure 3.2: Representation of factors in the plot and treatment sets.

Calculations are based on diagrams merging plot and treatment structures, in which double dots represent factors that belong to both. Since usually more than one subject receives each combination of treatment factors, however, the equality factors from each structure often differ and thus do not appear in the merged diagram with a double dot. In that case, the treatment equality factor can be labelled $\mathcal{T}$, placing its node above $\mathcal{E}$ and connected to the other treatment factors for which it is infimum.

To compute the degrees of freedom, one first identifies the number of classes in each factor. It is often useful to write each such number close to the corresponding node in the

Hasse diagram. Then, the degrees of freedom for any factor $\mathcal{F}$ equal the number of classes of that factor minus the sum of the degrees of freedom of all factors above (and linked to) the one under consideration in the Hasse diagram. This calculation corresponds to the formula:

$$d_{\mathcal{F}} = N_{\mathcal{F}} - \sum_{\mathcal{G}>\mathcal{F}} d_{\mathcal{G}}. \tag{3.23}$$

Starting from the universal factor $\mathcal{U}$, all degrees of freedom can be computed by iteratively descending through the factors at each level of the diagram. One then writes the resulting degrees of freedom together with the number of classes in each node.

Similar to the degrees of freedom, a cascade of calculations yields the sums of squares. The square norm of $\mathbf{P}_{V_{\mathcal{F}}}\mathbf{y}$ is called the *crude* (or *preliminary*) sum of squares of $\mathcal{F}$, which is denoted $CSS_{\mathcal{F}}$. Due to the W-subspaces being orthogonal, $CSS_{\mathcal{F}}$ can be decomposed into a sum of sums of squares:

$$CSS_{\mathcal{F}} = \left\|\mathbf{P}_{V_{\mathcal{F}}}\mathbf{y}\right\|^2 = \sum_{\mathcal{G}\geqslant\mathcal{F}} \left\|\mathbf{P}_{W_{\mathcal{G}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{W_{\mathcal{F}}}\mathbf{y}\right\|^2 + \sum_{\mathcal{G}>\mathcal{F}} \left\|\mathbf{P}_{W_{\mathcal{G}}}\mathbf{y}\right\|^2 = SS_{\mathcal{F}} + \sum_{\mathcal{G}>\mathcal{F}} SS_{\mathcal{G}}. \tag{3.24}$$

Rearranging this formula provides a method to calculate the sum of squares of $\mathcal{F}$:

$$SS_{\mathcal{F}} = \left\|\mathbf{P}_{W_{\mathcal{F}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{F}}}\mathbf{y}\right\|^2 - \sum_{\mathcal{G}>\mathcal{F}} \left\|\mathbf{P}_{W_{\mathcal{G}}}\mathbf{y}\right\|^2 = CSS_{\mathcal{F}} - \sum_{\mathcal{G}>\mathcal{F}} SS_{\mathcal{G}}. \tag{3.25}$$

The starting point of the cascade is always $CSS_{\mathcal{U}}$, which is immediate to compute:

$$CSS_{\mathcal{U}} = \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 = \left\|\mathbf{1}_N\mathbf{y}\right\|^2 = \frac{\left(\sum_{i=1}^{N} y_i\right)^2}{N} = SS_{\mathcal{U}}. \tag{3.26}$$

The remaining crude sums of squares can be calculated using:

$$CSS_{\mathcal{F}} = \sum_{j=1}^{N_{\mathcal{F}}} \frac{\left(\sum_{i=1}^{N_{\mathcal{F}_j}} y_{ij}\right)^2}{N_{\mathcal{F}_j}} \tag{3.27}$$

where $N_{\mathcal{F}_j}$ represents the size of the j-th $\mathcal{F}$-class; if $\mathcal{F}$ is uniform, $N_{\mathcal{F}_j} = K_{\mathcal{F}}\ \forall\, j$.

3.2.6 Analysis of Conventional Experimental Designs

The Calculus of Factors framework permits analysing the measurements from any experimental design whose factors are mutually orthogonal, This includes the widely adopted designs described in Sec. 3.1.3, whose handling within such framework is described next.

Completely Randomised Design A Completely Randomised Design (CRD) assumes no structure in both plots and treatments, which leads to a Hasse diagram as the one shown in Fig. 3.3. In vectorial form, the structural model in Eqn (3.1) changes into:

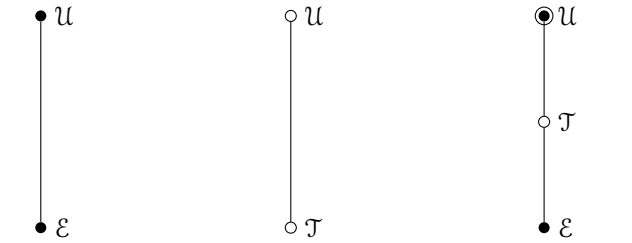
$$\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\tau} + \boldsymbol{\epsilon} \tag{3.28}$$

where $\boldsymbol{\tau}$ is an unknown vector in $W_{\mathcal{T}}$, the W-subspace associated with the treatment factor $\mathcal{T}$. This representation is often called an *effects* (or *null*) model, whose implicit question of interest is reflected in the following hypotheses pair:

$$\begin{aligned} &H_0 : \tau_1 = \tau_2 = \ldots = \tau_T = 0 \\ &H_1 : \tau_i \neq 0 \text{ for at least one } i. \end{aligned}$$

In other words, the effects model tests whether the effect of at least one of T treatments differs from the average. When all treatments have the same mean effect, all deviations from the overall mean must be 0. The effects model assumes a split of the data space into three orthogonal subspaces:

$$V = V_0 \oplus W_{\mathcal{T}} \oplus V_{\mathcal{T}}^{\perp}. \tag{3.29}$$



(a) Plots (b) Treatments (c) Combined

Figure 3.3: Hasse diagrams of a generic CRD.

Computing in cascade, starting from the top of the combined Hasse diagram, the degrees of freedom in a CRD are:

$$\begin{aligned} d_{\mathcal{U}} &= n_{\mathcal{U}} = 1 \\ d_{\mathcal{T}} &= n_{\mathcal{T}} - d_{\mathcal{U}} = T - 1 \\ d_{\mathcal{E}} &= n_{\mathcal{E}} - (d_{\mathcal{U}} + d_{\mathcal{T}}) = N - (1 + (T-1)) = N - T \end{aligned} \tag{3.30}$$

and the sums of squares:

$$
\begin{aligned}
SS_{\mathcal{U}} &= \left\|\mathbf{P}_{W_{\mathcal{U}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 \\
SS_{\mathcal{T}} &= \left\|\mathbf{P}_{W_{\mathcal{T}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} \\
SS_{\mathcal{E}} &= \left\|\mathbf{P}_{W_{\mathcal{E}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{E}}}\mathbf{y}\right\|^2 - (SS_{\mathcal{U}} + SS_{\mathcal{T}}) = \left\|\mathbf{P}_{V_{\mathcal{E}}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 = \left\|\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 .
\end{aligned}
\tag{3.31}
$$

Complete Block Design In a Complete Block Design (CBD), such as the one reflected in Eqn (3.3), plots are grouped into blocks of similar characteristics, with at least one plot per block receiving each treatment. Its structural model in vectorial form is:

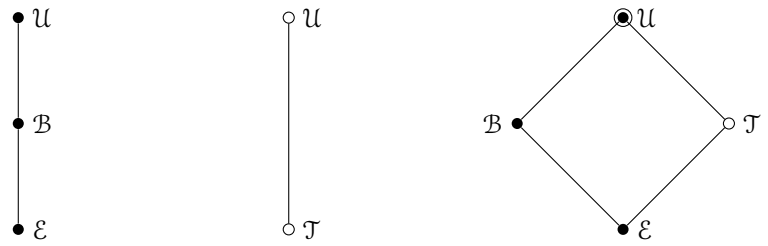
$$\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\tau} + \boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{3.32}$$

This splits the data space as the direct sum of four subspaces, each corresponding to one term in the model:

$$V = V_0 \oplus W_{\mathcal{T}} \oplus W_{\mathcal{B}} \oplus W_{\mathcal{E}}.$$

Note that, aside from the appearance of the W-subspace associated with the blocking factor, the subspace partition includes $W_{\mathcal{E}}$ instead of $V_{\mathcal{T}}^{\perp}$. The residual variance captures all the variability in the data not explained by either treatment or blocking effects, which means that $W_{\mathcal{E}} = (V_{\mathcal{T}} + V_{\mathcal{B}})^{\perp}$.

The most common implementation of a CBD considers blocks of constant size equal to the number of treatments T, so one and only one instance receives each treatment for each block. For simplicity of exposition, the following derivation assumes the experiment includes B blocks all of fixed size T: $K_{\mathcal{B}} = T$. The infimum between the blocking factor $\mathcal{B}$ and the treatment factor $\mathcal{T}$ in this case corresponds to the equality factor $\mathcal{E}$ in the plot set. Figure 3.4 shows the Hasse diagrams constructed under this assumption.



(a) Plots (b) Treatments (c) Combined

Figure 3.4: Hasse diagrams of a CBD, in the particular case of all B blocks of size equal to the number of treatments T.

For a CBD with $K_{\mathcal{B}} = T$, the degrees of freedom are:

$$\begin{aligned} d_{\mathcal{U}} &= N_{\mathcal{U}} = 1 \\ d_{\mathcal{B}} &= N_{\mathcal{B}} - d_{\mathcal{U}} = B - 1 \\ d_{\mathcal{T}} &= N_{\mathcal{T}} - d_{\mathcal{U}} = T - 1 \\ d_{\mathcal{E}} &= N_{\mathcal{E}} - (d_{\mathcal{U}} + d_{\mathcal{T}} + d_{\mathcal{B}}) = N - (1 + (T-1) + (B-1)) \\ &= N - T - B + 1 = T \cdot B - T - B + 1 = (T-1)(B-1) \end{aligned} \tag{3.33}$$

since the total number of observations coincides with all pairwise combinations of treatments and blocks, and the sums of squares:

$$\begin{aligned} SS_{\mathcal{U}} &= \left\|\mathbf{P}_{W_{\mathcal{U}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 \\ SS_{\mathcal{B}} &= \left\|\mathbf{P}_{W_{\mathcal{B}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{B}}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} \\ SS_{\mathcal{T}} &= \left\|\mathbf{P}_{W_{\mathcal{T}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} \\ SS_{\mathcal{E}} &= \left\|\mathbf{P}_{W_{\mathcal{E}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{E}}}\mathbf{y}\right\|^2 - (SS_{\mathcal{U}} + SS_{\mathcal{B}} + SS_{\mathcal{T}}) \\ &= \left\|\mathbf{y}\right\|^2 + \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{B}}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 \end{aligned} \tag{3.34}$$

Factorial Design Any Factorial Design (FD) with equal number of replications per factor combination, such as the one Eqn (3.2) reflects, is orthogonal. For the particular case of $F = 2$, with F representing the number of treatment factors, the vectorial structural model is:

$$\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\tau}_{\mathcal{F}} + \boldsymbol{\tau}_{\mathcal{G}} + \boldsymbol{\tau}_{\mathcal{FG}} + \boldsymbol{\epsilon} \tag{3.35}$$

where $\mathcal{FG}$ represents the interaction between treatment factors $\mathcal{F}$ and $\mathcal{G}$. Each term of the effects model corresponds to an orthogonal subspace of the data space V:

$$V = V_0 \oplus W_{\mathcal{F}} \oplus W_{\mathcal{G}} \oplus W_{\mathcal{T}} \oplus W_{\mathcal{E}} \tag{3.36}$$

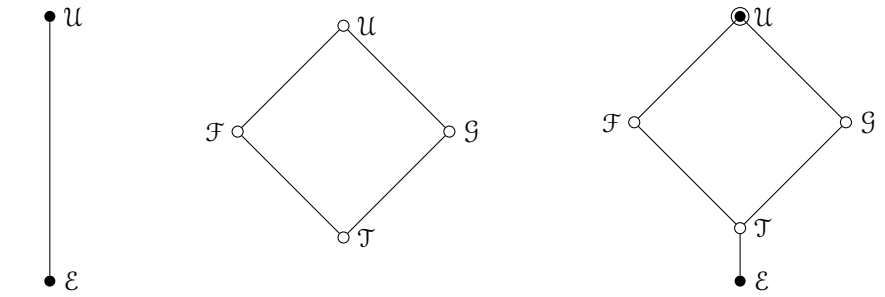
where $W_{\mathcal{T}} = W_{\mathcal{F}\wedge\mathcal{G}}$. Note the subspace for the supremum of the two factors is not included explicitly. In an orthogonal factorial design with equal replication per factor combination, the supremum of the treatment factors coincides with the universal factor $\mathcal{U}$ — the subspace of the supremum matches the subspace of the overall mean.

Adding a third factor, $\mathcal{H}$, leads to the following effects model:

$$\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\tau}_{\mathcal{F}} + \boldsymbol{\tau}_{\mathcal{G}} + \boldsymbol{\tau}_{\mathcal{H}} + \boldsymbol{\tau}_{\mathcal{FG}} + \boldsymbol{\tau}_{\mathcal{FH}} + \boldsymbol{\tau}_{\mathcal{GH}} + \boldsymbol{\tau}_{\mathcal{FGH}} + \boldsymbol{\epsilon}. \tag{3.37}$$

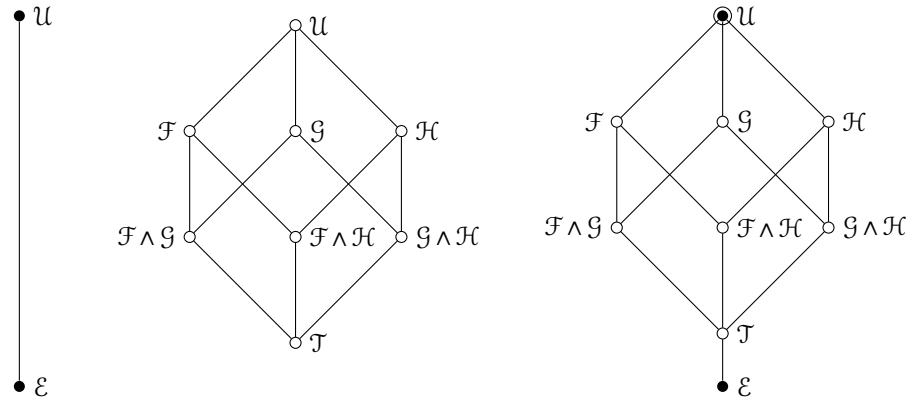
The number of terms in the model increases quickly as soon as further factors are considered. Figures 3.5 and 3.6 show the Hasse diagrams corresponding to $F = 2$ and $F = 3$,

respectively.



(a) Plots (b) Treatments (c) Combined

Figure 3.5: Hasse diagrams of a factorial design with $F = 2$ treatment factors.



(a) Plots (b) Treatments (c) Combined

Figure 3.6: Hasse diagrams of a factorial design with $F = 3$ treatment factors.

The cascading procedure to obtain the degrees of freedom and the sums of squares works exactly as before. For instance, for $F = 2$:

$$\begin{aligned}
d_{\mathcal{U}} &= N_{\mathcal{U}} = 1 \\
d_{\mathcal{F}} &= N_{\mathcal{F}} - d_{\mathcal{U}} = N_{\mathcal{F}} - 1 \\
d_{\mathcal{G}} &= N_{\mathcal{G}} - d_{\mathcal{U}} = N_{\mathcal{G}} - 1 \\
d_{\mathcal{T}} &= N_{\mathcal{T}} - (d_{\mathcal{U}} + d_{\mathcal{F}} + d_{\mathcal{G}}) = N_{\mathcal{F}} N_{\mathcal{G}} - (1 + (N_{\mathcal{F}} - 1) + (N_{\mathcal{G}} - 1)) \\
&\quad N_{\mathcal{F}} N_{\mathcal{G}} - N_{\mathcal{F}} - N_{\mathcal{G}} + 1 = (N_{\mathcal{F}} - 1)(N_{\mathcal{G}} - 1) \\
d_{\mathcal{E}} &= N_{\mathcal{E}} - (d_{\mathcal{U}} + d_{\mathcal{F}} + d_{\mathcal{G}} + d_{\mathcal{T}}) \\
&\quad = N - (1 + (N_{\mathcal{F}} - 1) + (N_{\mathcal{G}} - 1) + (N_{\mathcal{F}} - 1)(N_{\mathcal{G}} - 1)) = N - N_{\mathcal{F}} N_{\mathcal{G}}
\end{aligned} \tag{3.38}$$

and

$$
\begin{aligned}
SS_{\mathcal{U}} &= \left\|\mathbf{P}_{W\mathcal{U}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V\mathcal{U}}\mathbf{y}\right\|^2 \\
SS_{\mathcal{F}} &= \left\|\mathbf{P}_{W\mathcal{F}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V\mathcal{F}}\mathbf{y}\right\|^2 - SS_U \\
SS_{\mathcal{G}} &= \left\|\mathbf{P}_{W\mathcal{G}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V\mathcal{G}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} \\
SS_{\mathcal{T}} &= \left\|\mathbf{P}_{W\mathcal{T}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V\mathcal{T}}\mathbf{y}\right\|^2 - (SS_{\mathcal{U}} + SS_{\mathcal{F}} + SS_{\mathcal{G}}) \\
&\quad \left\|\mathbf{P}_{V\mathcal{T}}\mathbf{y}\right\|^2 + \left\|\mathbf{P}_{V\mathcal{U}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V\mathcal{F}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V\mathcal{G}}\mathbf{y}\right\|^2 \\
SS_{\mathcal{E}} &= \left\|\mathbf{P}_{W\mathcal{E}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V\mathcal{E}}\mathbf{y}\right\|^2 - (SS_{\mathcal{U}} + SS_{\mathcal{F}} + SS_{\mathcal{G}} + SS_{\mathcal{T}}) \\
&\quad \left\|\mathbf{y}\right\|^2 - (SS_{\mathcal{U}} + SS_{\mathcal{F}} + SS_{\mathcal{G}} + SS_{\mathcal{T}})
\end{aligned}
\tag{3.39}
$$

To derive these values for $F \geq 3$, it can be useful to keep in mind that $N_{\mathcal{F}\wedge\mathcal{G}} = N_{\mathcal{F}} \times N_{\mathcal{G}}$ in orthogonal factorial designs.

3.3 Design and Analysis of Classification Experiments

Despite being pervasive in many disciplines, classification experiments have not received much attention from an experimental design perspective. Langley (1988) argued that experiments in Machine Learning (ML), as a discipline "of the artificial", demand less effort to achieve the necessary rigour compared with sciences subject to the uncertainty of the physical world. Information Retrieval (IR), on the other hand, already had a long tradition of relying on statistical methods in their evaluations (Jones, 1981). Cohen (1995) reviews empirical practices for both expert systems and learning algorithms, including experiments that largely follow the paradigm described in Sec. 2.3 and statistical tools for analysing their results. The classification experiment paradigm has remained virtually unchanged since despite some authors having discussed the soundness of some conventional choices (e.g., Hand, 2006; Salzberg, 1999), including particular components of the pipeline such as resampling strategies (Dietterich, 1998) and performance metrics (Hand, 2012), and even the rationale for experimental performance assessment as a whole (Drummond, 2006, 2008).

Having previously introduced the necessary statistical concepts, this section describes the core experimental design and inference practices that underlie conventional classification experiments. The focus largely lies on the evaluation of learning algorithms, since much of the analysis in that regard has been conducted in that context (e.g., Alpaydin, 2014; Eugster, 2011; Hothorn et al., 2005). This means that, unless explicitly stated other-

wise, the pipeline in Fig. 2.2 simplifies, since the feature extractor e works previous to the experiment to create the input data. These data take the form of a *dataset* $\boldsymbol{D}$ comprised of N instances $d_n = (f_n, a_n)$, with $f_n = e(r_n)$. Similar to Sec. 3.1.4, the review below first deals with the particular case of pairwise comparisons between learning algorithms and later generalises to pools of algorithms of arbitrary size. Aside from reporting inferential tests common in the literature, as most texts in the topics usually do, this review also attempts to relate assumptions about the measurements with the fundamental concepts in experimental design.

3.3.1 Comparing Two Algorithms with Unstructured Measurements

A naive approach for comparing two learning algorithms, ℓ_1 and ℓ_2, on $\boldsymbol{D}$ would be to both train and test systems using the whole dataset. This would yield two performance measurements, y_1 and y_2, that one could directly compare. From an experimental design perspective, this naive approach is obviously problematic, failing to hold the principle of replication. Only one observation exists for each treatment in the experiment, the learning algorithms. This would be true even if one considered the responses as vectors of N losses instead of summary metrics, since a single realisation of each treatment would yield all measurements. Responses of this kind conflate the effect of the treatments with that of their particular realisations, thus precluding disentangling the contribution of the learning algorithms and the instances of the dataset — i.e., they are confounded.

A common approach to counter the lack of replication is to train multiple systems from each learning algorithm, with each system using a different subset of instances from $\boldsymbol{D}$. Let K_1 be the number of systems trained with ℓ_1, and K_2 with ℓ_2, each yielding a performance measurement. Assume one ignores any possible relationship between measurements other than their associated learning algorithm, i.e., their level in the factor $\mathcal{L}$, with $\mathcal{L}(i) \in \{\ell_1, \ell_2\}$. This then implicitly corresponds to a Completely Randomised Design (CRD), whose structural model mirrors the one in Eqn (3.1):

$$y_i = \mu + \tau_{\mathcal{L}(i)} + \varepsilon_i. \tag{3.40}$$

One could then use a two-group t-test such as the one in Sec. 3.1.4 to check for differences in performance between the two algorithms, replacing the group sizes N_1 and N_2 in the

formulas with the number of systems K_1 and K_2.

Readers familiar with classification experiments might find assuming lack of structure in the measurements unsettling. As shown next, common practices enforce relationships between measurements, yet one often ignores such relationships when analysing results (e.g., computing and reporting mean and standard deviation for each group).

3.3.2 Comparing Two Algorithms with Related Measurements

The resampling strategies commonly used in classification experiments introduce an inherent structure into the observations. Such strategies generate pairs of training and testing materials, each contributing to one measurement per learning algorithm. Given two response vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ of equal size K, this means each dimension in $\mathbf{y}_1$ directly relates with one in $\mathbf{y}_2$, since they share all their experimental material. In the language of DoE, these relationships form K blocks of units, with one replicate of each treatment level per block — a Complete Block Design (CBD). The structural model thus mirrors the one in Eqn (3.3):

$$y_i = \mu + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \varepsilon_i \tag{3.41}$$

with $\mathcal{K}$ a factor with K levels, one per train/test pair — e.g., the folds of a K-fold Cross-Validation resampling.

Matched Samples t-test The standard approach for comparing the performance of two learning algorithms on a single dataset acknowledges the inherent structure in the measurements, replacing the test statistic in Eqn (3.4) with a version of the t-test targeted for matched (or dependent) samples. In particular, the test statistic is defined as:

$$t = \frac{\bar{y}_1 - \bar{y}_2}{\bar{S}_d/\sqrt{K}} \tag{3.42}$$

with the standard deviation of the differences $\bar{S}_d$ being:

$$\bar{S}_d = \sqrt{\frac{\sum_{k=1}^{K}\left((y_{1k} - y_{2k}) - (\bar{y}_1 - \bar{y}_2)\right)}{K-1}} = \sqrt{\frac{\sum_{k=1}^{K}\left(y_{1k} - y_{2k}\right)^2 - \left(\sum_{k=1}^{K}\left(y_{1k} - y_{2k}\right)\right)^2/K}{K-1}}. \tag{3.43}$$

As before, to determine whether differences in mean performance are statistically significant, the test statistic is compared with a Student's t distribution, in this case with $K-1$ degrees of freedom.

The t-test makes strong assumptions about the distribution of the measurements and the underlying experimental material, such as normality and equal variance. These assumptions may not hold in performance measurements from classification experiments, regardless of the number of such measurements one collects. Some thus argue that non-parametric alternatives making no assumptions regarding the underlying distributions are necessary. One such alternative is McNemar's test (Dietterich, 1998; McNemar, 1947), which is described next.

McNemar's Test Japkowicz and Shah (2011) and Alpaydin (2014) present the McNemar's test as assuming a single hold-out train/test partition, with the related measurements being the predictions on each of N_p test instances. A contingency table counts the number of such instances according to whether the systems trained with each learning algorithm reproduce the ground truth labels. Let Y_{gh} be the number of instances in each cell of the contingency table, with $g, h \in 0, 1$, g indicating the response from the first algorithm and h the second, and 0 representing an error and 1 a success (e.g., Y_{01} is the number of instances that the first algorithm missclassifies but the second predicts correctly). The test statistic in a McNemar's test is, then:

$$\chi^2_M = \frac{(|Y_{01} - Y_{10}| - 1)^2}{Y_{01} + Y_{10}} \tag{3.44}$$

which follows a χ^2 distribution with 1 degree of freedom under the null hypothesis of equal error rates.

An issue often overlooked with the use of the McNemar's test in this manner is the lack of replicates, since the algorithms and their particular realisations are completely conflated. To the best of our knowledge, no generalisation of such test exists for predictions obtained after a resampling process. It thus remains unclear whether one could simply join all individual predictions from K train/test pairs associated with each algorithm (N in the case of K-fold Cross-Validation) and construct a contingency table relating successes and errors as in the conventional test. According to Dietterich (1998), however, a

McNemar's test on a single testing sample may have lower Type I error probability than various configurations of cross-validated t-tests.

Sign and Signed-Rank Tests Tests based on local superiority provide a way to compare algorithms on multiple samples without requiring the strong assumptions of parametric tests. The most basic of these is the Sign Test, in which the performance measurements over K testing samples are compared, counting the number of times each algorithm outperforms the other. Under the null hypothesis of equal expected performance, the counts follow a Binomial distribution, on which one compares the number of "victories" with the expected critical value for a given significance level.

The Wilcoxon's Signed-Rank test loosens the strict local superiority restrictions of the Signed Test, acknowledging that small local inferiority is compatible with overall superiority. The test works as follows. For each of K testing samples, one obtains the local difference in performance $d_k = y_{1k} - y_{2k}$ and ranks them according to their absolute value. The test statistic T_W is the minimum between two values, W_1 and W_2, such that:

$$W_1 = \sum_{k=1}^{K} I(d_k > 0)\ rank(d_k) + \frac{1}{2}\sum_{k=1}^{K} I(d_k = 0)\ rank(d_k)$$
$$W_2 = \sum_{k=1}^{K} I(d_k < 0)\ rank(d_k) + \frac{1}{2}\sum_{k=1}^{K} I(d_k = 0)\ rank(d_k)$$

where $I(\cdot)$ is an indicator function. The critical value of T_W is tabulated for $K \leq 25$; otherwise, its distribution can be treated as approximately normal.

3.3.3 Comparing Multiple Algorithms

For the general case of an experiment comparing $L \geq 2$ learning algorithms, it is not uncommon for studies to compute all pairwise comparisons using t-tests. As mentioned in Sec. 3.1.4, this is problematic since as the number of tests increases, also does the probability of Type I error. Adjustments to the significance level, such as the Bonferroni correction, aim to compensate for this issue. Alternatively, one can use a two-step process that only checks for pairwise differences (*contrasts*) if an *omnibus* test identifies any such difference exists.

Both parametric and non-parametric methods exists for this process, but the focus here is on the former, since they permit estimating contributions to measurements and

not only determining whether differences are significant. Parametic approaches, however, rely on strong assumptions that data from classification experiments might violate. If the reader is interested, Japkowicz and Shah (2011) provide a thorough overview of non-parametric alternatives, such as the Friedman-Nimenyi tests. These rely on a principle similar to the Wilcoxon test described above.

Omnibus Test: ANOVA The most widely accepted procedure for assessing the differences in performance between multiple algorithms on a classification experiment is to conduct an Analysis of Variance (ANOVA) of the measurements. ANOVA was previously introduced in its simplest form in Sec. 3.1.4, and the analysis approach described in Sec. 3.2 extends it to arbitrary factor structures as long as they satisfy orthogonality conditions. The models in Eqn (3.40) and (3.41) are orthogonal and properly represent the case of multiple algorithms if $N_{\mathcal{L}} = L$ — i.e, if the number of levels of the treatment factor is L.

Assume the measurements come from a resampling strategy generating K train/test pairs, and one acknowledges the structure this embeds in the measurements by modelling them as a CBD such as the one in Eqn (3.41). The hypothesis pair of interest considers no difference between the effects of the algorithms in the performance measurements:

$$H_0 : \tau_j = \tau_g \; \forall j, g$$
$$H_A : \exists j, g \; : \; \tau_j \neq \tau_g .$$

with $1 \leq j, g \leq J$, and τ_j the parameter associated with the fixed effect of algorithm ℓ_j. Eugster (2011) suggests modelling the parameters for the blocking variables β_k as random effects with $\beta_k \sim \mathcal{N}(0, \sigma_k^2)$, and the residuals $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. He justifies assuming normality in the residuals, and adopting parametric analysis in general, if one uses a resampling strategy that permits generating an arbitrarily high number of train/test pairs, such as the bootstrap that Hothorn et al. (2005) advocate. In that case, the process described in Sec. 3.2.6 for the analysis of CBD experiments can be followed to obtain variance ratios.

The flexibility of ANOVA permits analysing structures of arbitrary complexity. This means other factors of interest can be incorporated into the analysis. In the evaluation of learning algorithms, this is often the case with a dataset factor $\mathcal{D}$, since so-called multi-domain experiments are common. Eugster (2011) considers this introduces further random effects that interact with the effects of the particular samples. In applied Machine

Learning scenarios, such as in MCA studies, the effect of the feature extractors is likely of interest as well.

Post-Hoc Contrasts: Tukey's HSD Inferential analysis with ANOVA illuminates whether the levels of a factor differentially affect a response variable, but not which levels cause those differences. To identify such differences, one needs to compare all pairs. Tukey's Honestly Significant Differences (HSD) is an alternative to multiple pairwise t-tests to this end that keeps constant a standard error level for all comparisons, which avoids the increased risk of Type I error that multiple comparisons usually cause.

The procedure to identify differences in performance of learning algorithms with HSD is as follows. First, one computes the average of the performance measurements corresponding to each learning algorithm, $\bar{y}_j$, and the standard error SE_{HSD} as:

$$SE_{\text{HSD}} = \sqrt{\frac{MS_{\mathcal{E}}}{K}} \tag{3.45}$$

with $MS_{\mathcal{E}}$ the mean squares of the residual (see Sec. 3.2.5). Then, for each pair of algorithms ℓ_j and ℓ_g, one obtains a test statistic:

$$Q_{jg} = \frac{\bar{y}_j - \bar{y}_g}{SE_{\text{HSD}}} \tag{3.46}$$

whose absolute values are compared to the critical values compiled in devoted tables for particular significance levels. The degrees of freedom of the test match the degrees of freedom of the residual ($d_{\mathcal{E}}$). One rejects the null hypothesis of equal mean performance in those pairs for which $|Q_{jg}|$ exceeds the critical value.

3.4 Summary and Forward Look

Statistical Design of Experiments offers language and tools that facilitate the rigorous planning and analysis of empirical studies. Identifying the inherent structures in an experimental pipeline illuminates how various factors affect the measurements. The Calculus of Factors provides a general procedure to determine effects and assess differences between conditions as long as such structures only contain mutually orthogonal factors.

This seems to be the case in a conventional classification experiment, at least when assessing differences in performance between learning algorithms. Later chapters deal with similar analyses for the particular case of MCA experiments.

Applying the fundamental principles of experimental design helps prevent the most evident confounding issues, such as conflations between algorithm and sample effects in classification experiments. They cannot, however, directly fix the issues with the conventional evaluation in MCA and related disciplines that Sec. 2.5.2 highlighted. Those issues largely stem from a disconnection between the intended and actual outcomes of classification experiments, which requires more than statistical rigour to be suitably addressed. The following chapters report efforts targeted towards extending the conventional evaluation methodology in ways that illuminate the reasons behind the behaviour of the assessed systems and methods. The principles and tools of experimental design inform the systematisation of such extended methodology.

Part II

Contributions

CHAPTER 4

UNCOVERING REASONS BEHIND PERFORMANCE OF SCATTERING-BASED MUSIC GENRE RECOGNITION SYSTEMS

Music Content Analysis (MCA) systems trained and tested by Andén and Mallat (2014) reproduce a large amount of the ground-truth of the *GTZAN* Music Genre Recognition collection (Tzanetakis and Cook, 2002), achieving accuracies that are among the highest reported in the literature (Sturm, 2014d). As described in Sec. 2.4, they use Support Vector Machine (SVM) classifiers trained on features extracted from audio by the scattering transform, a non-linear spectrotemporal modulation representation using a cascade of wavelet transforms (Mallat, 2012). The mathematical derivation of the scattering transform enforces invariances to local time and frequency shifts, a desirable property for music description. Due to the complex representations they generate, however, it remains unclear on which cues such systems rely to predict genre labels on *GTZAN*.

Sturm (2014a, 2016b) proposes an evaluation methodology aimed at identifying whether systems rely on information presumably irrelevant for the problem they target to appear successful — whether they act as "horses". This chapter reports work inspired by such methodology, intended to uncover the reasons behind the performance that scattering-based systems achieve on *GTZAN*. The analyses here largely correspond to those reported by Rodríguez-Algarra et al. (2016), and include some previously un-

published results. In particular, system analysis (Sec. 4.1) and deflation manipulations (Sec. 4.2) inform concrete targeted interventions on the partitioning strategy, the learning algorithm and the frequency content of the audio data (Sec. 4.3). These analyses reveal that SVM systems using scattering-based feature representations exploit previously unknown information present at inaudible frequencies to reproduce *GTZAN* annotations.

4.1 System Analysis

Inspecting inside the "black box" of MCA systems helps illuminate which sources of information such systems likely exploit to reproduce the ground-truth of a collection. In particular, one can dissect systems into their feature extraction and classifier components to understand how each contributes to the overall behaviour. Due to the inherent complexity of trained systems, "white box" evaluation of this kind alone rarely suffices to assess their suitability. The insights that inspection provides, however, help target empirical approaches, as Sec. 4.3 shows.

The analysis here is mainly based on closely inspecting the code that Andén and Mallat (2014) provide, since the concrete implementation of their systems does not always reflect exactly the theoretical description that the authors provide. For simplicity, the analysis focuses on systems built using first- and second-layer time-scatering features (i.e., those that Table 2.1 calls `1-L Sc.` and `1&2-L Sc.` extractors). Systems built using `1,2&3-L Sc.` representations can be understood as a further iteration of the process described here. Moreover, time-frequency scattering feature representations (i.e., `TF Sc.` and `TF Adap. Sc.`) append dimensions to those `1&2-L Sc.` extracts, so any information present in the latter will also be available in systems based on the former. The main goal of the analysis is to identify sources of information from the raw data that SVM systems may exploit to make predictions.

4.1.1 `1-L Sc.` and `1&2-L Sc.` Feature Extractors

The procedure to obtain `1-L Sc.` and `1&2-L Sc.` feature representations from audio signals first extends a recording to be of length $2^{21} = 2,097,152$ samples using what the implementation by Andén and Mallat (2014) refers to as "symmetric boundary condition with half-sample symmetry" padding: the $N \approx 5 \cdot 2^{17}$ samples of an $r \in \mathcal{R}_\Theta$ are concatenated

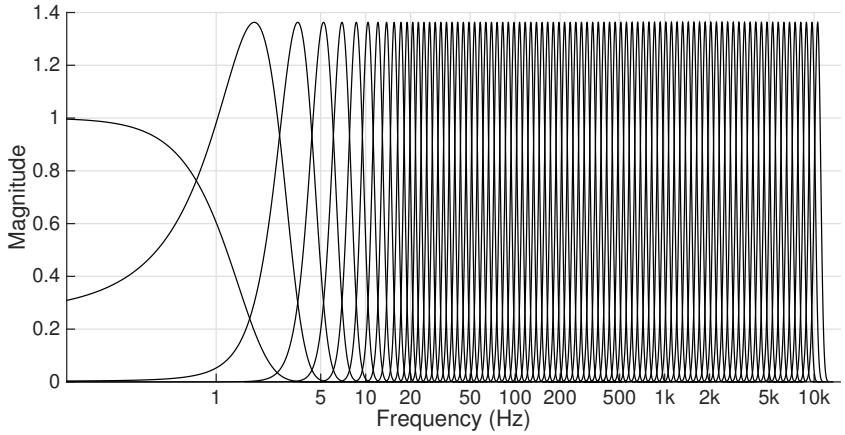
with the same samples but time-reversed, then concatenated with its first ~50,000 samples, and its last ~50,000 samples, and finally the time-reversed samples again. The authors do not provide any justification for this procedure. A Fast Fourier Transform (FFT) then converts this "padded" signal into the frequency domain. Subsequently, the complex spectrum is multiplied by the magnitude response of each of 85 filters of a filterbank designed using the scaling function and dilations of a one-dimensional Gabor mother wavelet with 8 wavelets per octave, up to a maximum dilation of $2^{73/8}$. (The bandwidth of the lowest 11 bands are made constant.) Figure 4.1(a) shows the magnitude responses of the bands of this first filterbank (FB1). Each spectrum product is then reshaped (equivalent to a decimation in the time-domain), transformed to the time domain by the inverse FFT, and then windowed to the portion corresponding to the original signal r in the padded sequences.

Next, the time-series output of each band of FB1 is rectified, padded using the same padding method as above, and transformed into the frequency domain by the FFT. This transformed representation is then multiplied by the magnitude response of each of 25 filters of a filterbank FB2. Figure 4.1(b) shows the magnitude responses of the bands of FB2. These filters are designed with the scaling function and dilations of a one-dimensional Morlet mother wavelet, with 2 wavelets per octave, up to a maximum dilation of $2^{23/2}$. Each FB2 spectrum product is then reshaped (again, equivalent to decimation in time-domain), transformed to the time domain by the inverse FFT, and then windowed corresponding to the original forward-going sequence in the padded sequences. This results in 80 feature vectors of size dependent on the number of scattering layers for each 30-second excerpt.[1] `1-L Sc.` retains only the 85 values related to the DC filter of FB2, computing the natural log of all values (added with a small positive value). `1&2-L Sc.` subsequently takes FB2 time-series outputs with non-negligible energy,[2] "renormalises" each non-zero frequency band (to account for energy captured in the first layer of scattering coefficients), and takes the natural log of all values (added with a small positive value).

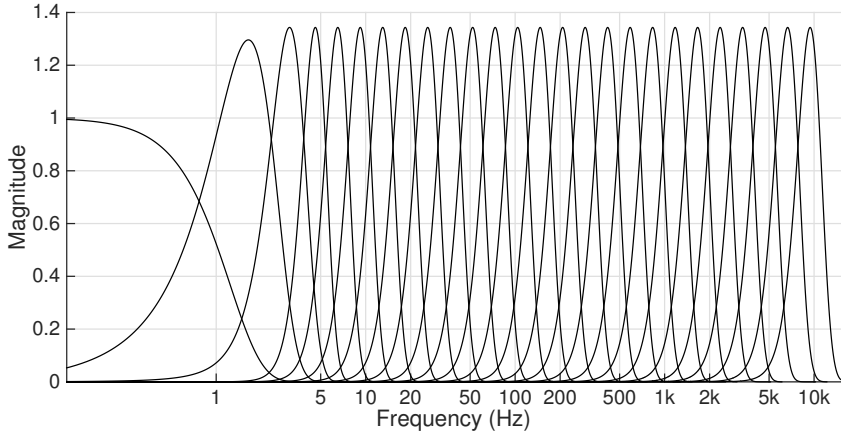
Figure 4.2 relates the dimensions of feature vectors extracted with `1-L Sc.` and

[1] The number of output vectors for each excerpt relates with the size of the employed mother wavelet being 8192 samples. In simple terms, this would be obtained as $30 \cdot 22050 \,/\, 8192 \approx 80.74$, whose closest inferior integer is 80.

[2] In fact, not every rectified FB1 band output is filtered by all FB2 bands because filtering by FB1 will remove all frequencies outside its band.



(a) Filterbank 1 (FB1)



(b) Filterbank 2 (FB2)

Figure 4.1: Magnitude responses of the filterbanks in `1-L Sc.` and `1&2-L Sc.` feature extractors.

`1&2-L Sc.` with the centre frequencies of FB1 and FB2 bands. The bottom-most row is from the scaling function of FB2. The 85 dimensions of a `1-L Sc.` vector are at the bottom, with dimensions [1, 75:85] coming from FB1 bands with centre frequencies below 20 Hz; dimensions [1, 75:85, 737:747] of `1&2-L Sc.` vectors also come from such bands. These are infrasonic frequencies, i.e., frequencies below the threshold of human hearing. Dimensions [2:12] of a `1-L Sc.` vector, and [2:12, 86:268] of a `1&2-L Sc.` vector, are from FB1 bands with centre frequencies above 4186 Hz (pitch C8). All other dimensions are from bands that span the fundamental frequency range of the modern piano.

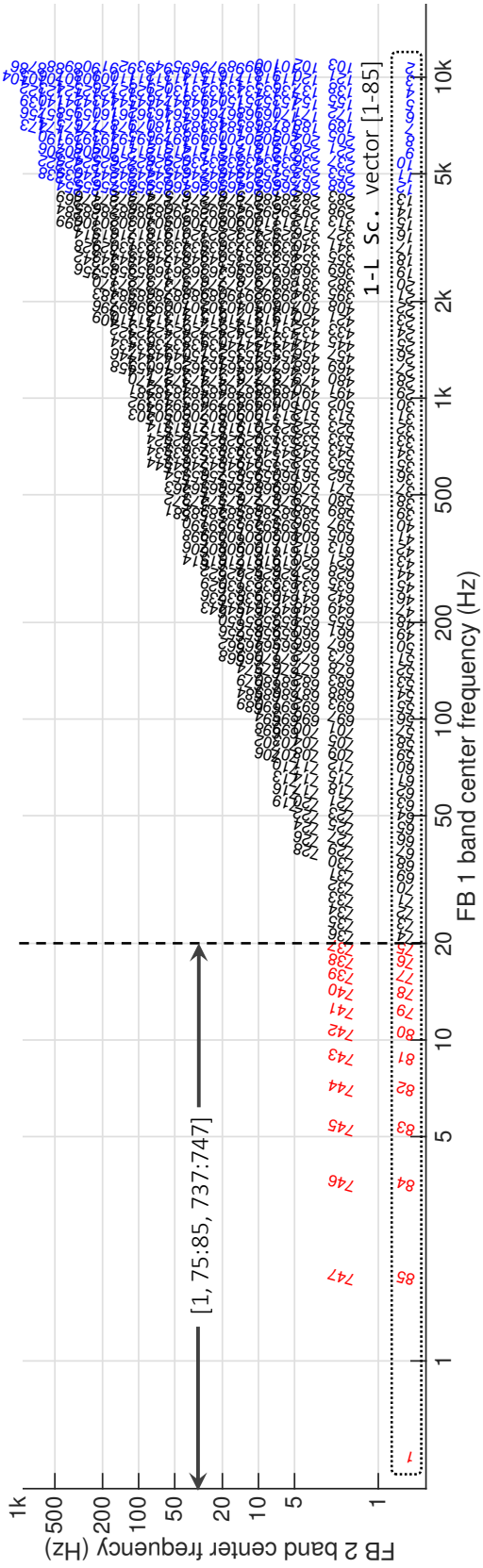


Figure 4.2: Relationship between the dimensions of feature vectors 1-L Sc. and 1&2-L Sc. with the centre frequencies of the bands in FB1 and FB2 filterbanks. Dimensions [1, 75:85] of 1-L Sc. vectors, and [1, 75:85, 737:747] of 1&2-L Sc. vectors, are from bands with centre frequencies below 20 Hz.

4.1.2 SVM Classifier

Define the number of support vectors of a trained SVM as $|SV|$. Classifiers p of the MCA systems by Andén and Mallat (2014) are characterised by a set of support vectors $\mathbf{S} \in \mathbb{F}^{|SV|}$, a Gaussian kernel parameter γ, a weight matrix $\mathbf{M} \in \mathbb{R}^{|SV| \times 45}$, and a bias vector $\boldsymbol{\rho} \in \mathbb{R}^{45}$. (45 is the number of pair-wise combinations of the 10 elements in $\mathcal{U}_{\mathcal{V},A}$, i.e., label 1 vs. label 2, label 1 vs. label 3, etc.) p maps $\mathcal{U}_{\mathbb{F},A'}$ to $\mathcal{U}_{\mathcal{V},A}$ by majority vote from the individual mappings of all elements $f_j \in \mathbb{F}$ of a sequence from $r \in \mathcal{R}_\Theta$ by an SVM classifier p'. p', thus, maps $\mathbb{F}$ to $\mathcal{U}_{\mathcal{V},A}$ by computing 45 pair-wise decisions by means of $sign(\mathbf{M}^T \mathrm{e}^{-\gamma \mathbf{K}(f)} - \boldsymbol{\rho})$, where $\mathbf{K}(f)$ is a vector of squared Euclidean norm of differences between f and all $v_j \in \mathbf{S}$. p' then maps f to $\mathcal{U}_{\mathcal{V},A}$ by majority vote from the 45 pair-wise decisions.

Andén and Mallat (2014) employ `LibSVM`[3] to build p' using a Gaussian kernel with a subset of the feature vectors (downsampled by 2). They optimise the SVM parameters by grid search and 5-fold Cross-Validation on some training recordings. `LibSVM` uses a 1 vs. 1 strategy to deal with multiclass classification, so each support vector receives a weight for each of the nine possible pair-wise decisions involving the class associated with the support vector. The matrix $\mathbf{M}$ contains weights associated with all possible 45 pair-wise decisions. The training of the SVM also generates the vector $\boldsymbol{\rho}$ containing a bias term for each pair-wise decision.

SVM classifiers trained by Andén and Mallat (2014) use all dimensions from their input feature vectors, regardless of the centre frequency of the band from which they originate. This means systems could exploit infrasonic information to predict labels if such information was available. Although this is not normally the case, the faults identified in *GTZAN* (see Sec. 2.4.1) motivate investigating further the potential impact of such frequencies.

4.2 Deflation Manipulations

The system analysis above suggests that scattering-based SVM systems rely on the relative energy levels between spectral bands to discriminate between classes, especially those systems with extractors consisting of a single layer of wavelet transforms. This closely relates to the timbral properties of the audio. Adding further layers and transforms over

[3] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

ID	Extractor	Original ER	Final ER
a	`Mel Sc.`	0.22	0.784
b	`1-L Sc.`	0.208	0.684
c	`1&2-L Sc.`	0.12	0.416
d	`TF Sc.`	0.128	0.368
e	`TF Adap. Sc.`	0.144	0.44
f	`1,2&3-L Sc.`	0.164	0.36

Table 4.1: Overall change in error rate (ER) over 30 steps of random filtering deflation for scattering-based SVM systems in *GTZAN*. (See Table 2.1 for a short description of each feature extractor.)

the frequency domain creates more complex relationships that are less obviously related to musical facets.

Sturm's (2014a) procedure to investigate whether systems exploit particular sources of information may help determine how reliant each scattering configuration is on the spectral shape of the audio. As mentioned in Sec. 2.5.3, this procedure involves iteratively manipulating the input signals in order to break the correlation between the information source of interest and the labels. If the manipulation affects the performances that systems achieve, then such systems must rely on the manipulated information to predict annotations. The exploratory analyses reported here follow this principle.

Random Filtering Figure 4.3 and Table 4.1 summarise the error rates obtained on *GTZAN* over 30 deflation steps with SVM systems using each of the feature representations in Table 2.1. Each data point in the figure corresponds to a single error measurement on a random train/test split used by Kereliuk et al. (2015). Step 1 is unfiltered; the remaining 29 steps each transform recordings predicted correctly on the previous step with randomly modified Near Perfect Reconstruction (NPR) filterbanks. A NPR filterbank splits an input signal into frequency subbands. In this case, 96 subbands are created, each with 129 filter taps, using Lubberhuizen's (2010) implementation. The energy in some randomly selected bands is attenuated a small random amount, changing the energy distribution across the spectrum in a mostly imperceptible manner.

As expected, the results in Figure 4.3 and Table 4.1 suggest that systems using any scattering feature representation rely on the relative energy at various frequency bands of the input to predict the *GTZAN* classes. The transformations, however, affect the systems' performances to different extents, forming two separate groups. The performance of both

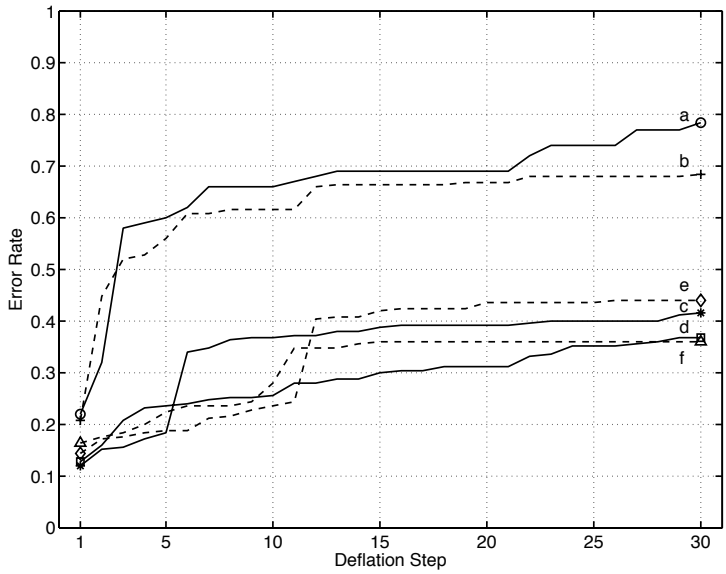


Figure 4.3: Change in error rate over 30 steps of random filtering deflation for scattering-based SVM systems in *GTZAN*. The letters labelling each line correspond to the shortcut identifiers presented in Table 4.1.

`Mel Sc.` and `1-L Sc.` systems decreases quickly, reaching high error rates within the 30 deflation steps. According to Andén and Mallat (2011, 2014) and the system analysis above, these two feature representations should differ only in the scaling of their underlying filterbanks, so it stands to reason that they suffer similarly from changes in the spectral content of the input recordings. The performance of all higher-order scattering features, on the other hand, seems to converge around 40% error rate. This suggests that deeper scattering layers capture information from the audio beyond the "surface" spectral shape.

Incremental Attenuation A modification of the deflation procedure helps illuminate how robust systems are to the scale of the transformations. Instead of transforming completely randomly, one can force all deflation steps to be of the same magnitude, at least on average. One can then gradually increase such magnitude, completing all deflation steps each time. Sturm (2016b) conducts a procedure of this kind changing the scale of a pitch-preserving time-stretching transformation, which shows how performances on distinct classes of the *BALLROOM* collection change differently as the scale increases.

The plots in Fig. 4.4 illustrate deflations of increasing magnitude for scattering-based SVM systems on *GTZAN*. Each measurement corresponds to the final error rate of one

of 10 deflation processes like the one in Fig. 4.3 for a given mean attenuation level (±0.01 dB), up to 9 dB. Error rates tend to increase as the mean attenuation level gets higher for all systems, with those relying on a single filterbank layer (i.e., `Mel Sc.` and `1-L Sc.`) reaching virtually 100% error rate much faster than others. Nevertheless, the trajectories followed by the average error rates are not always monotonic, with a few values being lower than those obtained at previous steps.

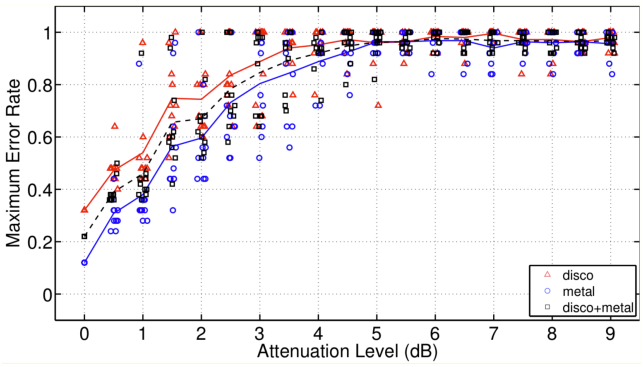
Focusing on some particular classes separately, as in Fig. 4.4, reveals that transformations do not affect all equally. Not only the average error rates differ at each attenuation level, with those in `disco` excerpts (red) apparently higher at lower attenuation levels but lower at higher levels, but also the individual measurements for the same system at the same level vary widely. This variability suggests that deflation processes with random transformations might not reliably capture whether a system exploits a particular source of information. Repeating the procedure multiple times with various magnitudes helps, but requires a possibly excessive number of computations without necessarily revealing the specific source of information that the systems exploit. Leveraging the specific knowledge gained during system analysis, on the other hand, often enables a much more focused approach, as shown below.

4.3 Targeted Interventions

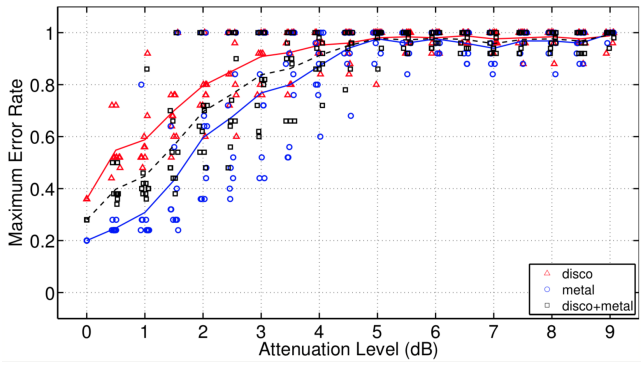
This section reports experiments aimed at illuminating whether a particular factor contributes to the results of scattering-based SVM systems in *GTZAN*. The code for such experiments is available online.[4] Each experiment modifies the classification experiment pipeline in a specific manner. Comparing the results obtained using such a modified pipeline with those from a conventional one reveals whether the modified element contributed to the original results. Each modification thus acts as an intervention (as defined in Sec. 3.1.1), introducing additional evaluation conditions to compare. Chapter 5 systematises and extends this idea.

The experiments here adapt the code implemented by Andén and Mallat (2014) (available online[5]). The original implementation ignores the known faults of *GTZAN* (see

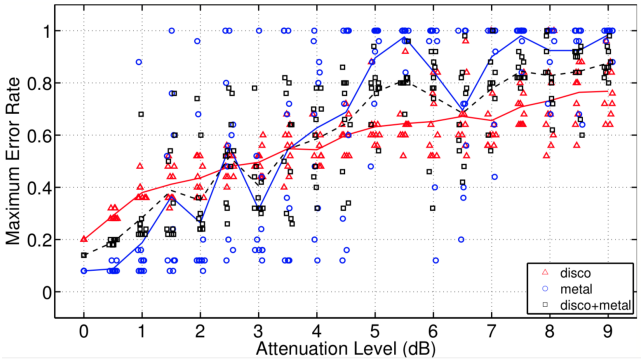
[4] `https://code.soundsoftware.ac.uk/projects/scatter-analysis`
[5] `http://www.di.ens.fr/data/software`



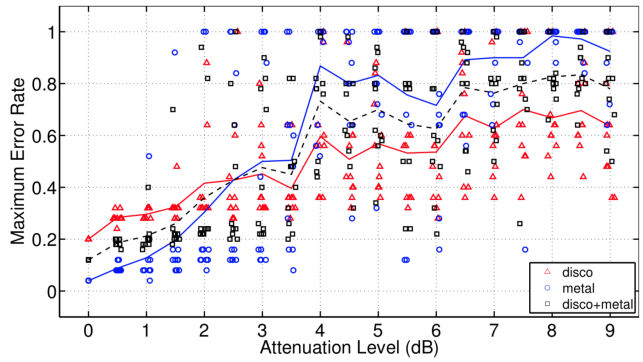
(a) `Mel Sc.`



(b) `1-L Sc.`



(c) `TF Sc.`



(d) `1,2&3-L Sc.`

Figure 4.4: Final error rates at different mean filter attenuation levels across 10 iterations of 30 deflation steps for scattering-based SVM systems, considering both `disco` and `metal` *GTZAN* excerpts. Shapes indicate individual error measurements of each deflation process; lines track average error rates across iterations for each attenuation level.

Sec. 2.4.1), and it is unclear whether scattering-based systems exploit such faults. The experiment reported in Sec. 4.3.1 uses two different train/test partitioning conditions to address this. The experiment in Sec. 4.3.2 replaces the SVM classifier with a Binary Decision Tree (BDT), which helps identify the impact of specific feature dimensions. Finally, the experiment in Sec. 4.3.3 alters the spectral content of test recordings in specific bands, revealing that scattering-based SVM systems trained and tested on *GTZAN* exploit acoustic information below 20 Hz.

4.3.1 Partitioning Intervention

As mentioned in Sec. 2.4.1, the *GTZAN* music collection contains faults, such as repetitions, distortions and mislabellings (Sturm, 2013c). Controlling the faults available through a curated "fault-filtered" train/test partition, such as the one Kereliuk et al. (2015) use, often reduces the amount of ground truth reproduced (see Fig. 4.5). This suggests that the faults in the collection affect the apparent success of evaluated systems (Sturm, 2014d). Therefore, the intervention experiment described here attempts to determine whether such faults also affect scattering-based systems.

Instead of relying on 10-fold stratified Cross-Validation, as Andén and Mallat (2014) do, the two evaluation conditions compared here employ hold-out train/test partitioning to facilitate controlling their contents. The first evaluation condition is `RANDOM`, which mimics the partitioning procedure Andén and Mallat (2014) use: 75% of the recordings of each *GTZAN* class are assigned to the training collection, leaving the remaining 25% for testing. The second is `CURATED`, which employs the "fault-filtered" partitioning procedure that Kereliuk et al. (2015) adopt, but merging training and validation collections. This partitioning procedure accounts for various *GTZAN* faults, removing 70 replicated or distorted recordings, assigning by hand 640 of the remaining recordings to the training collection and the other 290 to testing, avoiding repetition of artists across partitions such as in filtered partitioning (Pampalk et al., 2005). Figure 4.6 shows the number of recordings of each class contained in the `CURATED` partitions.

Due to memory constraints, aside from the partitioning strategy the implementation here also differs from the one Andén and Mallat (2014) use in that it decreases by a factor

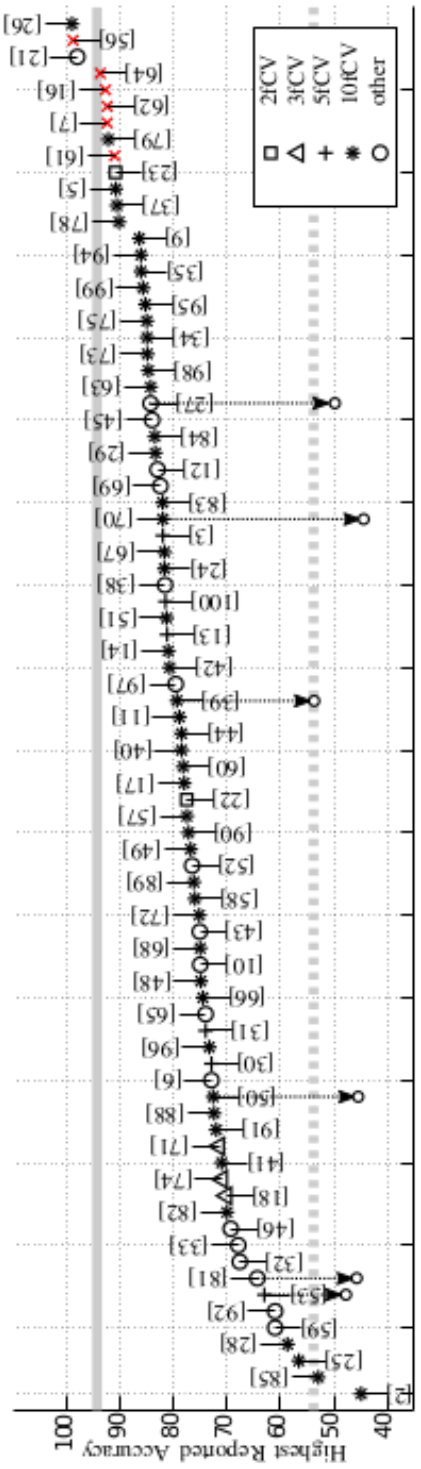


Figure 4.5: Normalised accuracies in *GTZAN* reported in the literature, including re-evaluations. Figure adapted from one by Sturm (2013c). The numbers point to references in that same publication. The arrows indicate the results obtained after accounting for the faults of the collection.

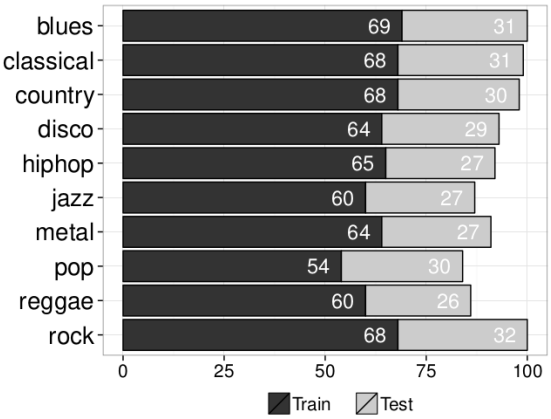


Figure 4.6: Number of recordings from each *GTZAN* class in the training and testing collections of the CURATED evaluation condition.

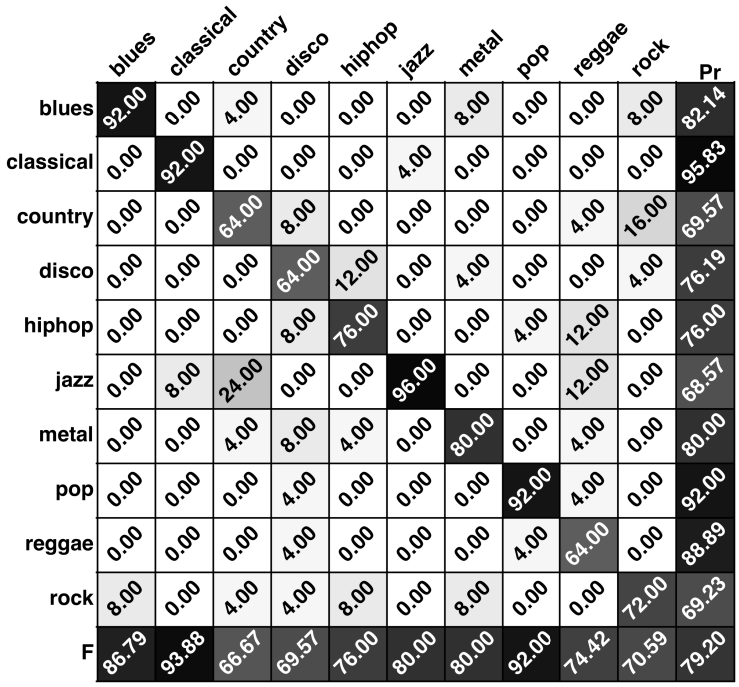
of 4 the number of scattering features in the pre-computation of the Gaussian kernel of the SVM. This reduces the computational cost without sacrificing much performance.[6]

Table 4.2 compares the normalised accuracies (mean recall) of the SVM systems trained here along with those Andén and Mallat (2014) report for the six scattering-based feature representations in Table 2.1. The use of mean recall as metric intends to compensate for class imbalances in the CURATED partitioning condition. The results that Andén and Mallat (2014) report differ slightly from those in RANDOM, but most of them are within reason considering the standard deviations — only TF Adap. Sc. and 1,2&3-L Sc. systems differ more than two standard deviations. Performance increases in RANDOM when including second-order scattering features (1-L Sc. to 1&2-L Sc.). Contrary to what Andén and Mallat (2014) report, however, including third-order features (1&2-L Sc. to 1,2&3-L Sc.) decreases performance.

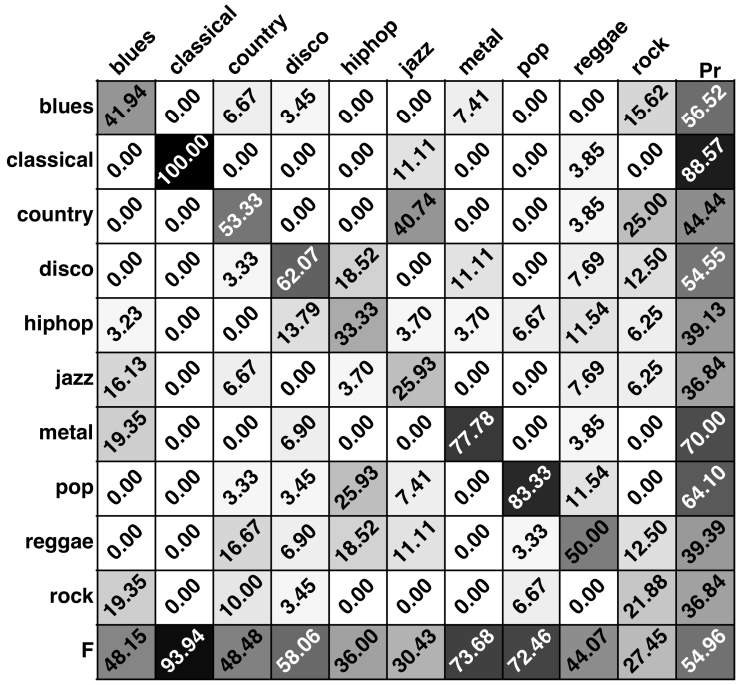
Regarding the consequences of the intervention, all systems decrease performance considerably between RANDOM and CURATED. Systems using TF Adap. Sc. features achieve the highest normalised accuracy in CURATED, which is almost 20 percentage points lower than the highest in RANDOM. The decrease across conditions is of a similar magnitude in all other systems, suggesting these systems exploit the faults of *GTZAN* to artificially inflate their apparent performance.

Figure 4.7 details the per-class performances of 1-L Sc. SVM systems in RANDOM and

[6] This was suggested by Joakim Andén, whose advice deserves acknowledgment.



(a) RANDOM



(b) CURATED

Figure 4.7: Performance measurements (in %) obtained by SVM systems using `1-L Sc.` feature representations on the (a) RANDOM and (b) CURATED *GTZAN* partitioning conditions. Column is ground truth, row is prediction. Far-right column is precision, diagonal is recall, bottom row is F-score, lower right-hand corner is overall normalised accuracy. Off-diagonals are confusions.

	Original Recordings			Filtered Recordings	
Extractor	**Andén and Mallat (2014)**	RANDOM	CURATED	RANDOM	CURATED
Mel Sc.	82.0 % ± 4.2	78.00 %	53.29 %	39.20 %	30.09 %
1-L Sc.	80.9 % ± 4.5	79.20 %	54.96 %	31.60 %	22.42 %
1&2-L Sc.	89.3 % ± 3.1	88.00 %	66.46 %	50.80 %	44.47 %
TF Sc.	90.7 % ± 2.4	87.20 %	68.49 %	62.40 %	55.11 %
TF Adap. Sc.	91.4 % ± 2.2	85.60 %	68.61 %	64.80 %	44.52 %
1,2&3-L Sc.	89.4 % ± 2.5	83.60 %	68.32 %	64.80 %	53.16 %

Table 4.2: Normalised accuracies (mean recall) obtained on *GTZAN* by scattering-based SVM systems by Andén and Mallat (2014) and systems using RANDOM and CURATED partitioning conditions, trained and tested with the original *GTZAN* recordings (left) and versions with information below 20 Hz attenuated (right).

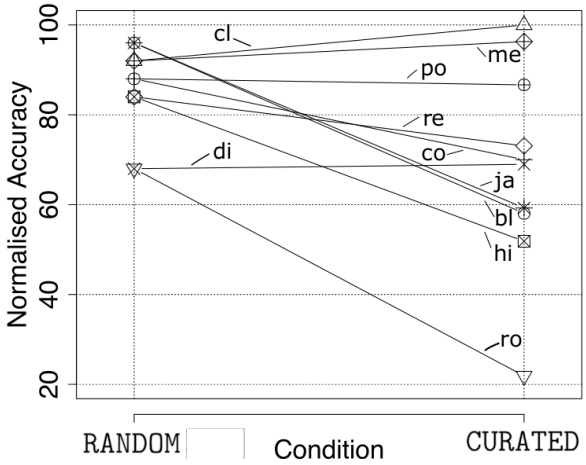


Figure 4.8: Interaction between partitioning conditions and *GTZAN* classes in recall measurements (in %) from SVM systems using TF Adap. Sc. feature representations. bl is blues, cl is classical, co is country, di is disco, hi is hiphop, ja is jazz, me is metal, po is pop, re is reggae, and ro is rock. Trends are similar across feature representations.

CURATED as confusion matrices. All performance measurements for every *GTZAN* class decrease, except for recall and F-measure in classical, which increase. Recall in metal recordings also increases for systems using TF Adap. Sc., as Fig. 4.8 shows. classical moves to perfect recall *for every* feature representation, with an average increase of 8.8 percentage points. These observations suggest that the faults in *GTZAN* affect the overall performance of scattering-based SVM systems in ways unique to each system and each *GTZAN* class.

A principal components decomposition helps to pinpoint the differences between the information available in each condition. Figure 4.9 shows the eigenvectors of the first

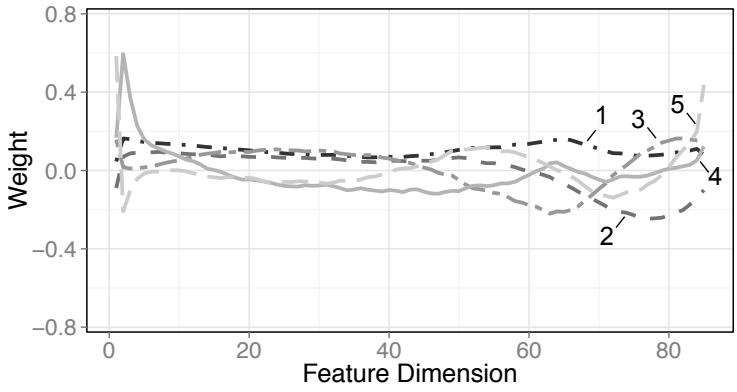
N	RANDOM	CURATED
1	52.99 %	51.82 %
2	65.05 %	65.27 %
3	71.42 %	71.62 %
4	75.53 %	75.80 %
5	79.48 %	79.74 %

Table 4.3: Cumulative percentage of variance captured by each of the first five principal components of the 1-L Sc. feature representations extracted from the training recordings in (a) RANDOM and (b) CURATED partitioning conditions of *GTZAN*.

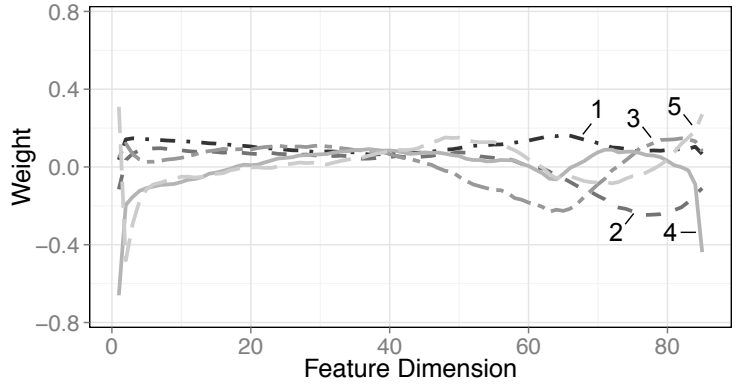
five principal components of first-layer time-scattering feature representations extracted from the training recordings in RANDOM and CURATED partitioning conditions of *GTZAN*. Table 4.3 includes the cumulative percentage of variance captured by each of such components. The feature dimensions considered match the input of 1-L Sc. systems, and are also included in all higher order time-scattering representations. The most striking differences appear in the lowest and highest dimensions of the fourth component, suggesting that these dimensions of the scattering feature representations capture information that differs between the recordings of each condition and may play a role in the performance differences highlighted above. These dimensions correspond to filters centred at frequencies below 20 Hz (See Fig. 4.2). The faults of *GTZAN* seem to relate, at least in part, with acoustic information at inaudible frequencies. Similar to the system analysis in Sec. 4.1, however, this does not mean that the systems actually exploit such information, only that the information is available and differs between conditions. The characteristics of SVM classifiers, however, make it difficult to determine the influence that each individual input feature dimension (or subset of dimensions) has in the overall performance of a system. Replacing the learning algorithm with a more interpretable one, such as the Decision Tree employed next, facilitates linking such dimensions with class predictions.

4.3.2 Classifier Intervention

SVM classifiers generate decision boundaries in multi-dimensional spaces, which benefits prediction but hampers their interpretability. The relevance of each individual dimension of the scattering feature vectors is therefore unclear for SVM systems. The experiments reported here replace the SVM with Binary Decision Trees (BDT), which construct a set of rules defined by linear splits of the feature space one dimension at a time. This construc-



(a) RANDOM



(b) CURATED

Figure 4.9: Eigenvectors of the first five principal components (labelled) of the 1-L Sc. feature representations extracted from the training recordings in (a) RANDOM (79.74% of variance captured) and (b) CURATED (79.48% of variance captured) partitioning conditions of *GTZAN*.

tion method makes BDT systems particularly useful to illuminate which input features are relevant to distinguish between classes. Systems relying on BDT classifiers are considered to be among the easiest to construct and interpret (Alpaydin, 2014), at the cost of potentially less accuracy.

Table 4.4 summarises the normalised accuracies that systems trained with Matlab's BDT algorithm[7] obtain under the two partitioning conditions of *GTZAN* defined in Sec. 4.3.1 for each scattering-based feature representation in Table 2.1. Overall, BDT systems achieve normalised accuracies around 8 percentage points lower in RANDOM than the SVM systems in Table 4.2. The results clearly differ between conditions, with falls in performance similar, if not larger, than those measured for SVM systems. SVM classifiers

[7] http://uk.mathworks.com/help/stats/classificationtree-class.html

Extractor	RANDOM	CURATED
Mel Sc.	72.80 %	45.70 %
1-L Sc.	71.60 %	42.35 %
1&2-L Sc.	80.00 %	49.91 %
TF Sc.	79.20 %	46.81 %
TF Adap. Sc.	79.60 %	44.77 %
1,2&3-L Sc.	79.20 %	46.48 %

Table 4.4: Normalised accuracies (mean recall, in %) obtained in *GTZAN* by scattering-based BDT systems using RANDOM and CURATED partitioning conditions, trained and tested with original *GTZAN* recordings. The description of the feature extractor related with each system is as in Table 2.1.

actually appear to mitigate the potential performance decrease in systems using scattering feature representations with more than one layer. The impact of the *GTZAN* faults in performance measurements thus seems to relate to the feature representations, but the choice of learning algorithm also appears to contribute to the scale of such impact.

In a BDT classifier, each node splits the data according to a threshold value of a single input feature dimension. This permits ranking the input dimensions according to their estimated importance on the class predictions using the following method (Nembrini et al., 2018; Perner, 2011). Define the Gini *impurity* of a node v as $\hat{\Gamma}(v) = \sum_{\forall a \in \boldsymbol{A}} \hat{P}_a(v)(1 - \hat{P}_a(v))$, where $\hat{P}_a(v)$ represents the proportion of instances of class a among those that reach node v. The importance of a node is assumed to relate to the decrease in impurity that it causes, i.e., the difference between the impurity at node v and the weighted sum of the impurities of its two child nodes. Summing the impurity importance values of all nodes in a tree that split using an input dimension thus estimates the importance of such a dimension. Though in different order, the five most important dimensions according to this criterion coincide (1, 2, 4, 84, and 85) in BDT systems using time-scattering feature representations (i.e., 1-L Sc., 1&2-L Sc., and 1,2&3-L Sc.), regardless of the partitioning condition. Dimensions 1, 2 and 85 also appear within the top five in systems including frequency scattering features (i.e., TF Sc. and TF Adap. Sc.). All these dimensions correspond to filters centred below 20 Hz or above 4,186 Hz, as Fig. 4.2 shows, with dimensions 1 and 85 being those closest to the DC component of the signal.

Since BDT classifiers are relatively fast and inexpensive to construct, an alternative procedure to estimate feature dimension importance involves training and testing multiple systems using a single dimension each time. Figure 4.10 shows the proportion of

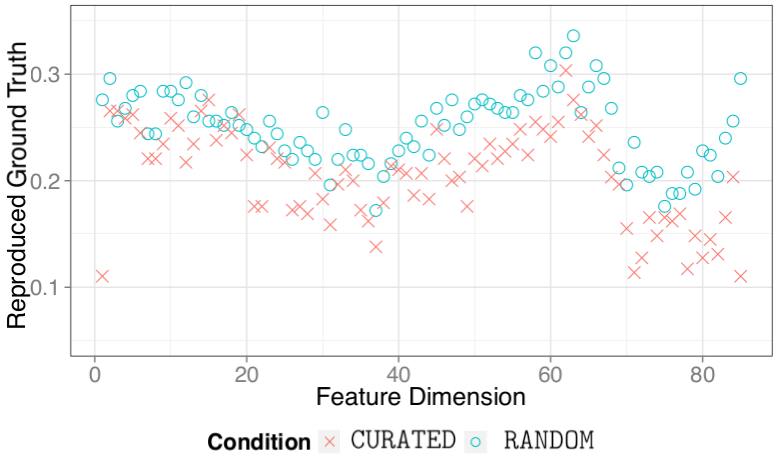


Figure 4.10: Proportion of ground truth annotations from test recordings that BDT systems using single dimensions from a `1-L Sc.` feature extractor correctly predict under `RANDOM` and `CURATED` partitioning conditions of *GTZAN*.

recordings in the test collection of both `RANDOM` and `CURATED` conditions that BDT systems using each of the 85 dimensions of `1-L Sc.` feature representations correctly predict. The most striking differences seem to occur at both extremes of the x-axis, in bands close to or outside the limits of normal human hearing (namely [1, 70:85]). As mentioned above, dimensions 1 and 85 appear highly important according to the Gini impurity criterion for all BDT systems, regardless of the partitioning condition (except maybe those using `Mel Sc.` feature representations, since the scaling of their filterbanks differ). Conversely, systems using exclusively these dimensions differ widely in their performance depending on whether the known faults of *GTZAN* are available. This suggests that patterns at frequencies below 20 Hz exist in *GTZAN* associated with the classes within the training recordings of both partitioning conditions, which the trees capture. Ensuring that no artists appear in both training and testing recordings, among other corrections, seems to break such association in the testing recordings of `CURATED`, leading to more frequent errors.

The conjecture that *GTZAN* recordings contain information at frequencies below 20 Hz linked with the genre labels is further supported by the results that BDT systems obtain using exclusively scattering feature dimensions 1 and 75 to 85. Figure 4.11 shows that such systems achieve a normalised accuracy of 60.40% in `RANDOM`, which is virtually identical to the performance that Tzanetakis and Cook (2002) originally reported for

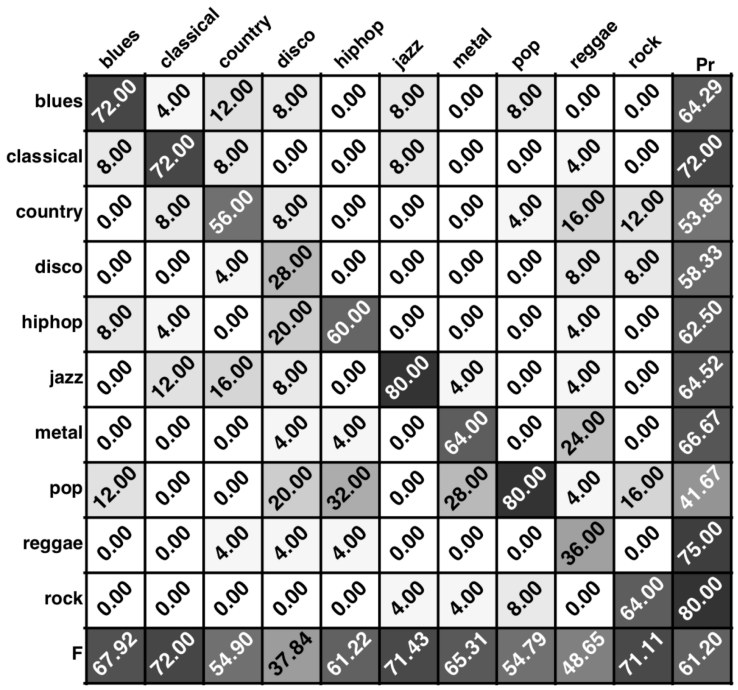
GTZAN. In CURATED, however, the normalised accuracy drops to 39.37%. Adding dimensions 737:747 from 1&2-L Sc. feature representations (modulations from FB2 of information below 20 Hz) only marginally increases the performance in both conditions.

Overall, the analyses conducted using BDT classifiers strongly suggest that the performance of scattering-based systems benefits from the presence of infrasonic information (i.e., below 20 Hz) in *GTZAN*. The intervention reported next addresses whether this hypothesis holds in SVM systems as well.

4.3.3 Filtering intervention

Informed by the previous interventions, the analysis here focuses on assessing the effects of infrasonic information on the results of scattering-based SVM systems in *GTZAN*. Similar to the filtering manipulations reported in Sec. 4.2, an intervention targeted to this end should alter the spectral content of the recordings. In this case, the intervention attenuates by at least 30 dB frequencies below 20 Hz of the test recordings in both RANDOM and CURATED partitioning conditions using a fifth-order Butterworth high-pass filter. The attenuation does not cause any perceptible change in the audio, but affects the amount of ground truth that systems are able to reproduce. This may motivate manipulating the data in a similar manner directly on the training recordings before constructing the systems to avoid the effect appearing in the first place, a possibility that we briefly explore at the end of this section.

Filtered Test Recordings The same SVM systems trained under the partitioning intervention reported in Sec. 4.3.1 are used to predict genre labels on high-pass filtered test recordings corresponding to their partitioning condition. The two right-most columns of Table 4.2 show the normalised accuracies these systems obtain on the filtered recordings. The figures clearly drop for all systems compared to those reported in Sec. 4.3.1, with systems using 1-L Sc. feature representations under RANDOM suffering the most striking performance decrease (close to 50 percentage points). The decrease in performance of systems using deeper scattering layers is smaller but still notable. Systems trained in the CURATED partitioning condition also suffer drops in performance when tested with filtered recordings.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock	Pr
blues	72.00	4.00	12.00	8.00	0.00	8.00	0.00	8.00	0.00	0.00	64.29
classical	8.00	72.00	8.00	0.00	0.00	8.00	0.00	0.00	4.00	0.00	72.00
country	0.00	8.00	56.00	8.00	0.00	0.00	0.00	4.00	16.00	12.00	53.85
disco	0.00	0.00	4.00	28.00	0.00	0.00	0.00	0.00	8.00	8.00	58.33
hiphop	8.00	4.00	0.00	20.00	60.00	0.00	0.00	0.00	4.00	0.00	62.50
jazz	0.00	12.00	16.00	8.00	0.00	80.00	4.00	0.00	4.00	0.00	64.52
metal	0.00	0.00	0.00	4.00	4.00	0.00	64.00	0.00	24.00	0.00	66.67
pop	12.00	0.00	0.00	20.00	32.00	0.00	28.00	80.00	4.00	16.00	41.67
reggae	0.00	0.00	4.00	4.00	4.00	0.00	0.00	0.00	36.00	0.00	75.00
rock	0.00	0.00	0.00	0.00	0.00	4.00	4.00	8.00	0.00	64.00	80.00
F	67.92	72.00	54.90	37.84	61.22	71.43	65.31	54.79	48.65	71.11	61.20

(a) RANDOM

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock	Pr
blues	46.05	5.85	8.70	8.60	8.60	8.65	5.10	12.35	5.80	4.55	40.31
classical	5.65	47.70	6.05	3.30	3.50	7.60	2.30	1.05	5.85	2.00	56.12
country	7.00	7.85	31.60	12.55	5.85	3.85	3.20	5.15	6.95	6.55	34.90
disco	2.35	4.30	8.65	21.85	6.15	2.75	7.00	6.65	12.65	9.50	26.70
hiphop	9.30	7.50	5.75	14.00	37.55	3.05	9.05	15.60	6.40	7.05	32.58
jazz	5.30	11.50	15.55	7.65	3.05	62.10	2.45	2.05	3.90	5.00	52.38
metal	5.90	3.90	4.40	6.60	8.45	2.70	39.30	11.60	16.65	6.80	36.97
pop	11.80	3.45	7.55	13.55	17.15	2.45	17.75	32.05	6.60	8.55	26.51
reggae	4.45	4.20	5.85	7.35	5.45	2.60	7.25	6.35	32.30	6.75	39.13
rock	2.20	3.75	5.90	4.55	4.25	4.25	6.60	7.15	2.90	43.25	51.00
F	42.99	51.57	33.17	24.03	34.89	56.83	38.10	29.02	35.39	46.81	39.37

(b) CURATED

Figure 4.11: Performance measurements (in %) obtained by BDT systems using exclusively dimensions [1, 75:85] from `1-L Sc.` feature representations on the (a) `RANDOM` and (b) `CURATED` *GTZAN* partitioning conditions. Column is ground truth, row is prediction. Far-right column is precision, diagonal is recall, bottom row is F-score, lower right-hand corner is overall normalised accuracy. Off-diagonals are confusions.

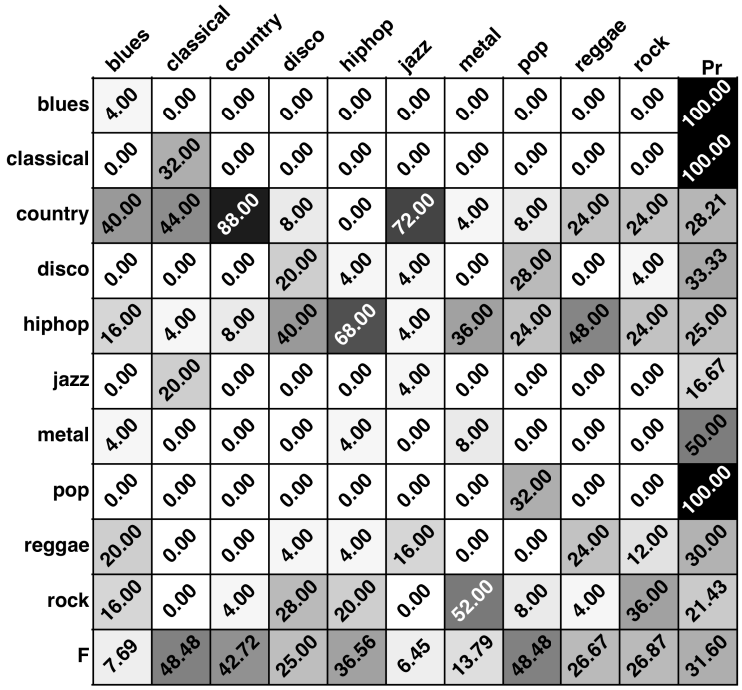


Figure 4.12: Performance measurements (in %) obtained in *GTZAN* by SVM systems trained on recordings from the `RANDOM` partitioning using `1-L Sc.` feature representations tested on recordings with content below 20 Hz attenuated. Column is ground truth, row is prediction. Far-right column is precision, diagonal is recall, bottom row is F-score, lower right-hand corner is overall normalised accuracy. Off-diagonals are confusions.

Figure 4.12 details the per-class performances that an SVM system trained in `RANDOM` with `1-L Sc.` features obtains when tested with high-pass filtered recordings. Compared with the measurements presented in Fig. 4.7(a), most figures decrease drastically when introducing the filtering. Nevertheless, specific measurements for some classes increase, although none in both precision and recall simultaneously (and thus never in terms of F-score). Although changes in performance between `RANDOM` and `CURATED` previously seemed closely linked to the presence of infrasonic information, the magnitude and in some cases the direction of the differences between Figs. 4.7(a) and 4.12 do not always match those between Figs. 4.7(a) and 4.7(b). More precisely, both recall and F-measure decrease instead of increase in `classical`, and the reverse in `country`. This suggests that partitioning and filtering interventions affect different factors influencing the amount of ground truth that systems reproduce, notwithstanding an interaction between them, as Figs. 4.9 and 4.10 suggest.

The results here, together with the previously reported interventions, seem to confirm

		Original Recordings		**Filtered Recordings**	
Extractor	**Learn. Alg.**	RANDOM	CURATED	RANDOM	CURATED
Mel Sc.	SVM	44.80 %	31.07 %	74.00 %	55.45 %
	BDT	62.80 %	37.25 %	68.80 %	43.38 %
1-L Sc.	SVM	30.40 %	23.20 %	73.60 %	52.59 %
	BDT	63.20 %	41.62 %	68.80 %	45.59 %
1&2-L Sc.	SVM	62.00 %	55.65 %	84.00 %	64.92 %
	BDT	68.00 %	50.80 %	72.80 %	50.18 %
TF Sc.	SVM	61.60 %	52.76 %	86.00 %	69.32 %
	BDT	70.40 %	49.44 %	80.00 %	53.17 %
TF Adap. Sc.	SVM	59.20 %	52.11 %	85.20 %	69.59 %
	BDT	71.20 %	56.80 %	80.80 %	54.00 %
1,2&3-L Sc.	SVM	64.80 %	57.02 %	86.00 %	66.15 %
	BDT	72.80 %	48.92 %	75.60 %	49.85 %

Table 4.5: Normalised accuracies (mean recall) obtained in *GTZAN* by scattering-based SVM and BDT systems using RANDOM and CURATED partitioning conditions, trained on recordings with information below 20 Hz attenuated and tested on both original (left) and filtered recordings (right).

that scattering-based SVM systems benefit from the faults of *GTZAN* and exploit infrasonic information to reproduce a large amount of ground truth. Overall, the infrasonic content seems to impact measurements more severely, especially in systems using scattering feature representations of lower order. Conducting both filtering and partitioning interventions jointly, as the right-most column of Table 4.2 shows, leads to substantial performance drops. Every system, however, achieves performances clearly above the random baseline even under such combined evaluation condition.

Filtered Training Recordings As the interventions above show, altering the spectral content at frequencies below 20 Hz of test *GTZAN* recordings affects the apparent performance of scattering-based systems. A reasonable countermeasure could involve high-pass filtering all recordings before training, since one expects a manipulation of this kind to remove all possible infrasonic information and thus lead to systems that avoid relying on such information. The brief analysis here attempts to elucidate whether this expectation holds for scattering-based systems.

Table 4.5 shows the normalised accuracies that SVM and BDT scattering-based systems achieve under the same partitioning conditions as above on both original and high-pass filtered *GTZAN* recordings, when trained using recordings filtered in the same way.

The performance of such systems on filtered test recordings is similar to what systems trained on original recordings achieve on their test counterparts in the partitioning intervention, as Tables 4.2 and 4.4 reflect. On the other hand, figures drop when tested on original recordings, particularly for SVM systems. The decrease is much smaller for BDT systems, leading to normalised accuracies higher than those of all their corresponding SVM systems under `RANDOM` partitioning condition, despite SVM systems consistently outperforming BDT systems whenever they are trained with original recordings and/or tested with filtered ones.

The changes in performance derived from the different acoustic conditions of the test recordings suggest that systems still identify and attempt to exploit patterns in the infrasonic dimensions of the scattering features regardless of whether the training recordings have been filtered. Although some might find this counter-intuitive, it actually makes sense in hindsight. Assume that the relative energy levels of two frequency bands, b_1 and b_2, is consistent across recordings of a class. If both b_1 and b_2 are attenuated in the same manner for all recordings, then their relative energy levels stay unchanged despite their magnitudes being lower. On the other hand, if only b_1 is attenuated, then the relative energy levels change but a new consistent relationship with b_2 appears in the class. Although in reality the patterns captured by scattering features are likely more complex than described here, a similar principle applies, especially in the dimensions corresponding to lower-order representations. As a consequence, regardless of whether recordings have been filtered, the same acoustic condition in training and testing permits systems to exploit infrasonic patterns if they originally exist.

4.4 Discussion

The analyses reported in this chapter illustrate the importance of extending the evaluation of MCA systems beyond counting the amount of ground truth that they reproduce in a benchmark collection. The various steps conducted illuminate, at least partially, the reasons behind the performance of scattering-based SVM systems on *GTZAN*. The systems that Andén and Mallat (2014) propose for MGR seem to rely heavily on some infrasonic information apparently linked with the genre labels in the collection. The presence of

this previously unknown information source, together with the known faults of *GTZAN*, caution against taking reported performance measurements at face value.

System analysis guides the design and implementation of appropriate empirical analyses that illuminate reasons behind performance, both through deflations and targeted interventions. The system analysis of time-scattering systems relates feature dimensions with energy levels at specific frequency bands. The results of deflation processes affecting the spectral content of the recordings show that such systems rely on the relative band energy levels to predict genre annotations, but to a different extent depending on both extractor configuration (e.g, its depth and domain) and *GTZAN* class. Analyses based on deflations, however, may prove excessively demanding in computational resources without necessarily clarifying much beyond what the system analysis alone reveals. This is largely the case in the analysis reported in Sec. 4.2.

Targeted interventions permit a much more precise and efficient probing procedure than deflations, leveraging the specific knowledge acquired in past studies and during system analysis to design sounder experiments. The partitioning intervention reported in this chapter suggests that, similar to other previous re-evaluations, scattering-based systems decrease their performance on *GTZAN* when they cannot exploit the known faults of the collection. The decrease in performance for each feature representation is similar between SVM and BDT systems. This indicates that each evaluation condition contains distinct acoustic information that each feature representation captures to a different degree, driving performance changes across conditions to a larger extent than the learning algorithms. Both a principal components analysis and the performance of single-dimension BDT systems suggest feature dimensions corresponding to frequencies below 20 Hz as possible causes of such changes, motivating further analysis.

The results of the filtering intervention provide compelling evidence that *GTZAN* contain infrasonic information, inaudible to human listeners but linked to the classes of the collection, thus available for systems to exploit despite being arguably unrelated with the concept of music genre. Scattering-based systems, such as those Andén and Mallat (2014) propose, seem to rely on such infrasonic information to reproduce a large amount of the *GTZAN* ground truth. Although the infrasonic information appears to relate to the faults of the collection (e.g., recordings by the same artist likely contain similar infrasonic

patterns), introducing both filtering and partitioning interventions jointly reduces performance even further than each separately, which suggests that they affect distinct factors. Moreover, a simple BDT classifier using exclusively information below 20 Hz achieves virtually the same performance on *GTZAN* as the one originally reported by Tzanetakis and Cook (2002). All this evidence puts into question the results of classification experiments on *GTZAN* as valid estimates of the ability of systems to recognise music genre.

Attenuating the problematic frequencies before training does not seem to suffice to completely remove the impact of infrasonic patterns when systems are trained and tested under different acoustic conditions. More complex filtering methods would be necessary to break all possible correlations between the information at different bands and thus increase the likelihood of trained systems ignoring the patterns specific to the recordings in *GTZAN*. Alternative training procedures could involve mixing original and filtered recordings for training, or even augmenting the training collection using multiple copies of each recording exposed to a variety of subtle spectral manipulations.

Although training procedures that aim to minimise the impact of infrasonic information during training might lead to more generalisable systems, it is unlikely that systems trained on *GTZAN* are deployed in real-life scenarios. *GTZAN* is largely used as a benchmarking tool that permits comparisons with most of the literature devoted to the problem of music genre recognition. As Sturm (2013c) suggests, uncovering faults in a collection does not necessarily make it unsuitable for evaluation as long as such faults are properly leveraged. Some faults seem to affect differently systems constructed with different methods, so introducing evaluation conditions that account for such faults enable comparing their robustness. Moreover, understanding how and why a system works is essential to determine its suitability for a specific problem, not to mention its future improvement, and the faults of a collection appear particularly appropriate candidates for targeted analyses.

The interventions reported in this chapter follow a factorial structure, since they affect different factors that can be combined in a single joint experiment, such as the one in Table 4.5. The factors considered here are feature extractor, learning algorithm, partitioning condition and filtering condition, but the procedure could be extended further if more interventions were deemed necessary. Despite this sound structure, a major drawback of these experiments is that they disregard resampling, thus only yielding a single measure-

ment per factor combination. Moreover, due to the limitations of manual curated partitioning, systems evaluated under `RANDOM` and `CURATED` differ both in their training and testing materials. The following chapter introduces a systematic evaluation approach that addresses these issues. Analyses similar to those presented here serve as an exploratory first step that informs such a systematic approach, illuminating potentially confounding factors whose impact on the results of conventional evaluations using a specific collection can be assessed together with the benchmarking of various system-construction methods.

CHAPTER 5

CHARACTERISING CONFOUNDING EFFECTS IN MUSIC CLASSIFICATION EXPERIMENTS WITH INTERVENTIONS

Analyses such as those presented in the previous chapter highlight that systems might exploit extraneous cues from an evaluation collection to appear successful on a problem. This brings into question the relevance and validity of the results of classification experiments. The present chapter extends and systematises the intervention approach used in Ch. 4, proposing and illustrating a procedure to assess how failing to control for particular sources of information in evaluation collections affects experimental results for a wider range of system-construction methods. The work presented here largely mirrors what Rodríguez-Algarra et al. (2019) report.

Interventions introduced in the classification experiment pipeline create regulated evaluation conditions that can be used to characterise how the outcomes of music classification experiments are affected by "confounding", a validity threat we examine in Sec. 5.1. Sec. 5.2 introduces a procedure for combining multiple interventions that overcomes the limitations discussed in Sec. 4.4, including a novel resampling strategy aimed at gauging confounding effects. The approach described here focuses on the effects of particular sources of confounding information on test results, as this is paramount for MIREX[1] and

[1] http://www.music-ir.org/mirex/wiki/MIREX_HOME

similar evaluation exchanges. Sec. 5.3 illustrates the proposed procedure analysing two known confounders in the *GTZAN* music genre collection (Tzanetakis and Cook, 2002): artist replication and infrasonic content. Extending the proposed procedure would permit its use to assess effects in the training of systems, as well as its application to other domains. Sec. 5.4 discusses the main limitations and broader implications of the work reported here.

5.1 Confounding in Classification Experiments

Confounding as a validity threat involves the inability of an experimental design to disentangle the effects of different variables on the measurements, as mentioned in Sec. 2.5.1. A clear example of such a phenomenon in classification experiments occurs if each evaluated system predicts annotations on a different selection of instances. Subtler forms of confounding affecting the conclusions of classification experiments are receiving increasing attention in the applied Machine Learning literature (e.g., Charalambous and Bharath, 2016; Chen and Asch, 2017). In particular, information not intrinsically linked with the problem of interest might incidentally relate with the annotations of evaluation collections, providing alternative means for systems to predict annotations on classification experiments. Causes of this phenomenon include selection bias (e.g., Mendelson et al., 2017) and leakage (Kaufman et al., 2011), which induce confounding by conflating success in addressing the target problem — the outcome of interest — with the exploitation of auxiliary information — an extraneous influence (Sturm, 2016a). This chapter focuses on identifying and analysing the effects of these forms of confounding information.

If a collection is used for the evaluation of diverse problems and use cases, each case implicitly determines which content is potentially confounding. For instance, tempo information in a collection may be legitimate for identifying dance style, since the speed of a piece influences which dance moves are feasible, but not for identifying rhythmic patterns, since these should be invariant to reasonable variations in speed (Dixon et al., 2004; Sturm, 2014a). Furthermore, artists tend to compose or perform music pieces of one or a few genres, yet artist properties are not essential to those genres (Flexer and Schnitzer, 2010). If one's sole aim is to attach genre tags to a fixed set of recordings, artist information will likely help; if one aims to assess whether a system captures the defining characteris-

tics of music genres instead, then artist-specific content is extraneous. Other properties, such as the infrasonic content present in *GTZAN* that the experiments in Ch. 4 identified, are unlikely to be legitimately informative for most problems.

This interpretation of confounding as dependent on a target use case implicitly distinguishes two different goals for classification experiments. In the first case, the collection on which one conducts the experiments is the actual target — i.e., one wants to find the representation that more closely resembles the process that generated that specific set of instances. This is the goal of pure Machine Learning research, and much of Information Retrieval as well, since the collection is considered as given and fixed, or randomly sampled from a larger one of interest. The *task* that the algorithms are required to perform matches what they will be asked to do if deployed, with generalisation being limited to unseen instances from the same data generating process. Confounding as described above would likely not apply in this case, other than possibly in the form of leakage (Kaufman et al., 2011), since any information associated with the classes in the test instances is assumed to also appear in deployment. In the second case, however, the collection is a *proxy* for an underlying target problem, and not the target itself — i.e., one wants to find the best representation to capture the defining traits of a particular concept that extends beyond the instances in the collection. This is arguably the case in much MIR research, as discussed in Sec. 2.1. For instance, it seems unlikely that MGR research evaluated on *GTZAN* intends to develop systems meant to work only on *GTZAN*, but instead to build systems that are able to "recognise genre" both within and beyond that specific collection. Therefore, which information is potentially confounding in such scenarios depends entirely on the intended application of the evaluated systems and not necessarily on the proxy task on which performance is estimated.

The performance of systems that rely on information about a potential confounder being present are unlikely to generalise, since there is no guarantee that the observed association between such information and the problem will remain outside the experimental setting. The research community has adopted practices to counter this pitfall. Filtered partitioning, for instance, yields performance estimates free of the influence of the regulated potential confounder (Pampalk et al., 2005). Others suggest leveraging data augmentation to avoid confounding information influencing the training process (Char-

alambous and Bharath, 2016; Stowell et al., 2019). This synthetically generates combinations of background information and target categories that force systems to learn general concepts rather than incidental correlations. These countermeasures certainly benefit the generalisability of trained systems, but obscure evaluation feedback necessary to improve such systems in the future. Creating and comparing both regulated and unregulated conditions, on the other hand, illuminates the effects of potential confounders.

5.2 Characterising Confounding Effects

As Ch. 4 illustrates, comparing results obtained under regulated and unregulated evaluation conditions with regards to a potential confounder reveals whether systems exploit its confounding information. The partitioning intervention there relied on filtered partitioning to this end. Nevertheless, a major limitation of filtered partitioning is that the regulated training and testing collections it creates likely contain different instances than those included in their unregulated counterparts. No single trained system is thus exposed to both regulated and unregulated testing conditions, which impedes disentangling the effects of training and testing. Moreover, as Marques et al. (2011) note, the makeup of some collections constrains how many disjoint regulated partitions one can create (e.g., the number of Cross-Validation folds cannot exceed the number of artists per class). This may conflate the effect of the particular instances — their "difficulty" — with that of the (lack of) regulation. This section extends and systematises the analysis approach based on targeted interventions presented in Ch. 4 to gauge how confounding impacts the outcomes of classification experiments, overcoming the limitations of filtered partitioning via a novel partitioning strategy.

5.2.1 Interventions on the Experimental Pipeline

In empirical studies, an intervention is the act of explicitly fixing a factor to one of its levels (Pearl, 2009). A conventional music classification experiment involves intervening on the system creation method, as Fig. 2.2 represents with a double-bordered node. This specifies evaluation conditions to compare, each with different feature extraction and/or learning algorithms, yielding estimates of differences in performance. Apart from such

conventional intervention, one might also intervene on other steps of the pipeline to create further evaluation conditions. These may reveal information unavailable otherwise, such as the impact of a potential confounder.

Consider the train/test pipeline of a classification experiment, with training and testing materials drawn from a collection $\boldsymbol{C}$. Let z be a potential confounder. If z correlates with the classes in some way within $\boldsymbol{C}$, legitimately or not, then such correlation should appear in both training and testing instances unless a regulation is introduced, making z available for both training and prediction. Interventions regulating z thus impede its availability in such steps by breaking its correlation with the classes.

A classification experiment pipeline offers many opportunities for intervening. One might intervene on training or prediction, altering methods and systems to avoid relying on z. For instance, knowing which dimensions of the feature representations capture information related with z, one might regulate by removing or masking such dimensions in the feature extractor. This is the case in the tempo-invariant features of Dixon et al. (2004). Previous studies, however, often intervene on the creation of training and testing materials, through either "instance assignment" or "data manipulation" interventions.

Instance Assignment interventions regulate ψ, the criterion for assigning instances to either training ($\boldsymbol{C}_t$) or testing ($\boldsymbol{C}_p$), taking z into account. These interventions thus require knowledge of z, i.e., the value that z takes for each instance. Properties such as artist, album, file format, or recording device are suitable for this approach.

Filtered partitioning belongs to this category, with the intervention involving an assignment function $\psi'(\boldsymbol{C})$ that creates $\boldsymbol{C}'_t$ and $\boldsymbol{C}'_p$ both containing different instances than their unregulated counterparts. Other strategies may distinguish between regulated and unregulated conditions only for testing, using the exact same training materials in both (i.e., $\psi'(\boldsymbol{C}) = (\boldsymbol{C}_t, \boldsymbol{C}'_p)$). This enables isolating the potential effect of z in the evaluation of fixed systems. If one aims to estimate the impact of z in system construction instead, a suitable intervention might fix the testing collection and create regulated and unregulated conditions distinguished only in the selection of training instances (i.e., $\psi'(\boldsymbol{C}) = (\boldsymbol{C}'_t, \boldsymbol{C}_p)$).

Data Manipulation interventions alter the raw data (e.g., audio recordings) in a way that preserves their membership to a class, but modifies the correlation between z and the

classes. Manipulations such as pitch-preserving time-stretching (Sturm, 2016b) and high-pass filtering (Rodríguez-Algarra et al., 2016) have been used to this end. These interventions do not require instance-level knowledge of z, and permit comparing predictions on the same instances (manipulated and not). Nevertheless, they require identifying and implementing suitable manipulations.

Similar to instance assignment interventions, data manipulation interventions may create regulated conditions in different ways. Given a class-preserving manipulation, one might transform instances in both $\boldsymbol{C}_t$ and $\boldsymbol{C}_p$ in the same way, thus obtaining a pair of regulated collections $(\boldsymbol{C}'_t, \boldsymbol{C}'_p)$. This, however, may not break correlations if the manipulation is deterministic, failing to regulate z. It is more appropriate to keep either $\boldsymbol{C}_t$ or $\boldsymbol{C}_p$ unaltered and manipulate the other.

Different types of interventions are often complementary, since they affect different steps of the experimental pipeline, but it is feasible to stack various interventions affecting the same step (e.g., time-stretching and filtering recordings). They might be integrated into the experiment using a Factorial Design (Montgomery, 2013), where each intervention creates an additional treatment factor with at least two levels: regulated and unregulated. Comparing measurements under combinations of such levels reveals the marginal and joint impact of the interventions, illuminating the effects of the potential confounders.

5.2.2 Analysing Confounding with Interventions

To date, interventions on the experimental pipeline have been used to reveal whether a potential confounder affects the evaluation of particular methods or systems. This approach can be extended to assess how such a potential confounder impacts evaluations conducted on an annotated music collection $\boldsymbol{C}$ over multiple methods, and how several potential confounders interact, using the following steps.

a) Identify potential confounders

As a prerequisite of the analysis, one should determine which potential confounders are worth considering for the collection and problem at hand. This may come from exploratory analyses of collections, published systems and/or domain knowledge.

b) Design interventions

For each identified potential confounder z, one should specify at least one suitable intervention to distinguish regulated and unregulated evaluation conditions with respect to z. The appropriate type of intervention depends on the nature of z.

c) Create train/test materials

Let $\boldsymbol{C}_t$ be a training collection drawn from $\boldsymbol{C}$, and $\boldsymbol{C}_p$ and $\boldsymbol{C}'_p$ a pair of testing collections associated with $\boldsymbol{C}_t$ that differ only in whether they regulate a potential confounder z. In particular, $\boldsymbol{C}_p$ is drawn from $\boldsymbol{C}$ (usually $\boldsymbol{C} \setminus \boldsymbol{C}_t$), and $\boldsymbol{C}'_p$ comes from an intervention on the experimental pipeline. For instance, $\boldsymbol{C}'_p$ might be a pruned version of $\boldsymbol{C}_p$ with instances whose value of z appears in $\boldsymbol{C}_t$ removed, or the result of a manipulation on the recordings in $\boldsymbol{C}_p$ for regulating z. If the analysis considers J interventions simultaneously, then one creates (at least) 2^J testing collections associated with $\boldsymbol{C}_t$, one for each combination of regulation condition.

To avoid the performance estimates being confounded with the selection of instances, it is advisable to create multiple training collections through a resampling strategy (Weihs et al., 2017). In this case, one would draw K training collections $\boldsymbol{C}_{t,k}$ and derive the testing collections associated with each as above. Conventional resampling strategies, however, cannot ensure testing collections from instance assignment interventions fulfil the intended regulation. The strategy we propose later in Sec. 5.2.3 addresses this issue.

d) Select methods

Characterising the impact of a potential confounder z requires a wide range of performance estimates. One may then train multiple systems on each $\boldsymbol{C}_{t,k}$ using diverse combinations of feature extraction and learning algorithms, for a total of M combined methods. These methods should cover a broad spectrum of modelling approaches and expected performance values. Optimisation is not essential if the goal is to gauge how different approaches behave when exposed to particular perturbations on the data and not to maximise performance, but plays an important role if one aims to identify the most suitable systems for deployment.

e) Obtain performance estimates

For each trained system s_j, $1 \leq j \leq K \cdot M$, one can then compute figures of merit (e.g., ac-

curacy, mean recall) in the corresponding testing collections. $\hat{y}$ and $\hat{y}'$ refer to the generic unregulated and regulated performance estimates, respectively.

f) Relate regulated and unregulated estimates

Since $\hat{y}$ and $\hat{y}'$ differ only in their regulation of z, one assumes any observed difference reflects an effect of z. Given enough $(\hat{y}, \hat{y}')$ pairs, one might estimate the expected relationship between regulated and unregulated measurements $\hat{y}' \sim f(\hat{y})$.[2] Fitting a model of $f(\hat{y})$ from data pairs $(\hat{y}, \hat{y}')$ describes the *confounding effect* of z in evaluations using $\boldsymbol{C}$. This reflects how a potential confounder tends to affect performance estimates of trained systems evaluated in the collection. For simplicity, one may use a linear model, such as

$$\hat{y}' \sim f(\hat{y}) = \alpha \cdot \hat{y} + \kappa \tag{5.1}$$

though other relationships (e.g., quadratic, exponential) could be preferable. If $\alpha \approx 1$ and $|\kappa| \gg 0$, the confounding effect of z is mostly additive (i.e., the relationship between $\hat{y}$ and $\hat{y}'$ appears as a fixed effect); if $\alpha \not\approx 1$ and $\kappa \approx 0$, it is mostly multiplicative (i.e., a gain). To estimate κ in the former case, one could average performance differences between conditions per iteration. Denote $\hat{y}_{m,k}$ the performance of a system trained with $\boldsymbol{C}_{t,k}$ using method m measured on a test collection $\boldsymbol{C}_{p,k}$, and $\hat{y}'_{m,k}$ the measurement on the associated regulated test collection $\boldsymbol{C}'_{p,k}$, then:

$$\hat{\kappa} = \frac{\sum_{k=1}^{K} \sum_{\forall m} (\hat{y}_{m,k} - \hat{y}'_{m,k})}{K \cdot M} \tag{5.2}$$

with K and M defined as above.

In the general case, $\hat{y}$ and $\hat{y}'$ will not keep a simple relationship over all observations. Different system-construction methods can exploit a potential confounder differently, and the effect might also differ across classes. One may thus analyse the data marginally to identify clearly distinct behaviours.

If the analysis involves multiple interventions, comparing marginal and joint measurements can elucidate whether the different confounders (or approaches to the same

[2] The symbol ~ indicates "modelled as".

confounder) interact. Let $\hat{y}$ be the performance estimated in the original testing collection, $\hat{y}'_1$ and $\hat{y}'_2$ the performances in testing collections from two different interventions, and $\hat{y}'_{1,2}$ the performance on a testing collection subjected to both interventions. Apart from relating $\hat{y}$ with both $\hat{y}'_1$ and $\hat{y}'_2$ to analyse the effects of each confounder separately, one might compare the sum of those two marginal effects with the difference between $\hat{y}$ and $\hat{y}'_{1,2}$. Let Δ_A be the "accumulated" variation, defined as:

$$\Delta_A = (\hat{y} - \hat{y}'_1) + (\hat{y} - \hat{y}'_2) \tag{5.3}$$

and Δ_R be the "real" variation:

$$\Delta_R = (\hat{y} - \hat{y}'_{1,2}). \tag{5.4}$$

The difference $\Delta_R - \Delta_A$ indicates whether the two confounding effects under study reinforce each other, do not interact, or overlap. This can be generalised to higher-order interactions if more interventions coexist.

5.2.3 Regulated Bootstrap Resampling

The procedure above requires multiple distinct train/test pairs. Various resampling strategies address this, but none is entirely suitable for instance assignment interventions. In particular, the fixed size of the partitions in K-fold Cross-Validation (K-CV) impedes adjusting to imbalances in the presence of the potential confounder z. Bootstrap sampling (Efron, 1977), drawing $|\boldsymbol{C}|$ training instances with replacement from the whole collection $\boldsymbol{C}$, overcomes this issue. Sampling with replacement is often preferred in the statistical learning literature (Hastie et al., 2009; Hothorn et al., 2005), since it enhances the statistical properties of the generated samples over K-CV, such as reducing the variance of the derived estimates (Efron, 1983; Efron and Tibshirani, 1997). Nevertheless, training collections generated with bootstrap sampling may not permit suitable regulations if, e.g., too many instances in $\boldsymbol{C}_p = \boldsymbol{C} \setminus \boldsymbol{C}_t$ have values of z also in $\boldsymbol{C}_t$.

Regulated bootstrap, a novel multi-phase resampling strategy expressed in Alg. 1, addresses the limitations of conventional strategies for instance assignment interventions. The algorithm takes as input a collection $\boldsymbol{C}$ (sequence of instances, each a tuple $(r, a, z)_i$

Algorithm 1 Regulated Bootstrap resampling strategy, given a collection $\boldsymbol{C}$ and a threshold $n_r \in \mathbb{N}$.

RegulatedBootstrap($\boldsymbol{C}$, n_r):
- Initialise: $\boldsymbol{C}_t \leftarrow (\emptyset)$, $\boldsymbol{C}_p \leftarrow (\emptyset)$
- For each $\mathtt{a} \in \boldsymbol{A}$:
  0. Define $\boldsymbol{C}_\mathtt{a}$ as the instances in $\boldsymbol{C}$ with $a_i = \mathtt{a}$;
  1. Phase 1: Stratified Bootstrap Sampling
     a) Create c_t by uniformly sampling with replacement $|\boldsymbol{C}_\mathtt{a}|$ instances from $\boldsymbol{C}_\mathtt{a}$;
     b) Create $c_p \leftarrow \boldsymbol{C}_\mathtt{a} \setminus c_t$;
  2. Phase 2: Size Verification
     a) Define Z_t as the union of all z_i in c_t;
     b) Create c'_p by selecting all instances $(r, a, z)_i$ in c_p with z_i not in Z_t;
     c) If $|c'_p| < n_r$, proceed to Phase 3, as it lacks enough regulated instances; otherwise, go to Phase 4;
  3. Phase 3: Curated Sampling
     a) Define $Z_\mathtt{a}$ as the union of all z_i in $\boldsymbol{C}_\mathtt{a}$;
     b) Initialise a hold-out collection $c_h \leftarrow (\emptyset)$;
     c) Randomly select a $\mathtt{z} \in Z_\mathtt{a}$, and remove it from $Z_\mathtt{a}$;
     d) Define $c_\mathtt{z}$ as the instances in $\boldsymbol{C}_\mathtt{a}$ with $\mathtt{z} \in z_i$;
     e) Append $c_\mathtt{z}$ to c_h: $c_h \leftarrow c_h{}^\frown c_\mathtt{z}$;
     f) If $|c_h| < n_r$, go to (3c), as c_h still lacks enough instances;
     g) Create c_t by uniformly sampling with replacement $|\boldsymbol{C}_\mathtt{a}|$ instances from $\boldsymbol{C}_\mathtt{a} \setminus c_h$;
     h) Create $c_p \leftarrow \boldsymbol{C}_\mathtt{a} \setminus c_t$;
     i) Go to Phase 2 to check size requirements;
  4. Phase 4: Concatenation
     a) Append c_t to $\boldsymbol{C}_t$: $\boldsymbol{C}_t \leftarrow \boldsymbol{C}_t{}^\frown c_t$;
     b) Append c_p to $\boldsymbol{C}_p$: $\boldsymbol{C}_p \leftarrow \boldsymbol{C}_p{}^\frown c_p$;
- Return: train/test pair $(\boldsymbol{C}_t, \boldsymbol{C}_p)$

of data element r_i, class annotation a_i from the set $\boldsymbol{A}$, and attribute z_i from the set Z) and the desired number of recordings per class n_r. It first attempts to create a pair $(\boldsymbol{C}_t, \boldsymbol{C}_p)$ using stratified bootstrap — sampling with replacement from each class separately. If this cannot derive a regulated testing collection $\boldsymbol{C}'_p$ of size n_r, it then proceeds to a partially-curated approach. This may be repeated an arbitrary number of times. The output of each sampling run can then be used to generate a $\boldsymbol{C}'_p$ through pruning: removing all instances in $\boldsymbol{C}_p$ with z also in $\boldsymbol{C}_t$. Although the pruned instances do not appear in $\boldsymbol{C}'_p$, they cannot be added to $\boldsymbol{C}_t$ since they remain in $\boldsymbol{C}_p$.

As an illustration of the regulated bootstrap algorithm, consider a collection $\boldsymbol{C}$ with 5 instances per class, aiming to create a pair $(\boldsymbol{C}_t, \boldsymbol{C}_p)$ from which to derive a $\boldsymbol{C}'_p$ with $n_r = 2$ instances per class. For a given class $\mathtt{a} \in \boldsymbol{A}$, assume $\boldsymbol{C}$ contains $\{(r_1, \mathtt{a}, \{\text{“E”}\})$, $(r_2, \mathtt{a}, \{\text{“E”}, \text{“F”}\})$, $(r_3, \mathtt{a}, \{\text{“F”}\})$, $(r_4, \mathtt{a}, \{\text{“G”}\})$, $(r_5, \mathtt{a}, \{\text{“H”}\})\}$. The set of values of z is thus $Z = \{\text{“E”}, \text{“F”}, \text{“G”}, \text{“H”}\}$, with some instances sharing values of z and one that takes two.

For simplicity, each instance is referred hereinafter by its index (i.e., 1, 2, 3, 4, 5).

i) Assume the bootstrap sampling in Phase 1 creates the pair $c_t = \{1,3,3,4,5\}$, $c_p = \{2\}$. Since the size of the test collection is smaller than n_r, failing the check in Phase 2, the procedure continues with the curated sampling in Phase 3.

ii) Assume $\mathtt{z}$ = "F" is drawn and thus instances $c_h = \{2,3\}$ are held out. Since the size of the hold-out collection matches n_r, we create a train/test pair avoiding instances in c_h for training, such as $c_t = \{1,1,4,5,5\}$, $c_p = \{2,3\}$.

iii) Even though c_p contains 2 instances, only one of them (3) is regulated, since the other (2) shares value of z ("E") with an instance in c_t (1). Therefore, the check in Phase 2 fails, so the curated sampling in Phase 3 is attempted again.

iv) Assume now $\mathtt{z}$ = "G" is drawn, thus holding out instance 4 ($c_h = \{4\}$). As the hold-out collection includes only one instance, further values are then drawn from Z.

v) Assume $\mathtt{z}$ = "H" is drawn and thus instance 5 is appended to the hold-out collection ($c_h = \{4,5\}$).

vi) Since $|c_h| = n_r = 2$, a new train/test pair is created avoiding instances in c_h for training, such as $c_t = \{1,1,3,3,3\}$, $c_p = \{2,4,5\}$.

vii) The number of regulated instances in the testing collection now meets the threshold, so the pair is accepted, finally proceeding to Phase 4.

One would repeat these steps for all other classes in the collection to create $\boldsymbol{C}_t$ and $\boldsymbol{C}_p$, and then $\boldsymbol{C}'_p$ through pruning.

Some aspects of the algorithm deserve clarification. First, it does not immediately accept the pair generated after Step (3h), since instances might relate with more than one value of z (e.g., a song might be a collaboration between two artists), making different $c_{\mathtt{z}}$ overlap. In that case, the number of unique elements of d_h might fall short of the specified minimum, requiring multiple attempts until finally succeeding. Second, the algorithm does not impose any restriction regarding the same value of z appearing across different classes to avoid benefiting systems exploiting z. For instance, if a system relied on artist-specific cues to predict class annotations, its estimated performance would benefit from removing recordings from a particular artist from the pruned test collection who appears in a different class on training recordings, since the system would tend to mislabel those in test. Finally, the sampling is performed at instance level to favour scalability of the

algorithm, allowing future regulations over multiple z simultaneously.

Although class-wise computations ensure stratification in the training collections, the associated testing collections will likely be imbalanced and of different size across iterations. Moreover, pruning causes regulated and unregulated testing collections to differ in size. If these issues raise reliability concerns, it might prove useful to randomly prune test collections under both conditions to a fixed size per class, such as n_r or a larger value if suitable. The choice of n_r depends on the context, but aiming at a number of regulated instances at least equal to the size of a fold in 10-CV might be a good rule of thumb, both overcoming these issues and avoiding sample size concerns. In case n_r is too high and it becomes impossible to create $\boldsymbol{C}'_p$, it is trivial to include an exit condition in the algorithm. Along with collecting instance-level information about z, if missing, only the choice of n_r requires human involvement in this otherwise automated resampling strategy.

5.3 Application to *GTZAN*

The study reported in this section illustrates the analysis procedure proposed in Sec. 5.2, applying it to investigate the confounding effects of artist replication and infrasonic content in classification experiments involving the *GTZAN* music genre collection (Tzanetakis and Cook, 2002). The presence of multiple known confounders that can be regulated using different intervention types makes this collection ideal to showcase the factorial analysis approach proposed. In particular, the recent identification of most excerpts in the collection reveals that a few artists dominate most classes, as Fig. 2.3 shows. The highly imbalanced artist distributions impede the use of conventional resampling strategies to create multiple train/test pairs regulated by artist; the regulated bootstrap algorithm overcomes this issue. Moreover, the infrasonic content uncovered in Ch. 4 permits showcasing how interventions of different type can be integrated and used to analyse the interaction between potential confounders. To this end, the case study here trains and tests systems built using multiple feature extraction and learning algorithms under evaluation conditions derived from both instance assignment and data manipulation interventions. The code is available online.[3]

[3] `https://code.soundsoftware.ac.uk/projects/confint`

Class	% Samples
blues	99.99 %
classical	98.62 %
country	01.93 %
disco	00.11 %
hiphop	79.19 %
jazz	99.22 %
metal	71.40 %
pop	98.31 %
reggae	98.11 %
rock	99.70 %

Table 5.1: Estimated percentage of train/test samples requiring curated sampling for each *GTZAN* class if drawn using Alg. 1 to regulate over artists, from 100,000 simulations with $n_r = 10$.

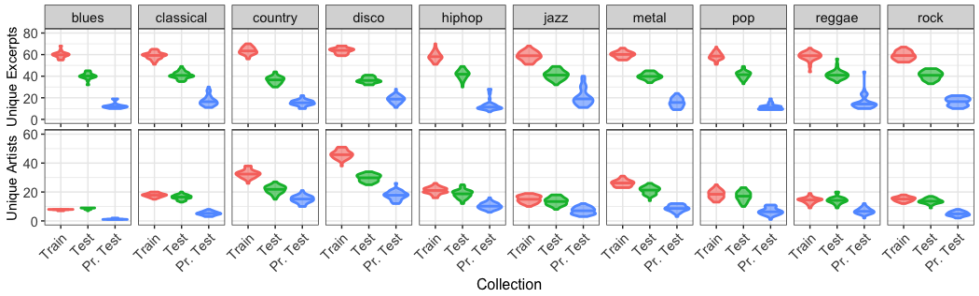


Figure 5.1: Distribution of the number of unique excerpts (Top) and artists (Bottom) per class in the training and testing collections sampled from *GTZAN* using bootstrap regulated over artists.

5.3.1 Evaluation Conditions

A total of $K = 40$ training and testing collection pairs are drawn from *GTZAN* using the regulated bootstrap resampling strategy described in Sec. 5.2.3, with $n_r = 10$. This ensures that at least 10 recordings per *GTZAN* class in each testing collection feature no artist that appears in its corresponding training collection. Table 5.1 includes estimates of the proportion of train/test samples that require curated sampling to achieve this for each *GTZAN* class. Consistent with the distributions in Fig. 2.3, the artist imbalance in some classes demands curation to ensure proper artist separation.

Figure 5.1 shows the distribution of the number of unique excerpts per class across iterations. Although all training collections contain exactly 100 excerpts per class, some of them are repeated. The expected number of unique instances in a bootstrap sample drawn from 100 elements is 63.2 (Efron and Tibshirani, 1997), approximately what Fig. 5.1

(Top) shows for the training collections despite the curation. The size of the testing collections (with and without pruning) matches their number of unique excerpts, as they contain no duplicates. Figure 5.1 (Top) also shows that training collections generally include more unique excerpts than their corresponding testing collections. Some outliers in `reggae` appear to be the exception due to the large proportion of Bob Marley recordings. Figure 5.1 (Bottom) highlights the expected decrease in artist variety after pruning. As suggested by Fig. 2.3, `blues` suffers from the lowest variety among all collections.

Every recording in *GTZAN* is also manipulated similarly to the audio filtering intervention by Rodríguez-Algarra et al. (2016) described in Ch. 4, but in this case using a high-pass Infinite Impulse Response (IIR) filterbank, with stop-band frequency at 19 Hz, pass-band frequency at 20 Hz, 60 dB attenuation in the stop-band, and maximum 1 dB ripple allowed in the pass-band. Combining which recordings are included in the collections with their audio filtering status defines six distinct evaluation conditions for each iteration. These conditions are hereinafter referred to as `train`, `test`, and `pr. test`, which stands for "pruned test", appending "`(filt.)`" to their name (e.g., `train (filt.)`) when their recordings have been high-pass filtered.

5.3.2 Feature Extraction and Learning Algorithms

Multiple prediction systems are built using various combinations of feature representations and learning algorithms. The learning algorithms employed cover a wide range of supervised learning approaches, both parametric and non-parametric. In particular, the analysis involves `scikit-learn`[4] implementations of: Naive Bayes (`NB`), 1- and 5-Nearest Neighbours (`1-NN` and `5-NN`), Decision Trees with and without AdaBoost (`ABDT` and `DT`), Random Forests (`RF`), Support Vector Machines (`SVM`), and Multi-layer Perceptrons (`MLP`). These use out-of-the-box implementations and avoid hyperparameter tuning, since the goal is not to maximise performance but gauge how confounding affects measurements for a wide range of modelling approaches and performance values, including those at the lower end of the performance axis that are only feasible from suboptimal systems. Ignoring tuning here permits increasing the number of methods and resampling iterations considered without requiring an excessive amount of time and computational resources.

[4] `http://scikit-learn.org/stable/`

Nevertheless, this means that the performances reported should not be taken as representative of the potential of each method.

The multiple feature representations selected focus on different aspects of the audio signals and come from two sources: the `Essentia` music extractor (Bogdanov et al., 2013) and the scattering-based audio features by Andén and Mallat (2014). The features extracted from `Essentia` are grouped into 8 disjoint sets for the analyses here — `Rhythm`, `Tonal`, `Tim+Dyn` (i.e., timbre plus dynamics), `MFCC`, `GFCC`, `Barkbands`, `Melbands`, and `Erbbands` —, referred jointly as `non-scattering` features hereinafter. Regarding the scattering-based features, Mel-scaled (`Mel Sc.`), first-layer (`1-L Sc.`), and joint first- and second-layer time-scattering features (`1&2-L Sc.`) are computed. Unlike `non-scattering` features, these express frame-level information, so excerpt-level summary statistics of first-layer time-scattering features (`Des. 1-L Sc.`) are also included.

5.3.3 Instance Assignment: Artist Information

The first analysis conducted compares measurements obtained on `test` and `pr. test` to assess the effect of artist replication. Other than size, these conditions differ only in whether their artist content is regulated. Systems are trained using every combination of the selected feature extractors and learning algorithms on each of the K training collections drawn, yielding $40 \times 12 \times 8 = 3840$ distinct systems. Figure 5.2 shows performance statistics across iterations, using mean recall as metric to compensate for class imbalances derived from the resampling strategy employed. The performances are systematically lower on `pr. test` than on `test`, agreeing with results that Sturm (2014d) reports. Systems thus tend to decrease their performance when the artist-specific cues of *GTZAN* are unavailable.

Since the pruning process that creates `pr. test` collections from their `test` counterparts also changes their size, it might be argued that size differences and not the regulation drive the performance drops that Fig. 5.2 show. Simulations suggest this is not the case. Only 12.8% of all measurements in `pr. test` are equal or superior to their counterpart in `test`. If size explained differences in performance, one would expect that percentage to be similar when computed on any subset of instances from `test` of the same size as `pr. test`. On 100 simulations using randomly generated subsets of `test` with identical

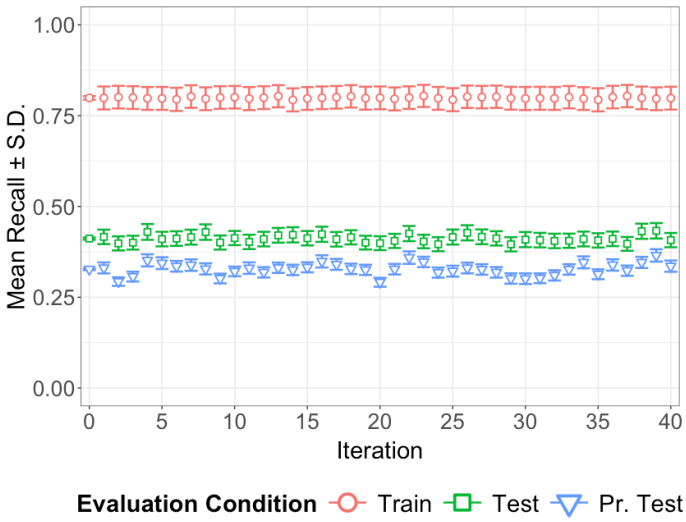


Figure 5.2: Mean recall (± standard deviation) on `train`, `test`, and `pr. test` for each regulated bootstrap iteration over all combinations of feature extraction and learning algorithms on original *GTZAN* recordings. Position 0 represents the mean recall over all iterations.

class sizes as `pr. test`, an average of 53.7% (± 2.3) of the performance measurements on the subsets are equal or superior to their `test` counterpart. Moreover, 15.6% (± 0.5) of measurements in `pr. test` are equal or superior to their counterpart in the simulations, compared to an average of 54.4% (± 2.3) between simulations (see Fig. 5.3) — i.e., performance is lower on a regulated subset than any other of the same size substantially more often than between two randomly generated ones. These observations suggest the regulation, and not the size differences, explain the differences in performance between `test` and `pr. test` evaluation conditions.

An estimate of κ according to Eq. (5.2) yields a decrease in mean recall of approximately $\hat{\kappa} \approx 0.085$ (8.5 percentage points). A closer look at the measurements reveals the naivety of this approach. Figure 5.4 shows that, despite consistently lower results on `pr. test` than `test`, the distribution of the performance measurements varies widely when marginalised over class, feature set or learning algorithm. This suggests the confounding effect of artist replication in *GTZAN* does not impact performance measurements as an additive fixed effect, i.e., that there exist interactions between that metric and the classes, features, and learning algorithms.

As the Top row of Fig. 5.4 shows, the differences in performance distribution between

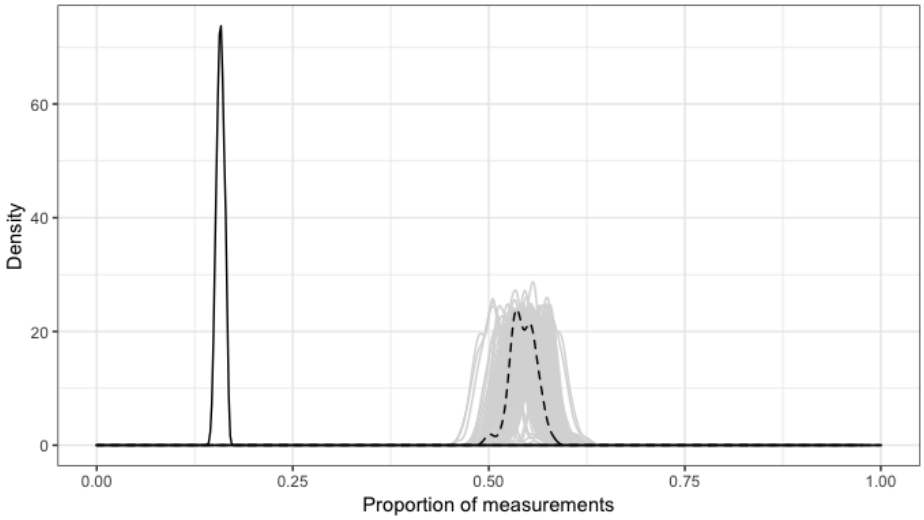


Figure 5.3: Proportion of mean recall measurements equal or superior to their corresponding one calculated on a different subset of `test` with identical number of instances per class. The solid black line represents the measurements on `pr. test` (i.e., regulated for artist replication) compared with measurements on 100 simulated subsets (mean 0.1586, standard deviation 0.0046); the grey lines represent the 100×100 pairwise comparisons between the measurements on simulated subsets, with the dashed black line representing their average (mean 0.544, standard deviation 0.0228). Each simulation involves a sampled set of instances for each of the 40 test collections drawn from *GTZAN* using regulated bootstrap.

`test` and `pr. test` vary across *GTZAN* classes. The largest difference by far occurs on `blues` recordings, with an average drop due to regulation of 19 percentage points — a relative decrease of more than 53%. This behaviour might be expected, since `blues` is the *GTZAN* class with the least artist variety. Similarly, the average recall on `reggae` recordings drops 9.7 percentage points with regulation (almost 30% relative decrease), which may relate to one artist dominating the class. The relative decrease on `pop` recordings is even higher (32.4%), and might arise from duplicate recordings in that class (Sturm, 2014d).

At the other end of the spectrum, `metal`, `classical` and `disco` suffer average relative decreases in recall below 10% (7.7%, 8.1%, and 9.6%, respectively). Fig. 2.3 shows `disco` is the class in *GTZAN* with largest artist variety. Despite having less than half the number of unique artists, however, `metal` and `classical` not only suffer the smallest relative average decrease, but also yield the highest average on both `test` and `pr. test`. This suggests these classes are so different from others in *GTZAN* that they are distinguished even without artist-specific information.

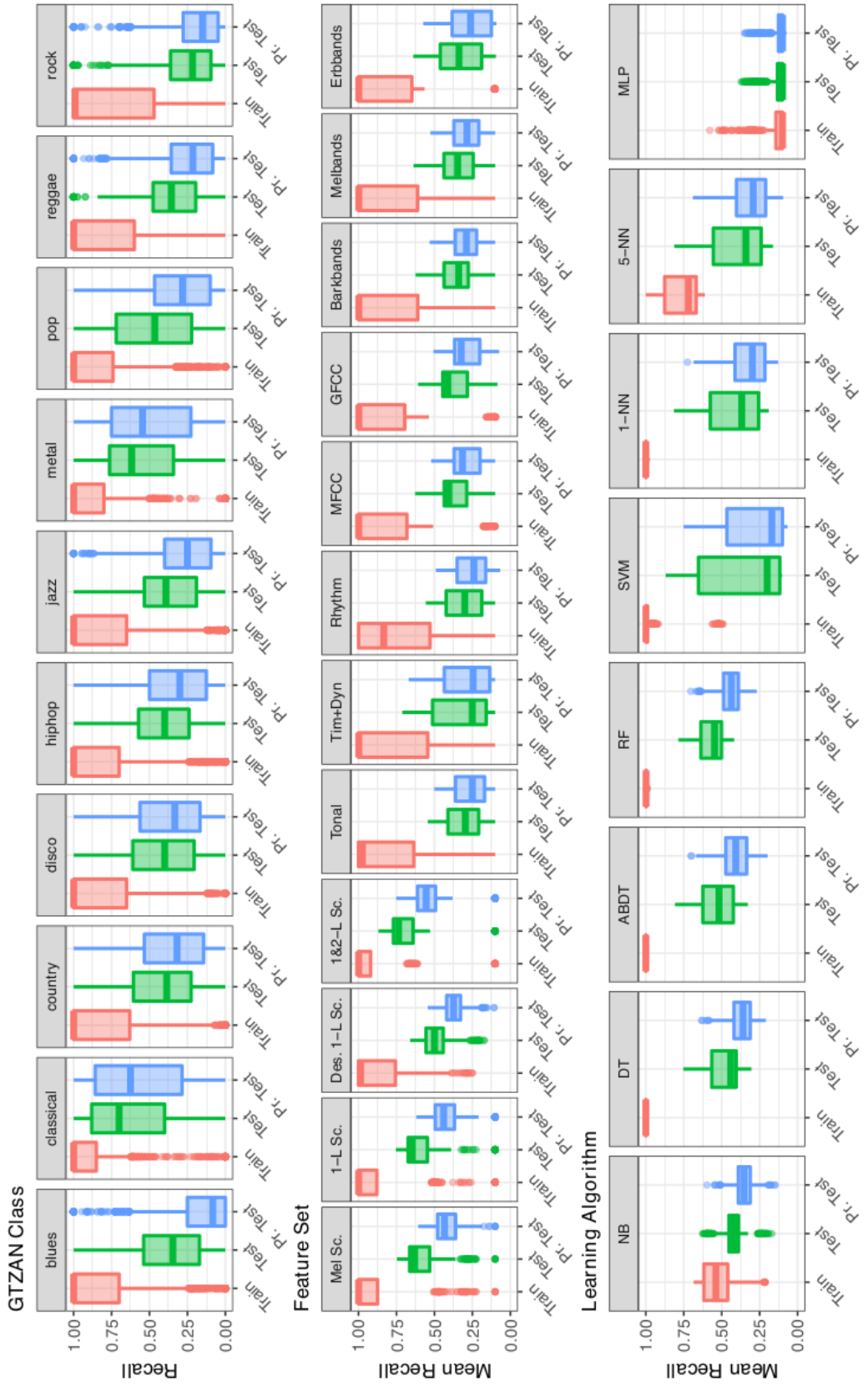


Figure 5.4: Quartiles of (mean) recall distribution obtained on `train`, `test`, and `pr. test`, marginalised over *GTZAN* class (Top), feature set (Middle), and learning algorithm (Bottom).

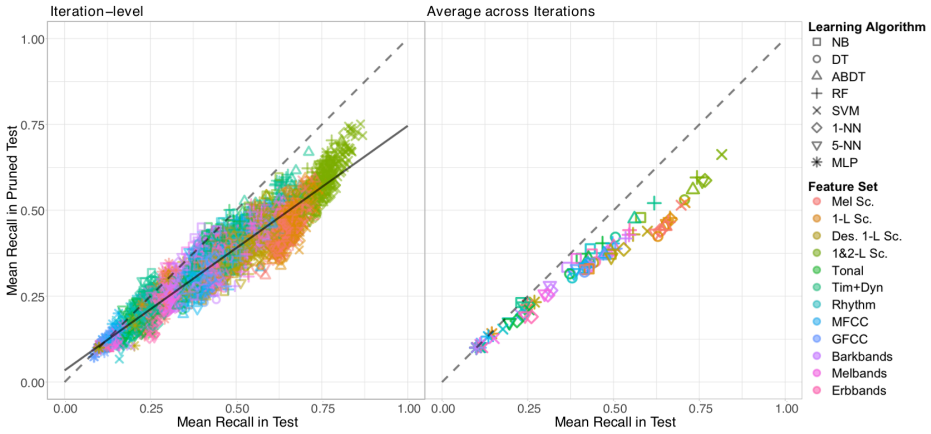


Figure 5.5: Relationship between mean recall on `test` and `pr. test` obtained by systems constructed with different combinations of feature representations and learning algorithms on training collections sampled from *GTZAN* with bootstrap regulated over artists, represented both as individual values for each system (Left) and averages across iterations (Right). The dashed line indicates the case of equal mean recall on `test` and `pr. test`; the solid line indicates the linear regression model fitting the data as in Eq. (5.1).

Marginalising over feature extraction method, Fig. 5.4 shows systems using scattering-based features tend to obtain higher performances than `non-scattering`, both on `test` and `pr. test`. Overall, differences in mean recall between `test` and `pr. test` are highest for both `Mel Sc.` and `1-L Sc.` features, with a decrease of approximately 15.8 percentage points in both — a decrease of 27.7% from `test`. The lowest drop, both in absolute and relative terms, occurs for `Tim+Dyn` systems (4 percentage points, 12% decrease from `test`).

Marginalising over learning algorithm also reveals clear differences in performance distribution. Systems constructed using the suboptimal `MLP` architecture tend to perform close to the random baseline of 0.1 mean recall. For every single learning algorithm, including `MLP`, performance tends to decrease between `train` and `test`, and between `test` and `pr. test`. Apart from `MLP`, `NB` is the only other algorithm that shows an average relative difference in mean recall between `test` and `pr. test` below 20%. It is also the algorithm that seems to suffer the least from overfitting. Despite a far lower performance on `train`, `NB` systems perform on average equivalently to `1-NN` systems on `test`, and slightly superior on `pr. test`, with substantially lower variance in both cases. Systems from all other algorithms decrease on average around 20.5% to 23.5% between `test` and `pr. test`, with `DT` having the largest drop.

Figure 5.5 relates the performance trained systems achieve on `test` to that on `pr. test`, both individually (left) and grouped by feature representation and learning algorithm (right). A linear fit gives a slope $\hat{\alpha} = 0.712 \pm 0.003$ and an intercept $\hat{\kappa} = 0.034 \pm 0.001$ ($R^2 = 0.929$). The slope is thus lower than the case of no confounding, represented with a dashed line in Fig. 5.5. This suggests regulating by artist on *GTZAN* attenuates the estimated performance to around 70% of its unregulated value. This equates to considering the confounding effect of artist replication in *GTZAN* as a gain in mean recall of approximately $1/0.712 \approx 1.4$.

The data points at the higher end of performance measurements in Fig. 5.5 deviate from the estimated regression line. This may suggest using more complex models, but exponential and polynomial models up to third degree do not substantially improve the fit. A model including both third degree polynomial and exponential terms increases R^2 to 0.932, but at the cost of hard to interpret coefficients and the risk of overfitting.

5.3.4 Data Manipulation: Infrasonic Content

The analysis reported in Ch. 4 suggests that previously unknown infrasonic content in *GTZAN* recordings affects performance estimates of scattering-based `SVM` systems (Rodríguez-Algarra et al., 2016). The results reported here extend such analysis to include non-scattering feature representations and a wider range of learning algorithms to gauge the extent of that effect. The performance measurements are obtained from the same systems in Sec. 5.3.3 and compared under `test` and `test (filt.)` evaluation conditions, which differ exclusively in sub-20 Hz content. Overall, the average decrease in mean recall between these two conditions calculated as in Eq. (5.2) is $\hat{\kappa} \approx 0.098$, slightly larger than the one observed for artist replication.

Figure 5.6 shows the observed performances, marginalised by *GTZAN* class, feature representation, and learning algorithm. The figure includes measurements on the training recordings and their filtered equivalents, revealing that performance estimates decrease between `train` and `train (filt.)` across system-construction methods and classes. Overall, the average decrease in mean recall between these two conditions is of 28 percentage points. Regardless of whether they exploit class-specific patterns of infrasonic content to predict annotations in unseen instances, systems trained on *GTZAN* seem to

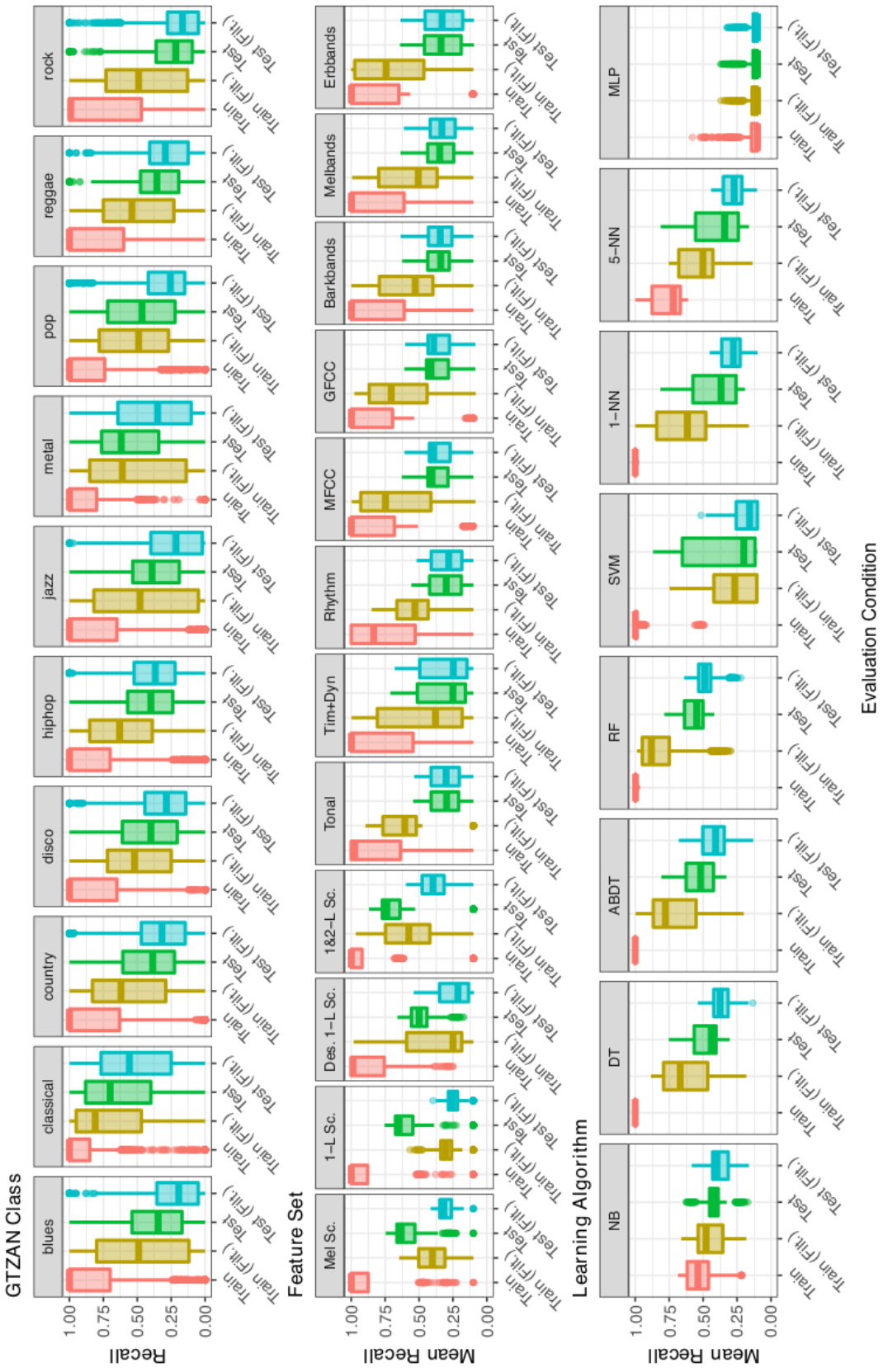


Figure 5.6: Quartiles of (mean) recall distribution obtained on `train`, `train (filt.)`, `test`, and `test (filt.)`, marginalised over *GTZAN* class (Top), feature set (Middle), and learning algorithm (Bottom). Note that the colours in this figure not matching those in Figs. 5.1, 5.2 and 5.4 correspond to different evaluation conditions.

rely often on such content (or related information, such as the overall energy level) to identify recordings previously seen during training and predict their class.

The *GTZAN* class with largest relative average decrease in recall between `test` and `test (filt.)` is `jazz`, with 37.2%, followed by `pop`, the largest drop in absolute terms, and `blues`, with 34.9% and 33.7%, respectively. The smallest decrease by far occurs for `hiphop` recordings, with an average 5.5% relative decrease. The closest classes are `reggae` and `classical`, both with over 16.5% relative decrease on average. Some might speculate these reductions in performance originate from removing information legitimately characteristic of some music genres, such as sub-bass kick drums in Hip-Hop recordings. Seeing how measurements in *GTZAN*'s `hiphop` are barely affected by the intervention compared to other classes that should not present any pattern at those frequencies (such as `jazz`), seems to disprove this explanation.

Marginal analysis of measurements by feature representation reveals two clearly distinct behaviours, and suggests models such as Eqn (5.1) might not apply in this case. The mean recall of scattering-based systems decreases on average between 41% (`1&2-L Sc.`) and 57% (`1-L Sc.`) when comparing `test` and `test (filt.)`. On the other hand, no average decrease of `non-scattering` features exceeds 4%, one order of magnitude lower. This brings the average performance of all scattering-based systems except those using `1&2-L Sc.` to the bottom of the list on `test (filt.)`, despite appearing substantially more successful than any `non-scattering` feature set on `test`. Feature representations such as `MFCC` discard infrasonic information, with all filters centred at frequencies above the human hearing threshold (Davis and Mermelstein, 1980). Scattering-based features, even those supposedly Mel-scaled, have multiple filters centred below 20 Hz, as it was described in Sec. 4.1.1. Figure 5.7 shows the distinct behaviour of each group, where measurements from systems using `non-scattering` feature representations follow quite closely the ideal behaviour indicated by the dashed line, whereas those from scattering-based systems tend to create clusters away from that line.

Among the considered learning algorithms, `SVM` is the one with largest drop in performance between `test` and `test (filt.)` — an average decrease of 42.6% in mean recall. Other than `MLP`, `NB` is the algorithm that suffers the lowest average decrease (10.5%), with the remaining algorithms decreasing between 16.7% and 31.7% mean recall on average.

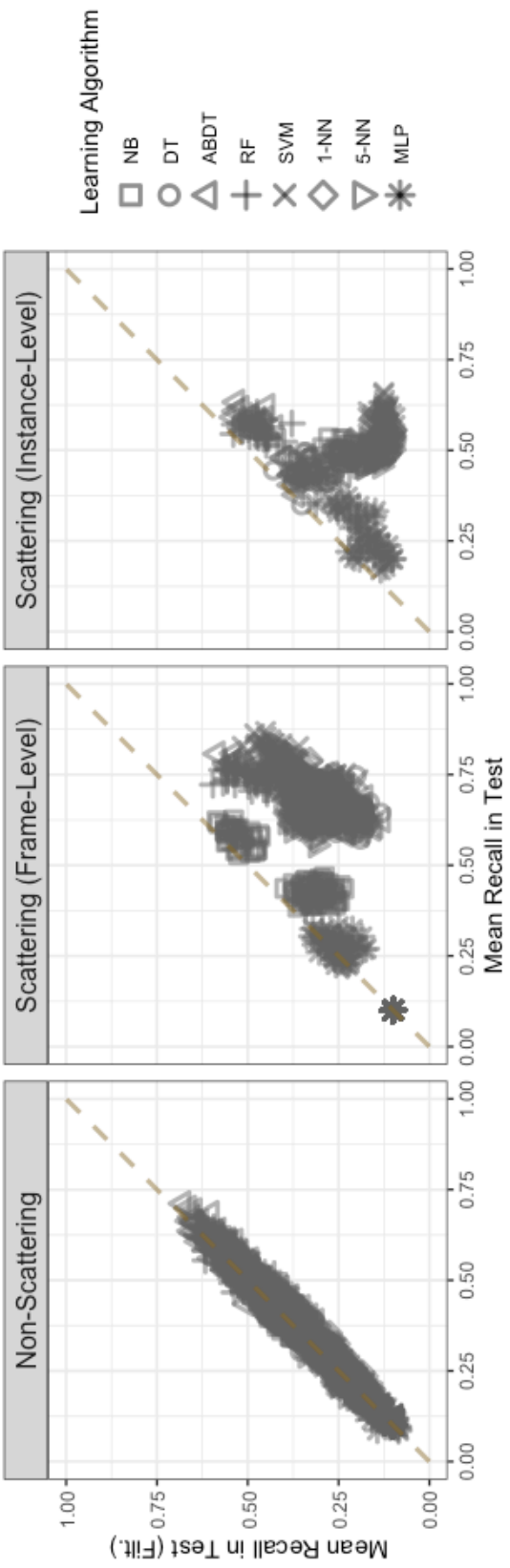


Figure 5.7: Relationship between mean recall in `test` and `test (filt.)` obtained by systems constructed with different combinations of feature representations and learning algorithms using training collections sampled from *GTZAN* with bootstrap regulated over artists, grouped by the source of feature set. Non-Scattering features are extracted with `Essentia`. Instance-level scattering features correspond to `Des. 1-LSc.`; the rest are frame-level. The dashed line indicates the case of equal mean recall on `test` and `test (filt.)`.

Figure 5.7 separates measurements from systems using `Des. 1-L Sc.` because the clusters they form suggest interactions with learning algorithms different from frame-level scattering systems. Leaving `MLP` systems aside, the clusters close to the dashed line in the middle panel only contain measurements from `NB` systems. Their average decrease in mean recall is of 9 percentage points, corresponding to a 19% drop. `NB` systems with `Des. 1-L Sc.` feature representations, however, suffer an average 52% decrease. Conversely, the clusters closer to the ideal case for `Des. 1-L Sc.` systems correspond to algorithms of a similar kind: `DT`, `ABDT`, and `RF`. The average drop in performance for these algorithms is between 15% and 25% with `Des. 1-L Sc.` feature representations, but `DT` is the algorithm with the largest drop for the rest of the scattering-based representations, with an average 61.5% decrease in mean recall; `ABDT` follows with 55.8% decrease.

5.3.5 Factorial Integration of Interventions

The separate analyses above highlight the particularities of each confounding effect. Both interventions are now conducted simultaneously in a factorial way, exposing every trained system to all evaluation conditions. In particular, `pr. test (filt.)` contains the same instances as `pr. test` but high-pass filtered.

Figure 5.8 summarises the performance distributions on `test` and `pr. test`, both under original and filtered audio conditions, marginalised by *GTZAN* class, feature representation and learning algorithm. The distribution on `pr. test (filt.)` is centred around lower values than those on any other evaluation condition for scattering-based representations. Systems using `non-scattering` feature representations only suffer drops when regulating over artist, but not due to high-pass filtering.

Combining multiple interventions permits analysing interactions between confounders. Using the notation in Sec. 5.2.2, let $\hat{y}, \hat{y}'_1$, $\hat{y}'_2$, and $\hat{y}'_{1,2}$ be the mean recall a system obtains on `test`, `pr. test`, `test (filt.)`, and `pr. test (filt.)`, respectively. Let Δ_A be the "accumulated" variation of mean recall, defined as in Eq. (5.3), and Δ_R be the "real" variation, defined as in Eq. (5.4). Figure 5.9 shows the distribution of $\Delta_R - \Delta_A$, grouped by origin of feature set. This difference is centred around 0 for systems using `non-scattering` feature representations, since the overall confounding effect in those systems originates mainly from artist replication. On the other hand, the difference tends

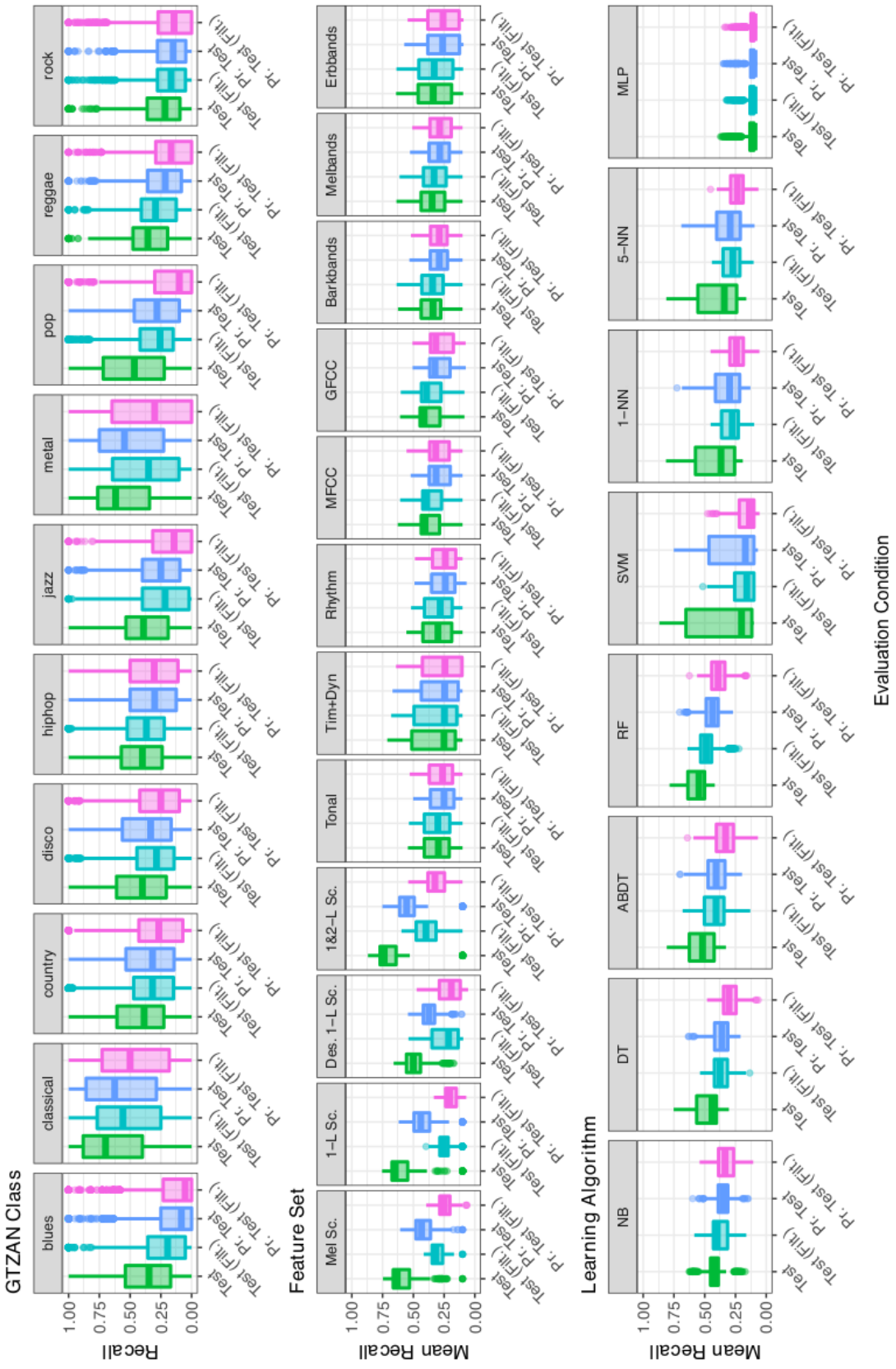


Figure 5.8: Quartiles of (mean) recall distribution obtained on `test`, `test (filt.)`, `pr. test`, and `pr. test (filt.)`, marginalised over *GTZAN* class (Top), feature set (Middle), and learning algorithm (Bottom). Note that the colours in this figure not matching those in Figs. 5.1, 5.2, 5.4 and 5.6 correspond to different evaluation conditions.

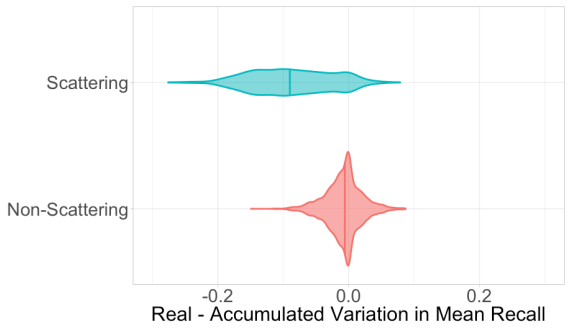


Figure 5.9: Distribution of differences between the real variation Δ_R and the accumulated variation Δ_A in mean recall for artist and infrasonic regulation interventions on *GTZAN*, grouped by the source of feature set.

to be negative for systems using scattering-based feature representations. This suggests the two confounding effects overlap for those systems, which stands to reason since the recording conditions of excerpts from the same artist are likely similar.

Confounders not only impact the magnitude of performance estimates, as seen before, but also alter their ranking. For instance, Fig. 5.10 shows that, for systems trained using `1&2-L Sc.`, `NB` goes from the lowest (ignoring `MLP`) to the highest position depending on whether one manipulates the data; similar interactions arise in other methods. Sturm (2014d) also observes that Naive Bayes systems seem less vulnerable to the faults of *GTZAN*.

The overall ranking of systems depends on the evaluation condition, as Fig. 5.11 reflects. Kendall's τ provides estimates of concordance between rankings, with 1 meaning exact match, -1 completely reversed match, and 0 non-correlation (Kendall, 1938). The value of τ between `test` and `pr. test` is fairly high (0.91), which aligns with our interpretation that artist information biases performance estimates in a similar way across methods (i.e., without substantially altering their ordering). τ decreases between `test` and `test (filt.)` (0.52) and between `test` and `pr. test (filt.)` (0.45), reflecting the fact that infrasonic content affects ranking to a higher degree.

5.4 Discussion

The proposed procedure for characterising confounding effects in music classification experiments helps to understand how particular confounders impact evaluation outcomes.

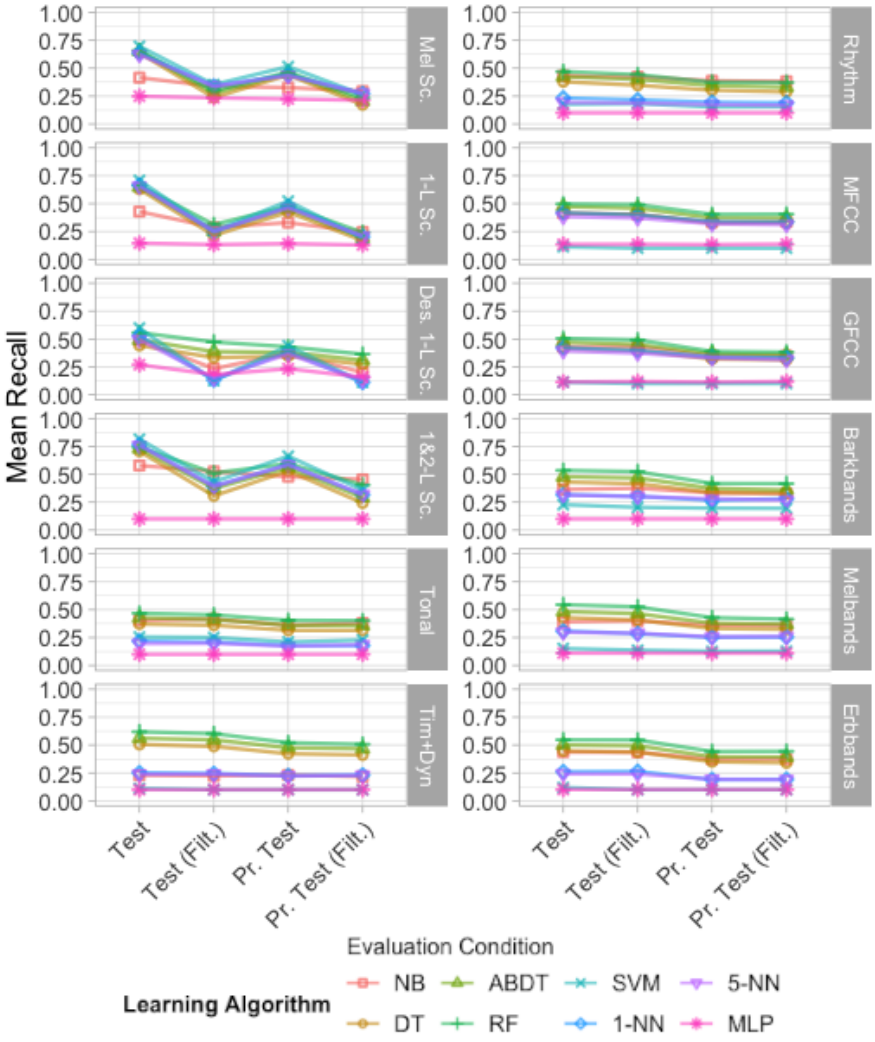


Figure 5.10: Interaction between learning algorithm and evaluation condition in average mean recall for systems constructed using training collections sampled from *GTZAN* with bootstrap regulated over artists across feature sets.

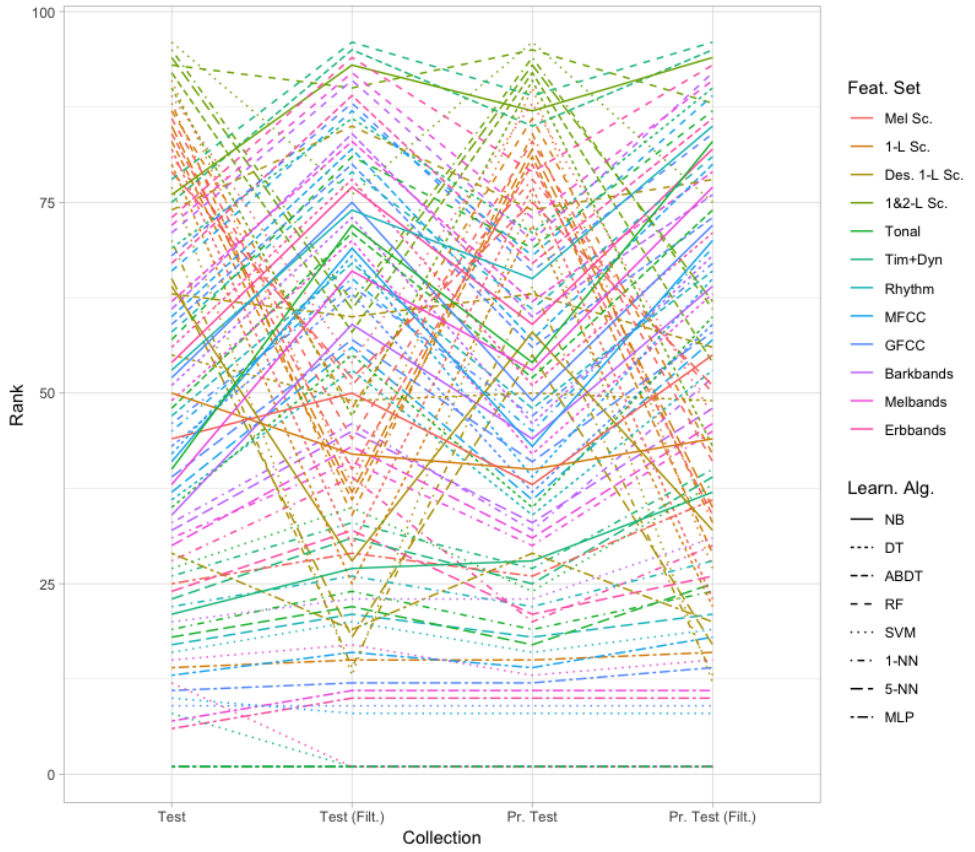


Figure 5.11: Interaction between system-construction method and evaluation condition in rank of average mean recall for systems constructed using training collections sampled from *GTZAN* with bootstrap regulated over artists. Ties in average mean recall are resolved by assigning the minimum possible rank to all involved methods.

It extends well-established practices in MIR, such as filtered partitioning, overcoming their limitations. In particular, the approach presented here enables to integrate multiple types of interventions, targeted to the same or distinct potential confounders (but not necessarily multiple interventions of the same type). Introducing a suitable resampling strategy, such as the described regulated bootstrap, is key to this integration. This provides a distribution of regulated/unregulated measurement pairs instead of single sample comparisons, such as those found in previous studies (e.g., Rodríguez-Algarra et al., 2016). It also enables to disentangle the effects of confounding between training and prediction.

The example application using *GTZAN* showcases the benefits of the proposed procedure. The factorial structure across runs of the experiment enables both marginal and joint analyses, revealing distinct behaviours when systems are exposed to each poten-

tial confounder, as well as their interactions. These observations, however, are subject to some caveats discussed next.

Systems in the case study underperform due to the lack of hyperparameter tuning. Variety is deliberately prioritised over optimisation to gather performance estimates of different magnitude and susceptibility to confounding. The evidently unsuitable `MLP` architecture chosen is a clear example of this, yielding measurements close to the random baseline that could still be affected by the regulations. Alternatives to achieve measurements in the lower end, such as random or systematic classifiers, would by definition remain unaffected regardless of the condition. Tuning model hyperparameters, while relevant for benchmarking studies, would likely concentrate performances at the high end of the axis, thus hampering the intended illustration of the proposed methodology. Further studies could incorporate additional treatment conditions in the experimental design related with hyperparameter optimisation, which may help illuminate how tuning impacts the susceptibility to confounding effects.

The analysis suggests the confounding effect of artist replication in *GTZAN* appears multiplicative rather than additive. This might seem obvious knowing that the performance metric used is bounded between 0 and 1. As Carterette (2012) mentions, additive effects could easily make predicted values exceed those boundaries. In reality, current proposals for modelling measurements from classification experiments, such as those reported in Sec. 3.3, assume additive effects for all components of the experiment, ignoring the boundary problem. This motivates revising those models, potentially using *logit* transformations to convert multiplicative effects into unbounded additive components, although it might be unnecessary if one's only concern is the ranking between systems.

The clear divergence between the proposed linear model and the observations of the highest end of performance measurements in Fig. 5.5 might require collecting further data, either from not yet considered methods or through the optimisation of existing ones. That divergence, however, illuminates a substantial difference in slope between observations using a particular feature representation and the overall trend. This seems to reflect Simpson's paradox (Pearl, 2014; Simpson, 1951), in which behaviour per group diverges from, or even completely reverses, the aggregated pattern. Together with the clusters suggested in Fig. 5.7 for the case of infrasonic content, this highlights the need to study

interactions between learning algorithms and feature representations under various potentially confounding environments.

A general limitation of the proposed method regards its scope, since it neither illuminates previously unknown confounders nor prevents confounding from affecting performance estimates. It is actually impossible to guarantee that confounding does not appear at all, as there might be a plethora of yet unknown potential confounders still affecting observations to some extent. Devoted exploratory analyses informed by both domain knowledge and system analysis are necessary to uncover further potential confounders before assessing their impact using intervention-type approaches. This enables to design or improve system-construction methods accounting for that risk and devise train/test mechanisms that prevent them from appearing. To this end, it is of paramount importance for MIR researchers to devote efforts to expose such potential confounders and assess their effects.

The study reported in this chapter does not consider all possible effects of confounding, focusing on characterising its effects on evaluation results, but leaving aside other equally relevant research questions for the moment. In particular, by introducing and comparing new conditions only at the prediction stage, the effects of confounding on the training of systems are ignored. This might be easily addressed for data manipulation interventions by adding training conditions with manipulated recordings, thus multiplying the number of models to consider and experimental conditions to analyse. In the case of instance assignment interventions, however, it would require modifying the regulated bootstrap resampling strategy to enable the creation of regulated and unregulated collections for both training and testing simultaneously. This is a promising research path for the future.

The threshold number of recordings n_r, as presented in Alg. 1, poses a further limitation to the current implementation of the regulated bootstrap resampling strategy. Since its value is absolute, all classes in the collection must adhere to it regardless of their size. For imbalanced collections, this might be problematic. Further implementations of the algorithm can easily overcome this issue by replacing the n_r parameter in the function definition with a relative threshold η_r, with value between 0 and 1. It would then suffice to append $n_r \leftarrow \lfloor \eta_r \cdot |\boldsymbol{C}_\mathrm{a}| \rfloor$ in step 0 of Alg. 1 to retain the original class size distribution. The

symbol pair $\lfloor\,\rfloor$ represents a floor function, but any other rounding transformation would be equally suitable. Since the classes in *GTZAN* are balanced, modifying the resampling algorithm in this manner would not alter the results reported in Sec. 5.3.

Some may argue the curation process inherent to regulated bootstrap resampling introduces biases in the performance estimates, and thus in the comparisons between conditions, questioning the validity of the extracted conclusions. This process, however, increases control over the measurements, not unlike the stratification performed in conventional classification experiments, as well as blocking in statistical Design of Experiments (Montgomery, 2013). In particular, stratification preserves the distribution of annotations present in the original collection, thus facilitating performance estimates within the collection that approximate what systems would have achieved had they used the whole collection, but does not account for the likely imbalances that real life data could have. This favours internal over external validity, a methodological trade-off often encouraged to create experimental conditions that differ only in the factor under study and warrant against external factors affecting the conclusions (Shadish et al., 2002).

The size of the testing collections generated might also cause concern, since there is no guarantee that the original class balance remains and, by definition, the number of instances decreases after pruning. The use of mean recall as performance metric should compensate for imbalances, and, in the case study conducted here, the differences in performance between collections of the same iteration clearly exceed the differences across iterations. This suggests unequal size should not affect the conclusions reached here. As mentioned before, in the general case, one might want to introduce a further control step that forces all original and pruned testing collections, and all classes within those collections, to match in size, such as randomly selecting a fixed number of instances. This might also alleviate the likely lack of independence between instances from the curation involved in their sampling. Due to the infeasibility of pure random sampling from the whole population, the convenience sampling often involved in the construction of evaluation collections hampers independence in the first place. Curation thus does not necessarily affect in this regard.

The analysis approach described and exemplified in this chapter can be applied to a wider range of collections, Machine Learning methods and potential confounders than

the ones considered here. Published studies and evaluation exchanges, such as MIREX, could incorporate similarly extended pipelines to assess the susceptibility of proposed systems to a set of interventions. Domains other than music would also benefit from similar analysis approaches. Despite its caveats, the insights obtained through this kind of analysis should help building more robust systems and obtaining performance estimates that generalise to deployment scenarios.

CHAPTER 6

STRUCTURAL MODELLING OF MEASUREMENTS IN CLASSIFICATION EXPERIMENTS

The extensions of the conventional classification experiment proposed in previous chapters leverage fundamental principles of experimental design, especially factorisation and replication, to better understand evaluated systems through interventions. The example analyses largely focused on comparing regulated and unregulated conditions, but, similar to what one would commonly find in benchmarking studies, also highlighted differences between system-construction methods. For the sake of simplicity, however, such analyses did not explicitly express the underlying structural models leading to the decomposition of measurements into contributions.

This chapter introduces structural models suitable for the analysis of classification experiments in applied Machine Learning scenarios, including those with pipeline interventions. These models help distinguish between relevant and nuisance contributions to the performance measurements, and are fundamental for any inferential analysis one aims to conduct. Although the nomenclature used here follows the conventions of frequentist statistics, the ideas behind the presented models could easily be adapted to a Bayesian context. This chapter also discusses the suitability of the assumptions underlying linear additive models for the analysis of measurements from classification experiments, com-

paring them with alternative logistic approaches. Finally, based on the insights gained from the structural models discussed, this chapter revisits some implementation decisions from the case studies in previous chapters, and suggests steps that readers might want to undertake when conducting their own studies.

6.1 Fundamental Structural Models for Classification Experiments

The pipeline illustrated in Fig. 2.2 shows that each measurement in a classification experiment depends on at least five components: the collection $\boldsymbol{C}$, the assignment function ψ, the iteration k, the system construction method m, and a performance metric function ϕ, i.e., $\hat{y} = f(\boldsymbol{C}, \psi, k, m, \phi)$, even though $\boldsymbol{C}$, ψ, and ϕ are likely to be fixed in a particular study. The relationship between those components is anything but trivial. Successively substituting backwards from the performance estimate $\hat{y}$ yields:

$$\begin{aligned}
\hat{y} &= \phi(\hat{\boldsymbol{A}}_p,\ \boldsymbol{A}_p) \\
&= \phi(p(\boldsymbol{F}_p),\ \boldsymbol{A}_p) \\
&= \phi(\ell(\boldsymbol{F}_p \mid \boldsymbol{F}_t, \boldsymbol{A}_t),\ \boldsymbol{A}_p) \\
&= \phi(\ell(e(\boldsymbol{R}_p) \mid e(\boldsymbol{R}_t), \boldsymbol{A}_t),\ \boldsymbol{A}_p) \\
&= \phi(\ell(e(\psi_{p,R}(\boldsymbol{C}, k) \mid e(\psi_{t,R}(\boldsymbol{C}, k)), \psi_{t,A}(\boldsymbol{C}, k)),\ \psi_{p,A}(\boldsymbol{C}, k)))
\end{aligned} \tag{6.1}$$

where the subindices in ψ indicate which parts of the output of the assignment at an iteration k to keep in each case (t for training, p for prediction, R for raw data, and A for annotations), and the composition of e and ℓ forms m. The relationship in Eqn (6.1) is excessively complex for most practical purposes. Instead, analyses implicitly or explicitly presume simpler relationships between the effects of each component of interest and the measurements in the form of structural models.

The models presented here follow the same basic assumptions as those in Sec. 3.3 for the particular case of the evaluation of learning algorithms. They are mixed-effects linear additive models, meaning they express the measurements as the sum of parameters for both fixed and random effects, plus a residual that captures the variability not explained by the other considered parameters. Fixed-effects parameters represent factors whose impact on the measurements is of interest, whereas random-effects parameters represent nuisance factors. Which ones should be considered as one type or the other largely depends on the research question one aims to answer.

Classification experiments can be conducted for several reasons, such as to select a

fixed system for deployment, to compare the suitability of different methods, or to assess specific components or hyperparameters of such methods. Although the computational machinery employed in any of those scenarios is virtually identical, which information is of interest differs, implicitly demanding different structural models. What follows introduces various of these possible structural models, starting from the simplest scenario and subsequently increasing complexity.

6.1.1 Assessing Fixed Systems

Consider J fixed systems, s_j ($j = 1,\ldots,J$), are assessed on their performance in reproducing the annotations of a given collection $\boldsymbol{C}$. These systems are treated as pure black boxes, with their evaluation completely ignoring how they are constructed. Assume that the whole $\boldsymbol{C}$ remains available to perform predictions, either because the systems are expert agents or have all been trained using a separate set of instances, so overfitting is not a concern and no further partitioning is necessary.[1] The goal is to determine whether any system appears superior with regards to how predictions $\hat{\boldsymbol{A}} = (\hat{a}_1,\ldots,\hat{a}_N)$ match the annotations $\boldsymbol{A}$, with $\hat{a}_n = s(r_n)$ — the output of the system to the raw data instance r_n.[2] The measurements are then derived from a performance metric $\phi(\hat{\boldsymbol{A}},\boldsymbol{A})$, which can be analysed treating the whole collection as a single observational unit, or splitting the collection into multiple smaller subsamples and obtaining separate performance measurements for each. These two options are discussed next.

Single Sample One approach for comparing multiple systems involves measuring the overall prediction performance of each on a collection $\boldsymbol{C}$. Each run of a classification experiment, then, fixes a system factor variable $\mathcal{S}$ to a particular level $\mathcal{S}(i) = s$. The different conditions to compare correspond to the J systems, hence the levels of $\mathcal{S}$ comprise the treatment set. In the simplest situation $J = 2$, with one of the treatment levels representing a single actual system and the other a baseline.

Each system corresponding to a level $s \in \mathcal{S}$ produces a single sequence of predictions from $\boldsymbol{C}$, with each run of the experiment yielding a response $y = \phi(\hat{\boldsymbol{A}},\boldsymbol{A})$. $\boldsymbol{C}$ as a whole

[1] If only part of a collection remained available, then that part would become $\mathbf{C}$ for the purposes of the present analysis.

[2] For simplicity, the subindex j is removed hereinafter, writing s instead to express a generic level in $\mathcal{S}$, unless necessary to disambiguate.

acts as observational unit. Note no structure is considered in either treatments or units. This means that any common characteristics that would group treatments into factors or units into blocks are ignored. The structural model that describes this scenario includes a single parameter related to the system factor $\mathcal{S}$, which is assumed to produce a fixed effect $\tau_{\mathcal{S}(i)}$:

$$y_i = \mu + \tau_{\mathcal{S}(i)} + \varepsilon_i. \tag{6.2}$$

This mirrors the CRD model in Eqn (3.1).

This setting omits some of the principles of DoE introduced in Sec. 3.1.2. In particular, randomisation is unnecessary, since the batches of instances considered as observational units are duplicated in multiple runs of the experiment, applying every distinct treatment to every unit. Problematically, replication is also ignored, since a single observation is obtained for each treatment level in the study, i.e., only one measurement per system. Since the number of observations N matches the number of treatments J, the degrees of freedom of the equality factor $\mathcal{E}$, computed as in Eqn (3.30), are $d_{\mathcal{E}} = N - J = N - N = 0$. This means the effect of the systems, if any, cannot be disentangled from the particular sample used in the study.[3] No statistical machinery can overcome this issue. One might try to address this issue by computing multiple prediction sequences $\hat{\boldsymbol{A}}_k$, obtaining K performance measurements $\phi(\hat{\boldsymbol{A}}_k, \boldsymbol{A})$ for each system. These, however, are "false replications" since they are not independent observations, and so do not provide any extra degrees of freedom for isolating the system effect in Eqn (6.2).

Multiple Samples To overcome the issue above, one may consider multiple distinct batches of instances as observational units instead of a single one. Splitting $\boldsymbol{C}$ into K disjoint samples and performing (at least) one measurement in each sample for each system avoids conflating the effect of particular samples with that of the systems.

Ignoring structure in the units, thus considering only the system factor $\mathcal{S}$ as relevant, leads to a structural model identical to that in Eqn (6.2), but with $J \times K$ "true" measurements in this case. Since $N = J \times K$, the degrees of freedom of the equality factor are

[3] Incidentally, this also means the residual parameter $\boldsymbol{\varepsilon}$ could be dropped from the model in Eqn (6.2) without loss of information, since the estimated treatment effect alone suffices to explain completely the variability of the measurements.

$$d_{\mathcal{E}} = N - J = (J \times K) - J = J \times (K-1) > 0,\ \forall K > 1$$

unlike in the single sample case above. Fitting such a model to the observed performances yields estimates of the overall system effects on $\boldsymbol{C}$, enabling comparisons.

Not considering structure assumes that the K samples affect all measurements in exactly the same way (or none at all). In other words, it does not account for possible differences in "difficulty" among samples that may consistently introduce variability into the observations not explained by the systems. The k-th run associated with each system yields a performance measurement on exactly the same sample, $\boldsymbol{C}^k$, which exposes a clear structure in the units. This suggests considering samples as blocking variables.

If the effect of particular samples on the measurements is unimportant, with only that of the systems being deemed relevant, it may be assumed that the samples introduce a random effect with zero mean and unknown variance, as Eugster (2011) suggests for the evaluation of learning algorithms. The structural model that reflects this situation is:

$$y_i = \mu + \tau_{\mathcal{S}(i)} + \beta_{\mathcal{K}(i)} + \varepsilon_i. \tag{6.3}$$

This only differs from Eqn (6.2) in that it adds the parameter $\beta_{\mathcal{K}(i)}$ to represent the random effect of the samples. The model matches the one usually considered for a CBD in conventional DoE, such as in Eqn (3.3). Fitting (6.3) enables to estimate system effects, while controlling for the variability introduced by the samples.

This approach assumes the effect of each sample introduces the same variability independently of the system. In other words, this model does not permit estimates of the potential *interaction* between systems and samples. The total number of observations in this approach matches the number of possible combinations of systems and samples ($J \times K$). The infimum $\mathcal{S} \wedge \mathcal{K}$ between the system and sample factors is thus equivalent to the equality factor $\mathcal{E}$. This means only one measurement is obtained per combination, hence one lacks replicates to conclude anything with regards to the effect of such combinations.

Although assuming no interaction might often be reasonable, previous research shows that particular samples may affect the performance of distinct systems differently (e.g., Pampalk et al., 2005; Rodríguez-Algarra et al., 2016). Regarding fixed systems as the treatments of the study completely obscures this circumstance. Repeating measurements

would not help in this case either, since, with rare exceptions, systems yield the same predictions for the same samples and thus lead to the same performance measurements. Therefore, obtaining more measurements for the same combinations of systems and samples would not add any power to the analysis. Pipeline interventions might serve to address this issue, introducing further evaluation conditions for each combination.

6.1.2 Assessing System-Construction Methods

Suppose each of J systems comes from the application of one of $M \leq J$ distinct methods, and one is interested in assessing differences in performance between such methods. Each of the methods is represented by a level of a factor variable $\mathcal{M}$, which becomes the main treatment variable of the experiment instead of the systems. In here, entire methods are considered as indivisible entities, but Sec. 6.1.3 later deals with the particular contributions of the feature extraction and learning algorithms that form such methods.

Similar to Sec. 6.1.1, one can formulate structural models with varying complexity to assess system-construction methods. Adapting Eqns (6.2) and (6.3) to this case leads to the following models:

$$y_i = \mu + \tau_{\mathcal{M}(i)} + \varepsilon_i \tag{6.4}$$

$$y_i = \mu + \tau_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \varepsilon_i. \tag{6.5}$$

The parameter $\tau_{\mathcal{M}(i)}$ represents the fixed effect of the method $\mathcal{M}(i)$, and replaces $\tau_{\mathcal{S}(i)}$. The main difference lies in which units have the same treatment factor level, since all units with the same level in $\mathcal{S}$ necessarily have the same level in $\mathcal{M}$, but not the other way around. The factor $\mathcal{S}$ is thus nested in $\mathcal{M}$.

The difference between system and method levels stands out in a conventional K-fold Cross-Validation (K-CV) setting. In K-CV, one constructs K distinct systems for any particular method $m \in \mathcal{M}$, each associated with one of K train-test sample pairs. This means K measurements are made at each level of $\mathcal{M}$. Using the approach in Sec. 6.1.1, on the other hand, only yields a single observation per level of $\mathcal{S}$ regardless of the number of splits in $\boldsymbol{C}$, leading to the issues highlighted above.

Resampling schemes, such as K-CV or the bootstrap, produce economic estimates of method rather than system effects. These conventional resampling schemes do not over-

come the lack of interaction replications previously discussed, however, since the total number of observations is $M \times K$. This means there are exactly as many observations as combinations of levels of $\mathcal{M}$ and $\mathcal{K}$, hence leaving no room for estimating interaction effects between methods and samples.

Note that the formalisation here not only resembles the models introduced above for the evaluation of fixed systems, but it is also almost identical to those presented in Eqns (3.40) and (3.41) for the evaluation of learning algorithms. When comparing learning algorithms, one conventionally assumes all such algorithms receive identically represented data — i.e., features have been extracted from the raw data using the same extractor. In that case, each system-construction method corresponds with a particular learning algorithm, hence $\mathcal{L} \equiv \mathcal{M}$. This is not generally true in applied Machine Learning scenarios, where multiple feature extraction and learning algorithms might appear simultaneously in a single study. What follows considers this situation.

6.1.3 Assessing Method Components

Researchers often intend to evaluate particular *components* of methods, such as feature extractors and/or learning algorithms, rather than entire methods. The structural models below address this under various assumptions with increasing complexity. Similar models would also apply in the case of hyperparameter tuning, replacing (or nesting) the factor variables for each component with a variable for each hyperparameter, whose levels represent the distinct values that one wishes to consider (such as in a grid-search optimisation strategy).

No interaction between components Denote $\mathcal{X}$ a factor variable with levels representing feature extraction algorithms; similarly, denote $\mathcal{L}$ a factor representing the learning algorithms. Both $\mathcal{X}$ and $\mathcal{L}$ are treatment factors, since their effects constitute the evaluation conditions to compare. A simple modelling approach that considers both factors only includes parameters for the fixed effects of $\mathcal{X}$ and $\mathcal{L}$:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \varepsilon_i . \tag{6.6}$$

This implicitly assumes that neither the samples nor the interaction between components affect the measurements.

In practice, much published research simplifies this even further. Studies often attempt to assess one or more feature extraction algorithms on a collection using a single learning algorithm. This is the case, for instance, of the study conducted by Andén and Mallat (2014) described in Sec. 2.4. Such a study compares a series of feature extraction techniques against what they consider a state-of-the-art baseline algorithm on a benchmark collection. For this purpose, the authors solely use a Support Vector Machine (SVM) as learning algorithm. This situation mirrors the CRD model in Eqn (6.4), with a structural model such as:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \varepsilon_i \tag{6.7}$$

implicitly assuming that the measurements depend solely on the feature extractors to compare. Conversely, some other studies fix the feature representation and compare learning algorithms, which would entail replacing $\tau_{\mathcal{X}(i)}$ with $\tau_{\mathcal{L}(i)}$ in the model, but would be otherwise equivalent.

Neither of these cases produces completely generalisable estimates. Measurements that fix one component do not provide any support for conclusions beyond the combination with that particular component. For instance, differences between feature extractors in the study conducted by Andén and Mallat (2014) could have arisen merely as a consequence of their combination with SVM, and not appear if used alongside other learning algorithms.

A straightforward alternative involves comparing various combinations of feature extractors and learning algorithms. The structural model in Eqn (6.6) seems to suit this situation, since it separates the effects of $\mathcal{X}$ and $\mathcal{L}$. Note, however, that both effects are presumed fixed. This implicitly assumes that the particular selection of levels covers all possibilities of interest, which rarely holds when evaluating a component irrespective of the choice of the other. If the levels of one component factor only cover a narrow selection among a wide range of possibilities, it may be more appropriate to treat their effects as random instead of fixed. The following models express this situation:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \beta_{\mathcal{L}(i)} + \varepsilon_i \tag{6.8}$$
$$y_i = \mu + \beta_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \varepsilon_i. \tag{6.9}$$

They both take the same form as the CBD in Eqn (6.5), though modelling different effects. While technically part of the treatment, the factor whose effect is modelled as random acts as a blocking variable. These models remove the variability associated with the blocks from the estimate of interest. For instance, Andén and Mallat (2014) could have included a selection of learning algorithms beyond SVM in their study, obtaining measurements for all combinations of feature extractors and algorithms but focusing their analysis on the feature extractors by modelling the effects of the learning algorithms as random, such as in Eqn (6.8).

Method effects The previous models ignore possible interactions between components, even though it is often interesting to determine whether particular combinations appear more successful than others. A classification experiment that includes all possible combinations of the levels of $\mathcal{X}$ and $\mathcal{L}$ matches an FD, such as the one in Eqn (3.2). This allows estimates of both individual and interaction effects:

$$\begin{aligned} y_i &= \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \tau_{\mathcal{X}\mathcal{L}(i)} + \varepsilon_i \\ &= \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \tau_{\mathcal{M}(i)} + \varepsilon_i. \end{aligned} \tag{6.10}$$

The effect of the interaction $\mathcal{X}\mathcal{L}$ is expressed as the contribution of the methods, $\tau_{\mathcal{M}(i)}$, because each combination of a feature extractor e and a learning algorithm ℓ defines a different method. The use of fixed effects here reflects that one focuses on the particular selection of feature extractors and learning algorithms considered in the study.

Estimating interaction effects may provide relevant information even when the particular combinations included in the study are irrelevant. Say one wants to compare a feature extraction technique with a state-of-the-art approach. To this end, one trains a number of common learning algorithms, but the aim is to assess whether there exists a differential effect between approaches regardless of the learning algorithm. Considering interaction effects may reveal that differences only hold under particular circumstances and not others. In this scenario, both the effect of the learning algorithms as well as that

of the interaction should be modelled as random, suggesting a so-called *generalised block design* (GBD) model:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \beta_{\mathcal{L}(i)} + \beta_{\mathcal{M}(i)} + \varepsilon_i. \tag{6.11}$$

Furthermore, swapping the parameter for fixed and random effects compares learning algorithms regardless of the feature representations:

$$y_i = \mu + \beta_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{M}(i)} + \varepsilon_i. \tag{6.12}$$

These GBD models thus permit focusing on the effects of a single component controlling for both the effects of another and their interaction.

Regardless of whether one uses an FD or a GBD model in the analysis, the factors in all these cases follow the same structure as the one depicted in Fig. 3.5(c), albeit with differences in which factors belong to the treatment set and which to the plot set. The calculation of the degrees of freedom matches the procedure that Eqn (3.38) reflects. In that case, for $d_{\mathcal{E}}$ to be positive, the number of observations must exceed the number of combinations between feature extractors and learning algorithms. Therefore, including the interaction effects in the analysis is only feasible if multiple performance measurements are obtained from each method, such as with a resampling strategy.

Sample effects and interactions All models discussed so far in this section ignore the effects of the particular samples on the measurements — i.e., they assume that the particular instances employed to obtain each measurement do not affect the performance estimates. To account for the random effect of the samples, including a parameter $\beta_{\mathcal{K}(i)}$ in the FD in Eqn (6.10) results in a so-called *blocked factorial design* (BFD):

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \tau_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \varepsilon_i \tag{6.13}$$

with $\mathcal{K}$ denoting a factor variable whose levels represent train/test samples, such as in Eqns (6.3) and (6.5). Figure 6.1 shows the factorial structure of a BFD. Models of this kind enable estimating the fixed effects of both method components as well as their interactions, controlling at the same time for the nuisance contribution of the particular samples employed. The GBD models in Eqn (6.11) and (6.12) would require similar modifications.

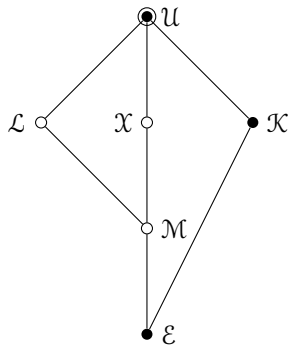


Figure 6.1: Hasse diagram of a blocked factorial design for the analysis of measurements from a classification experiment.

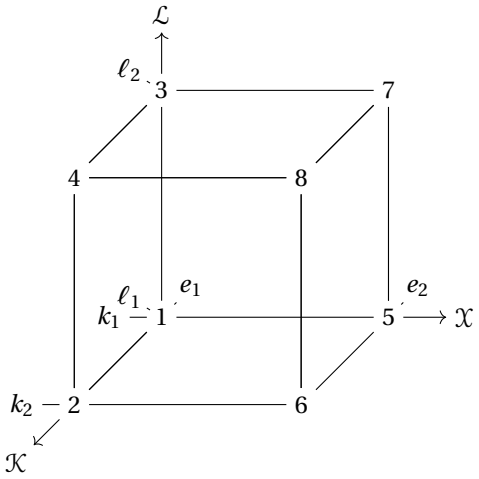


Figure 6.2: Schematic representation of the factor levels corresponding to each observation i in a 2-CV experiment with two feature extractors and two learning algorithms.

Note the model in Eqn (6.13) implicitly assumes that each sample affects the measurements in exactly the same way regardless of its combination with method components. Conventional resampling schemes replicate the different combinations of method components and samples, which enables estimating their possible interactions. Take, for instance, Fig. 6.2 and Table 6.1, which both represent all possible combinations of the levels of $\mathcal{X}$, $\mathcal{L}$, and $\mathcal{K}$ in a 2-CV experiment (i.e., $\mathcal{K}(i) \in \{k_1, k_2\}$), with two feature extractors and two learning algorithms (i.e., $\mathcal{X}(i) \in \{e_1, e_2\}$ and $\mathcal{L}(i) \in \{\ell_1, \ell_2\}$). Each row in the table and node in the figure represents an observation i. There are two observations per level of the interaction factors $\mathcal{M}$, $\mathcal{X}\mathcal{K}$, and $\mathcal{L}\mathcal{K}$, and no two columns follow the same "pattern". All factors are thus mutually orthogonal, which ensures that their effects can be disentangled.

i	$\mathcal{X}$	$\mathcal{L}$	$\mathcal{K}$	$\mathcal{M}$	$\mathcal{X}\mathcal{K}$	$\mathcal{L}\mathcal{K}$	$\mathcal{M}\mathcal{K}$
1	e_1	ℓ_1	k_1	$e_1\ell_1$	e_1k_1	ℓ_1k_1	$e_1\ell_1k_1$
2	e_1	ℓ_1	k_2	$e_1\ell_1$	e_1k_2	ℓ_1k_2	$e_1\ell_1k_2$
3	e_1	ℓ_2	k_1	$e_1\ell_2$	e_1k_1	ℓ_2k_1	$e_1\ell_2k_1$
4	e_1	ℓ_2	k_2	$e_1\ell_2$	e_1k_2	ℓ_2k_2	$e_1\ell_2k_2$
5	e_2	ℓ_1	k_1	$e_2\ell_1$	e_2k_1	ℓ_1k_1	$e_2\ell_1k_1$
6	e_2	ℓ_1	k_2	$e_2\ell_1$	e_2k_2	ℓ_1k_2	$e_2\ell_1k_2$
7	e_2	ℓ_2	k_1	$e_2\ell_2$	e_2k_1	ℓ_2k_1	$e_2\ell_2k_1$
8	e_2	ℓ_2	k_2	$e_2\ell_2$	e_2k_2	ℓ_2k_2	$e_2\ell_2k_2$

Table 6.1: Factor levels and their interactions for each observation of a 2-CV classification experiment with two feature extractors and two learning algorithms.

No two entries of the column corresponding to $\mathcal{M}\mathcal{K}$ in Table 6.1 match, which indicates that every single observation corresponds with a unique combination of feature extractor, learning algorithm and train/test sample. This means there are no replications left available for estimating possible three-way interaction effects (i.e., between entire methods and samples). As usual, this leads to non-positive degrees of freedom for the equality factor. Including the factor $\mathcal{M}\mathcal{K}$ in the analysis would lead to a factor structure like the one in Fig. 3.6(c), other than possibly the set to which each factor belongs. Let L, X and K be the number of levels of factors $\mathcal{L}$, $\mathcal{X}$ and $\mathcal{K}$, respectively. Following the cascading procedure to obtain the degrees of freedom for each factor yields:

$$
\begin{aligned}
d_{\mathcal{U}} &= 1 \\
d_{\mathcal{L}} &= L - d_{\mathcal{U}} = L - 1 \\
d_{\mathcal{X}} &= X - d_{\mathcal{U}} = X - 1 \\
d_{\mathcal{K}} &= K - d_{\mathcal{U}} = K - 1 \\
d_{\mathcal{M}} &= L \cdot X - (d_{\mathcal{U}} + d_{\mathcal{L}} + d_{\mathcal{X}}) = L \cdot X - (1 + (L-1) + (X-1)) = (L-1)(X-1) \\
d_{\mathcal{L}\mathcal{K}} &= L \cdot K - (d_{\mathcal{U}} + d_{\mathcal{L}} + d_{\mathcal{K}}) = L \cdot K - (1 + (L-1) + (K-1)) = (L-1)(K-1) \\
d_{\mathcal{X}\mathcal{K}} &= X \cdot K - (d_{\mathcal{U}} + d_{\mathcal{X}} + d_{\mathcal{K}}) = X \cdot K - (1 + (X-1) + (K-1)) = (X-1)(K-1) \\
d_{\mathcal{M}\mathcal{K}} &= L \cdot X \cdot K - (d_{\mathcal{U}} + d_{\mathcal{L}} + d_{\mathcal{X}} + d_{\mathcal{K}} + d_{\mathcal{M}} + d_{\mathcal{L}\mathcal{K}} + d_{\mathcal{X}\mathcal{K}}) \\
&= L \cdot X \cdot K - (1 + (L-1) + (X-1) + (K-1) + (L-1)(X-1) + (L-1)(K-1) + (X-1)(K-1)) \\
&= L \cdot X \cdot K - L \cdot X - L \cdot K - X \cdot K + L + X + K - 1 = (L-1)(X-1)(K-1) \\
d_{\mathcal{E}} &= N - (d_{\mathcal{U}} + d_{\mathcal{L}} + d_{\mathcal{X}} + d_{\mathcal{K}} + d_{\mathcal{M}} + d_{\mathcal{L}\mathcal{K}} + d_{\mathcal{X}\mathcal{K}} + d_{\mathcal{M}\mathcal{K}}) \\
&= N - ((L \cdot X \cdot K - d_{\mathcal{M}\mathcal{K}}) + d_{\mathcal{M}\mathcal{K}}) = N - L \cdot X \cdot K = 0
\end{aligned}
$$

since $N = L \cdot X \cdot K$. Including a parameter for the interaction $\mathcal{M}\mathcal{K}$ would thus cause the model to be *saturated*, leading to a perfect fit of the measurements but leaving no variance to conduct statistical analyses.

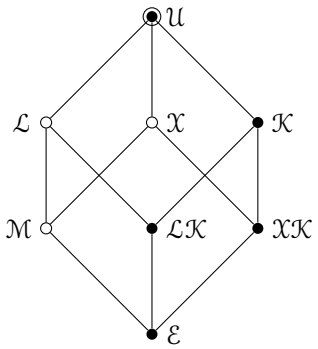


Figure 6.3: Hasse diagram of a generalised blocked factorial design for the analysis of measurements from a classification experiment.

Suitable models for the analysis of measurements from conventional classification experiments thus cannot include the three-way interaction between learning algorithms, feature extractors and train/test samples. All pair-wise interactions, however, are feasible. A *generalised blocked factorial design* (GBFD) with model:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \tau_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \beta_{\mathcal{X}\mathcal{K}(i)} + \beta_{\mathcal{L}\mathcal{K}(i)} + \varepsilon_i \tag{6.14}$$

reflects this. Figure 6.3 shows the factorial structure of a GBFD, which largely resembles the treatment structure of a factorial design with $F = 3$, as in Fig. 3.6(b). Removing the interaction $\mathcal{M}\mathcal{K}$ from the analysis means that the degrees of freedom of the equality factor are now $d_{\mathcal{E}} = (L-1)(X-1)(K-1)$ (formerly the value of $d_{\mathcal{M}\mathcal{K}}$), and the other remain as above.

Some authors, however, seem to disagree on whether structural models including interactions between plot and treatment factors are suitable. Bailey (2008), for instance, rejects such possibility. She argues that the effects of plot factors are modelled as random because their levels generally cover only a subset of all possible values, so interactions between such a subset and the treatment factors would distort the generalisation of conclusions beyond the selected subset. This is a reason why in the Calculus of Factors approach one usually constructs separate Hasse diagrams for plot and treatment sets, and joins them only after each has been finalised. Experimental designs including plot-treatment interactions, however, have long been employed (e.g., Shuster and Eys, 1983). Eugster (2011) uses a similar model when discussing domain-based classification experiments for the evaluation of learning algorithms.

Since it has been shown before that interactions between algorithms and samples exist in classification experiments, what follows assumes that models including parameters for such interactions are acceptable. Models such as:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \beta_{\mathcal{L}(i)} + \beta_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \beta_{\mathcal{X}\mathcal{K}(i)} + \beta_{\mathcal{L}\mathcal{K}(i)} + \varepsilon_i \tag{6.15}$$

if one is only interested in the effect of the feature extractors, and:

$$y_i = \mu + \beta_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \beta_{\mathcal{X}\mathcal{K}(i)} + \beta_{\mathcal{L}\mathcal{K}(i)} + \varepsilon_i \tag{6.16}$$

if the focus lies on the learning algorithms, would therefore be suitable. Including the parameters for the interaction effects in these models removes all nuisance variability from the effects of interest on the measurements. Bailey's (2008) argument, however, encourages a careful interpretation of any estimate of such parameters that could be obtained.

6.2 Design and Analysis of Intervened Classification Experiments

The models presented above permit a more thorough and controlled analysis of measurements obtained from conventional classification experiments. Nevertheless, they do not consider the consequences of modifying the experimental pipeline, as presented in the previous chapters. This section discusses how pipeline interventions translate into the language of experimental design and introduces structural models for the analysis of measurements from classification experiments that include such interventions.

6.2.1 Pipeline Interventions as Factors

As introduced in Sec. 5.2.1, interventions on the experimental pipeline provide global explanations of performance by comparing measurements obtained through the conventional pipeline with those resulting from a specific manipulation designed to alter, or even block, the way systems exploit some source of information. A manipulated pipeline yields a *regulated* evaluation condition, in contrast with the *unregulated* conventional one. To distinguish between these two conditions, an apostrophe represents regulated elements (e.g., ψ' indicates a regulated partitioning function).

Unlike physical experiments, the artificial nature of a classification experiment enables the duplication of any raw data point (a "subject" of the study) an arbitrary number

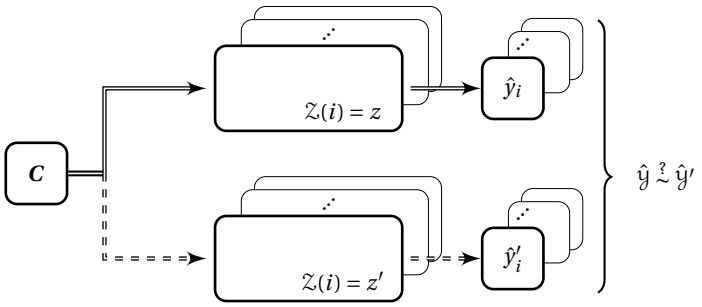
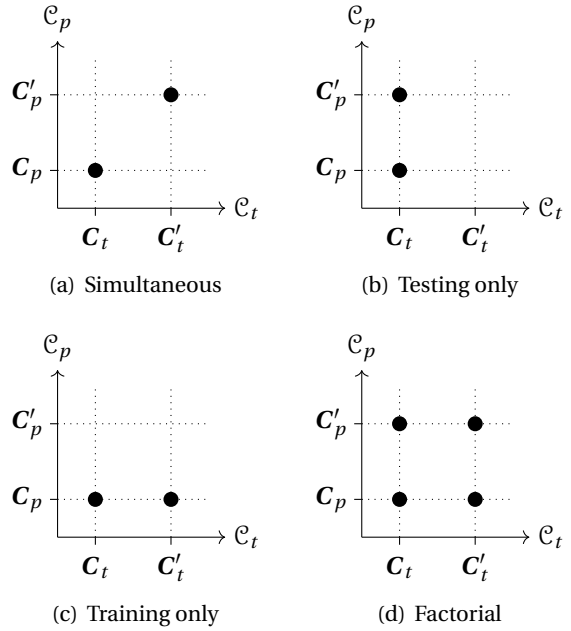


Figure 6.4: Schematic representation of a pipeline intervention creating unregulated (solid arrow path) and regulated (dashed arrow path) evaluation conditions regarding the availability of information source z. $\hat{\mathcal{Y}}$ and $\hat{\mathcal{Y}}'$ represent the distributions of performance measurements $\hat{y}_i$ and $\hat{y}'_i$, respectively.

of times. Each such duplication may be exposed to any number of distinct pipelines without fear of "spills" occurring between measurements. Therefore, the same raw data points may be safely exposed to both unregulated and regulated conditions. Each such condition can be expressed as a level of a factor variable, e.g., $\mathcal{Z}(i) \in \{z, z'\}$. Figure 6.4 schematically represents this, with one path consisting of all measurements for which the factor $\mathcal{Z}$ is z (unregulated) and the other z' (regulated). The goal is for such conditions to differ only in the availability of a particular source of information. In that case, differences in measurements must come from differences in the exploitation of such information. Studies involving pipeline interventions can then assess whether the distributions of measurements under both conditions coincide to conclude whether the considered information source explains the original performance estimates.

From a Calculus of Factors perspective, introducing a pipeline intervention is thus equivalent to adding a further factor $\mathcal{Z}$. For instance, for the artist regulation in the case study of Sec. 5.3, each level of $\mathcal{Z}$ coincides with a different implementation of the partitioning function ψ, i.e., $\mathcal{Z}(i) \in \{\psi, \psi'\}$. Since each level of the new factor essentially copies the entire previous factor structure, the orthogonality of the experimental design is preserved.

Not every manipulation regulating a particular information source is equally suitable as a pipeline intervention, however. Consider a manipulation that, directly or indirectly, alters the makeup of the training and/or testing materials. This results in two factor variables, $\mathcal{C}_t$ (for training) and $\mathcal{C}_p$ (for testing) including one or two levels depending on whether the regulation affects them. Both instance assignment and data manipulation interventions act in this manner. Figure 6.5 shows how different choices in the design



(a) Simultaneous (b) Testing only

(c) Training only (d) Factorial

Figure 6.5: Common combinations of regulated and unregulated collections for training and testing in pipeline interventions, depending on which manipulations are introduced. The axes represent the possible levels of the factor variables governing training ($\mathcal{C}_t$) and testing ($\mathcal{C}_p$); each dot represents an evaluation condition present in the study.

of the regulations impact the resulting training and testing collections; these choices are briefly discussed next.

Simultaneous In Fig. 6.5(a), the manipulation affects both training and testing simultaneously, meaning that in each evaluation condition either both the training and testing collections are regulated, or neither is. The regulated level of the factor $\mathcal{Z}$ then corresponds to the combination $(\boldsymbol{C}'_t, \boldsymbol{C}'_p)$, and the unregulated to $(\boldsymbol{C}_t, \boldsymbol{C}_p)$. The measurements in each evaluation condition are thus obtained from both different systems (different level in $\mathcal{C}_t$) and on different data (different level in $\mathcal{C}_p$). The effects of the regulation on each are confounded, and thus cannot be disentangled. As stated previously, this is the main drawback of filtered partitioning for creating suitable pipeline interventions.

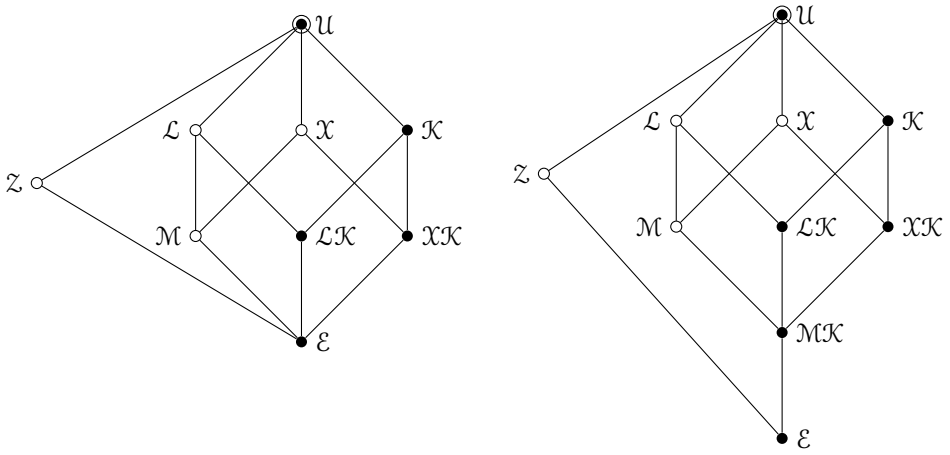
Marginal Affecting a single collection permits the analysis to focus on the effects of the regulation on either system construction or prediction. A suitable intervention would, for instance, fix $\boldsymbol{C}_t$ for every iteration sharing the level $\mathcal{K}(i)$, yet differ in which instances $\boldsymbol{C}_p$ includes depending on whether they are regulated or not

(Fig. 6.5(b)). In other words, the regulated level of $\mathcal{Z}$ would correspond to the combination $(\boldsymbol{C}_t, \boldsymbol{C}'_p)$, and the unregulated to $(\boldsymbol{C}_t, \boldsymbol{C}_p)$. The reverse, fixing $\boldsymbol{C}_p$ and distinguishing conditions only in $\boldsymbol{C}_t$ (Fig. 6.5(c)), would, for example, illuminate the extent to which limiting the information available during system construction benefits generalisation.

Factorial Finally, one could design experiments combining regulations on training and testing in a factorial way, thus creating four associated evaluation conditions (Fig. 6.5(d)). This may reveal potential interactions between conditions.

Note that in the factorial option above, the factor governing the intervention would consist of four different levels, each corresponding to a dot in Fig. 6.5(d). It might be better to express this as two distinct factors, each governing either training or testing, combined factorially (hence the name). In general, a single study might include multiple pipeline interventions simultaneously, as long as their corresponding manipulations are not in conflict. Chapter 5 shows an example of this, including both an instance assignment intervention to regulate artist information and a data manipulation intervention to regulate infrasonic content on an MGR classification experiment with *GTZAN*. This not only illuminates the effect that each individual information source has on the performance estimates, but also their potential interaction.

Altering the raw data, whether through alternative partitioning or directly affecting the contents, is not the only possible approach for introducing suitable pipeline interventions. Any element of the classification experiment pipeline is suitable for manipulation if necessary to target some specific source of information, albeit their use might be less common. For instance, one could design an intervention on the feature extraction function e that removes or masks some specific dimensions (or directly modifies the extracted feature representations used for training and/or testing). It may even be feasible to modify the trained classifier p under certain circumstances, or tweak the annotations $\boldsymbol{A}_t$ and/or $\boldsymbol{A}_p$. The most suitable intervention will depend on the specific information source one wants to affect. In any case, the philosophy remains the same: create additional evaluation conditions through a regulation that reveals whether an information source affects the performance estimates from a benchmark classification experiment.



(a) No Three-way Interaction (b) Three-way Interaction $\mathcal{MK}$

Figure 6.6: Hasse diagrams corresponding to experimental designs for the analysis of measurements from a classification experiment with a single pipeline intervention, ignoring its interactions with other factors in the experiment.

6.2.2 Structural Models for Intervened Classification Experiments

Introducing pipeline interventions in benchmark classification experiments requires revisiting the structural models in Sec. 6.1 to account for the additional evaluation conditions. Pipeline interventions may be introduced in studies targeting any of the analysis levels reviewed previously. To avoid excessive redundancy, the explanation here builds upon the model in Eqn (6.14), since it provides the most detailed and generally suitable analysis level — i.e., estimates from any of the simpler models can be derived from it.

No Interactions with the Intervention Factor As described above, any pipeline intervention can be represented as an additional factor variable in the experiment. Let $\mathcal{Z}$ be one such variable, with Z its number of levels. Since only the differences in measurement between these specific levels are of interest, their effects are considered as fixed. This means the resulting structural model should include a parameter $\tau_{\mathcal{Z}(i)}$. Adding such a parameter to the GBFD in Eqn (6.14), then, would lead to a model such as:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \tau_{\mathcal{M}(i)} + \beta_{\mathcal{XK}(i)} + \beta_{\mathcal{LK}(i)} + \tau_{\mathcal{Z}(i)} + \varepsilon_i \tag{6.17}$$

whose corresponding Hasse diagram is shown in Fig. 6.6(a).

If $\mathcal{Z}$ is the only pipeline intervention in an experiment, then the total number of measurements is $Z \times L \times X \times K$. Using the cascading procedure, the degrees of freedom of the equality factor are, in this case:

$$\begin{aligned} d_{\mathcal{E}} &= N - (d_{\mathcal{U}} + d_{\mathcal{L}} + d_{\mathcal{X}} + d_{\mathcal{K}} + d_{\mathcal{M}} + d_{\mathcal{L}\mathcal{K}} + d_{\mathcal{X}\mathcal{K}} + d_{\mathcal{Z}}) \\ &= Z \cdot L \cdot X \cdot K - (1 + (L-1) + (X-1) + (K-1) + \\ &\qquad (L-1)(X-1) + (L-1)(K-1) + (X-1)(K-1) + (Z-1)) \\ &= Z \cdot L \cdot X \cdot K - (L \cdot X \cdot K - (L-1)(X-1)(K-1) + (Z-1)) \\ &= (Z-1)(L \cdot X \cdot K) + (L-1)(X-1)(K-1) - (Z-1) \\ &= (Z-1)(L \cdot X \cdot K - 1) + (L-1)(X-1)(K-1) \end{aligned}$$

which is positive for any combination of factor sizes greater than 1. Note that, for $Z = 1$, $d_{\mathcal{E}}$ takes the same value as for the GBFD in Eqn (6.14).

As a side effect of the pipeline intervention, then, the overall degrees of freedom of the experiment increase, creating replicates of previously unreachable three-way interactions, such as $\mathcal{M}\mathcal{K}$. This means that one could include in the model a parameter $\beta_{\mathcal{M}\mathcal{K}(i)}$ for the random effect of the interaction between the methods and the samples, such as in the following model:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \tau_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \beta_{\mathcal{X}\mathcal{K}(i)} + \beta_{\mathcal{L}\mathcal{K}(i)} + \beta_{\mathcal{M}\mathcal{K}(i)} + \tau_{\mathcal{Z}(i)} + \varepsilon_i \qquad (6.18)$$

whose corresponding Hasse diagram is represented in Fig. 6.6(b). The degrees of freedom of the equality factor $\mathcal{E}$ are in this case:

$$\begin{aligned} d_{\mathcal{E}} &= N - (d_{\mathcal{U}} + d_{\mathcal{L}} + d_{\mathcal{X}} + d_{\mathcal{K}} + d_{\mathcal{M}} + d_{\mathcal{L}\mathcal{K}} + d_{\mathcal{X}\mathcal{K}} + d_{\mathcal{M}\mathcal{K}} + d_{\mathcal{Z}}) \\ &= Z \cdot L \cdot X \cdot K - (L \cdot X \cdot K - d_{\mathcal{M}\mathcal{K}} + d_{\mathcal{M}\mathcal{K}} + d_{\mathcal{Z}}) \\ &= Z \cdot L \cdot X \cdot K - L \cdot X \cdot K + (Z-1) = (Z-1)(L \cdot X \cdot K) - (Z-1) \\ &= (Z-1)(L \cdot X \cdot K - 1) \end{aligned}$$

which, again, is positive for all factor sizes greater than 1.

Building upon the 2-CV example above may help see how the introduction of a pipeline intervention increases the number of replicates and thus the degrees of freedom. Table 6.2 includes all combinations of factor levels when an intervention with two levels, regulated (z') and unregulated (z), is added to the 2-CV experiment in Table 6.1. The only combination left out is the four-way interaction $\mathcal{Z}\mathcal{M}\mathcal{K}$, since it lacks replicates. The interaction $\mathcal{M}\mathcal{K}$ now has two replicates for each of its levels, thanks to the doubling of the number of observations that the intervention causes.

i	$\mathcal{X}$	$\mathcal{L}$	$\mathcal{K}$	$\mathcal{M}$	$\mathcal{X}\mathcal{K}$	$\mathcal{L}\mathcal{K}$	$\mathcal{M}\mathcal{K}$	$\mathcal{Z}$	$\mathcal{Z}\mathcal{X}$	$\mathcal{Z}\mathcal{L}$	$\mathcal{Z}\mathcal{K}$	$\mathcal{Z}\mathcal{M}$	$\mathcal{Z}\mathcal{X}\mathcal{K}$	$\mathcal{Z}\mathcal{L}\mathcal{K}$
1	e_1	ℓ_1	k_1	$e_1\ell_1$	e_1k_1	ℓ_1k_1	$e_1\ell_1k_1$	z	ze_1	$z\ell_1$	zk_1	$ze_1\ell_1$	ze_1k_1	$z\ell_1k_1$
2	e_1	ℓ_1	k_2	$e_1\ell_1$	e_1k_2	ℓ_1k_2	$e_1\ell_1k_2$	z	ze_1	$z\ell_1$	zk_2	$ze_1\ell_1$	ze_1k_2	$z\ell_1k_2$
3	e_1	ℓ_2	k_1	$e_1\ell_2$	e_1k_1	ℓ_2k_1	$e_1\ell_2k_1$	z	ze_1	$z\ell_2$	zk_1	$ze_1\ell_2$	ze_1k_1	$z\ell_2k_1$
4	e_1	ℓ_2	k_2	$e_1\ell_2$	e_1k_2	ℓ_2k_2	$e_1\ell_2k_2$	z	ze_1	$z\ell_2$	zk_2	$ze_1\ell_2$	ze_1k_2	$z\ell_2k_2$
5	e_2	ℓ_1	k_1	$e_2\ell_1$	e_2k_1	ℓ_1k_1	$e_2\ell_1k_1$	z	ze_2	$z\ell_1$	zk_1	$ze_2\ell_1$	ze_2k_1	$z\ell_1k_1$
6	e_2	ℓ_1	k_2	$e_2\ell_1$	e_2k_2	ℓ_1k_2	$e_2\ell_1k_2$	z	ze_2	$z\ell_1$	zk_2	$ze_2\ell_1$	ze_2k_2	$z\ell_1k_2$
7	e_2	ℓ_2	k_1	$e_2\ell_2$	e_2k_1	ℓ_2k_1	$e_2\ell_2k_1$	z	ze_2	$z\ell_2$	zk_1	$ze_2\ell_2$	ze_2k_1	$z\ell_2k_1$
8	e_2	ℓ_2	k_2	$e_2\ell_2$	e_2k_2	ℓ_2k_2	$e_2\ell_2k_2$	z	ze_2	$z\ell_2$	zk_2	$ze_2\ell_2$	ze_2k_2	$z\ell_2k_2$
9	e_1	ℓ_1	k_1	$e_1\ell_1$	e_1k_1	ℓ_1k_1	$e_1\ell_1k_1$	z'	$z'e_1$	$z'\ell_1$	$z'k_1$	$z'e_1\ell_1$	$z'e_1k_1$	$z'\ell_1k_1$
10	e_1	ℓ_1	k_2	$e_1\ell_1$	e_1k_2	ℓ_1k_2	$e_1\ell_1k_2$	z'	$z'e_1$	$z'\ell_1$	$z'k_2$	$z'e_1\ell_1$	$z'e_1k_2$	$z'\ell_1k_2$
11	e_1	ℓ_2	k_1	$e_1\ell_2$	e_1k_1	ℓ_2k_1	$e_1\ell_2k_1$	z'	$z'e_1$	$z'\ell_2$	$z'k_1$	$z'e_1\ell_2$	$z'e_1k_1$	$z'\ell_2k_1$
12	e_1	ℓ_2	k_2	$e_1\ell_2$	e_1k_2	ℓ_2k_2	$e_1\ell_2k_2$	z'	$z'e_1$	$z'\ell_2$	$z'k_2$	$z'e_1\ell_2$	$z'e_1k_2$	$z'\ell_2k_2$
13	e_2	ℓ_1	k_1	$e_2\ell_1$	e_2k_1	ℓ_1k_1	$e_2\ell_1k_1$	z'	$z'e_2$	$z'\ell_1$	$z'k_1$	$z'e_2\ell_1$	$z'e_2k_1$	$z'\ell_1k_1$
14	e_2	ℓ_1	k_2	$e_2\ell_1$	e_2k_2	ℓ_1k_2	$e_2\ell_1k_2$	z'	$z'e_2$	$z'\ell_1$	$z'k_2$	$z'e_2\ell_1$	$z'e_2k_2$	$z'\ell_1k_2$
15	e_2	ℓ_2	k_1	$e_2\ell_2$	e_2k_1	ℓ_2k_1	$e_2\ell_2k_1$	z'	$z'e_2$	$z'\ell_2$	$z'k_1$	$z'e_2\ell_2$	$z'e_2k_1$	$z'\ell_2k_1$
16	e_2	ℓ_2	k_2	$e_2\ell_2$	e_2k_2	ℓ_2k_2	$e_2\ell_2k_2$	z'	$z'e_2$	$z'\ell_2$	$z'k_2$	$z'e_2\ell_2$	$z'e_2k_2$	$z'\ell_2k_2$

Table 6.2: Factor levels and their interactions in a 2-CV experiment with two feature extractors and two learning algorithms, including a pipeline intervention with two levels.

Interactions with the Intervention Factor As Table 6.2 suggests, introducing an intervention on a classification experiments yields replicates of all two- and three-way interactions with the intervention factor $\mathcal{Z}$. This may provide insights on, for instance, whether the pipeline intervention affects differently each considered system-construction method. Introducing a pipeline intervention in this way does not affect the orthogonality of the design — i.e., each column in the table still follows a different pattern. Only the four-way interaction $\mathcal{Z}\mathcal{M}\mathcal{K}$ (omitted in the table) lacks replicates.

A complete model that accounts for all possible interactions would then be as follows:

$$\begin{aligned} y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \tau_{\mathcal{M}(i)} + \beta_{\mathcal{X}\mathcal{K}(i)} + \beta_{\mathcal{L}\mathcal{K}(i)} + \beta_{\mathcal{M}\mathcal{K}(i)} + \\ \tau_{\mathcal{Z}(i)} + \tau_{\mathcal{Z}\mathcal{X}(i)} + \tau_{\mathcal{Z}\mathcal{L}(i)} + \beta_{\mathcal{Z}\mathcal{K}(i)} + \tau_{\mathcal{Z}\mathcal{M}(i)} + \beta_{\mathcal{Z}\mathcal{X}\mathcal{K}(i)} + \beta_{\mathcal{Z}\mathcal{L}\mathcal{K}(i)} + \varepsilon_i \end{aligned} \tag{6.19}$$

whose corresponding Hasse diagram appears in Fig. 6.7. Since all factor combinations appear, this model is also a GBFD. The calculation of the degrees of freedom follows exactly the same pattern as in the previous cases, such as in Eqn (6.14), thus $d_{\mathcal{E}} = (Z-1)(L-1)(X-1)(K-1)$. Adding the four-way interaction $Z\mathcal{M}\mathcal{K}$ to the analysis would cause the degrees of freedom of the equality factor to be 0.

It may be argued that including parameters for the interactions makes the model excessively convoluted and impractical. Moreover, recall that authors such as Bailey (2008)

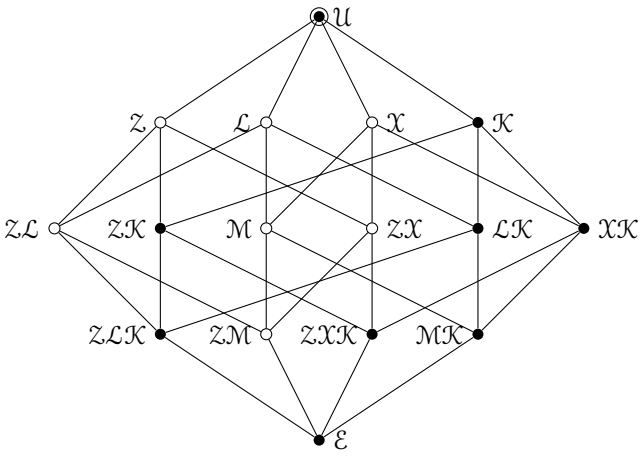


Figure 6.7: Hasse diagram of a generalised blocked factorial design for the analysis of measurements from a classification experiment with a single pipeline intervention.

discourage the use of plot-treatment interactions. In this sense, one could simplify the structural model by ignoring all interactions that involve $\mathcal{K}$, accumulating all their potential effects on a single parameter $\beta_{\mathcal{K}(i)}$. The structural model expressing this situation would be:

$$y_i = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \tau_{\mathcal{M}(i)} + \tau_{\mathcal{Z}(i)} + \tau_{\mathcal{Z}\mathcal{X}(i)} + \tau_{\mathcal{Z}\mathcal{L}(i)} + \tau_{\mathcal{Z}\mathcal{M}(i)} + \varepsilon_i \tag{6.20}$$

which keeps the effects most likely to be of interest in a study. The corresponding factor structure shown in Fig. 6.8 largely resembles the one in Fig. 6.6(b), but swapping the locations of $\mathcal{Z}$ and $\mathcal{K}$, which keeps the plot and treatment factors separate. The degrees of freedom also follow the same pattern, swapping Z with K, so $d_{\mathcal{E}} = (K-1)(L \cdot X \cdot Z - 1)$.

Multiple Interventions As mentioned in Sec. 6.2.1, multiple pipeline interventions can coexist in a single study. Let $\mathcal{W}$ be a second pipeline intervention with W levels. The total number of observations then increases to $Z \times W \times L \times X \times K$, which again permits higher-order interactions to have replicates. However, to avoid excessive complexity of exposition, it will be assumed that the only three-way interactions of interest are those that relate to $\mathcal{M}$. In that case, one could express a structural model such as:

$$\begin{aligned} y_i = \mu &+ \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \tau_{\mathcal{M}(i)} + \tau_{\mathcal{Z}(i)} + \tau_{\mathcal{W}(i)} + \\ &\tau_{\mathcal{Z}\mathcal{X}(i)} + \tau_{\mathcal{Z}\mathcal{L}(i)} + \tau_{\mathcal{Z}\mathcal{M}(i)} + \tau_{\mathcal{W}\mathcal{X}(i)} + \tau_{\mathcal{W}\mathcal{L}(i)} + \tau_{\mathcal{W}\mathcal{M}(i)} + \tau_{\mathcal{Z}\mathcal{W}(i)} + \varepsilon_i. \end{aligned} \tag{6.21}$$

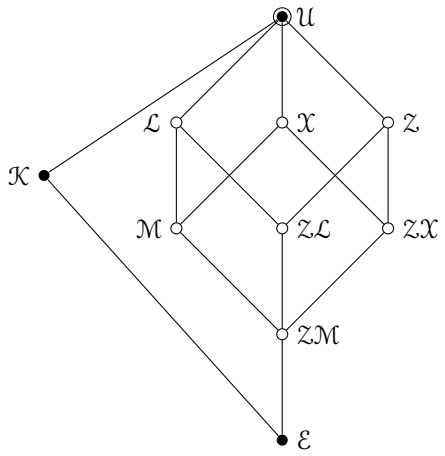


Figure 6.8: Hasse diagram corresponding to an experimental design for the analysis of measurements from a classification experiment with a single pipeline intervention, ignoring all plot-treatment interactions.

This would enable analyses of not only the effect of each intervention separately, but also their interaction with the methods and between them. The same approach could be extended to any number of complementary interventions targeting different information sources, or even the same one but through distinct manipulations.

An obvious alternative to the complex models above would entail focusing solely on the interventions and their mutual interaction, disregarding their effects on the system-constructing methods. This would correspond to a factorial structural model such as:

$$y_i = \mu + \tau_{\mathcal{Z}(i)} + \tau_{\mathcal{W}(i)} + \tau_{\mathcal{Z}\mathcal{W}(i)} + \varepsilon_i \tag{6.22}$$

which could incorporate random effect parameters $\beta_{\mathcal{L}(i)}$, $\beta_{\mathcal{X}(i)}$ and/or $\beta_{\mathcal{K}(i)}$ to account for the variability that such components introduce in the measurements. Analyses of this kind would be suitable if one's sole interest is characterising the effects of potentially confounding information on evaluation results, without checking whether such effects differ across algorithms.

6.3 Logistic Structural Models

The structural models presented so far reflect some implicit assumptions often found in the literature. In particular, they are all linear additive models, which means they assume

that performance estimates can be expressed as the sum of a fixed number of contributions. This, however, ignores that measurements obtained from classification experiments intend to be estimates of probability, either of success or of failure. There is no guarantee that the sum of a number of unconstrained values, such as the ones obtained from estimating the parameters of the model, will be bounded to the range $[0,1]$, as any probability must. This clearly violates the linear nature of such models. This has been noted before, such as by Carterette (2012), who suggests that p-values of performance differences between systems in some Information Retrieval (IR) tasks remain largely unaltered despite the violation. It is not clear the extent to which estimates themselves may be affected, especially when decomposing the measurements into a multitude of potentially contributing factors.

The Item Response Theory (IRT) paradigm mentioned in Sec. 2.5.3 goes a step further. In IRT, each measurement is not an aggregation of individual successes and/or failures, but the individual successes and/or failures themselves. In the context of this chapter, this means that each measurement would not be a summary performance metric over a whole test collection, but instead a binary value $\{0,1\}$ indicating whether a particular prediction matches the expected annotation or not. Directly using the linear models introduced so far in this case would lead to an even clearer violation of the linearity assumption: the sum of some arbitrary real-valued numbers will very unlikely lead to exactly 0 or 1. To overcome this issue, IRT relies on *logistic* models, i.e., models that employ a *logit* transformation of the independent variable, with the *logit* link function being:

$$logit(x) = log\Big(\frac{x}{1-x}\Big) \tag{6.23}$$

where $log(\cdot)$ represents the natural logarithm, and its inverse:

$$logit^{-1}(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}. \tag{6.24}$$

Let u_i be the loss of a particular prediction, $u_i \in \{0,1\}$. The goal is to estimate the contribution of some specific factors, such as the system construction method employed $\mathcal{M}(i)$, to the probability that u_i equals 1. Let $\pi(i)$ be such probability, i.e., $\pi(i) = P(u_i = 1 \mid \mathcal{M}(i), \ldots)$. The *logit* transformation then relates $\pi(i)$ with a linear combination of contributing parameters, such as:

$$logit\big(\pi(i)\big) = \mu + \tau_{\mathcal{M}(i)} \tag{6.25}$$

which follows the convention of removing the residual term ε_i in this type of model, since they explicitly model probabilities and not exact values. Existing software implementations of Generalised Linear Models, such as R's `lme4` package (Bates et al., 2015), provide the necessary functionality to obtain the desired parameter estimates automatically.

One can make logistic structural models as simple or as complex as the situation requires. For instance, in a benchmark classification experiment involving a single pipeline intervention, the model in Eqn (6.20) can be easily converted to:

$$logit\big(\pi(i)\big) = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \tau_{\mathcal{M}(i)} + \tau_{\mathcal{Z}(i)} + \tau_{\mathcal{Z}\mathcal{X}(i)} + \tau_{\mathcal{Z}\mathcal{L}(i)} + \tau_{\mathcal{Z}\mathcal{M}(i)} \tag{6.26}$$

where all parameters are identically expressed as before, but now represent contributions to a *logit*-converted probability of success. All structural models introduced previously would solely require this simple transformation to overcome the violation of linearity. Although the effect estimates may not substantially change, this transformation enhances the statistical rigour of the analysis and thus improves the validity of its conclusions.

A major consequence of increasing the granularity of the observational units on the experiment is that factors that were previously unreachable are now available to be parametrised in structural models to estimate their effects and account for the variability they introduce. Similar to the "difficulty" and "discrimination" latent variables in IRT, this includes instance-specific characteristics as well as other potentially interesting factors, such as the class to which an instance belongs. This can be reflected through the inclusion of a parameter $\beta_{\mathcal{A}(i)}$ in the structural model, with the levels of $\mathcal{A}$ corresponding to the different class annotations in the collection. The variability associated with the distinct classes would then be removed from the estimates of interest. Other factors, such as artists for the case study described in Sec. 5.3, may also be worth considering.

The improvement in both rigour and resolution that logistic models provide obviously comes at a cost. The number and complexity of the computations required to calculate parameter estimates increases, which may complicate analyses on very large collections. Although the procedure to obtain measurements remains the same, since systems predict at instance level regardless of how one estimates performance, the increased number

of data points with which the structural model needs to be fit, as well as the inclusion of the *logit* transformation, hinder the process. Nevertheless, the hardware resources commonly available nowadays should more than suffice in most cases.

A second drawback of logistic structural models is that the parameter estimates lose their straightforward interpretation, so one needs to make additional effort to determine the effects of interest. As an illustration, consider the following toy example. Assume one wants to compare two methods, say m_1 and m_2, using the structural model in Eqn (6.25), with m_1 acting as baseline. Assume as well that predictions from multiple systems built using both methods are obtained, using the measured losses to fit the structural model. Imagine that the estimated parameters are $\hat{\mu} = 0.9$ and $\hat{\tau}_{m_2} = 0.2$. If the model was linear, such as the one in Eqn (6.4), $\hat{\mu}$ would directly correspond to the baseline effect of m_1 (i.e., its estimated mean performance), and $\hat{\tau}_{m_2}$ to the differential effect of m_2 (i.e., its increase or decrease in mean accuracy with respect to m_1). This is obviously not the case here, since these estimates sum far above 1. Instead, it is necessary to reverse the *logit* transformation to obtain estimates of π_{m_1} and π_{m_2}. More precisely, $\hat{\pi}_{m_1} = logit^{-1}(\hat{\mu}) \approx 0.71$ and $\hat{\pi}_{m_2} = logit^{-1}(\hat{\mu} + \hat{\tau}_{m_2}) \approx 0.75$. Nevertheless, the differential effect of m_2 does not correspond to $logit^{-1}(\hat{\tau}_{m_2})$, as one may have expected ($\hat{\pi}_{m_2} - \hat{\pi}_{m_1} \approx 0.04 \neq logit^{-1}(\hat{\tau}_{m_2}) \approx 0.55$). Individual parameter estimates from logistic structural models, such as $\hat{\tau}_{m_2}$ here, do not directly reflect the differential effects of the factors they represent. Careful interpretation of parameter estimates is thus necessary (Agresti, 2002).

Logistic modelling thus offers a valuable resource to model measurements from classification experiments, improving both statistical rigour and resolution. These benefits, however, come at the cost of increased computational demands and harder interpretability of the estimates obtained from fitting the structural models.

6.4 Implications for Intervention-based Evaluation Studies

The insights gained throughout this chapter shed new light on the analysis approaches presented previously in this dissertation. This section thus first revisits the case studies in Ch. 4 and 5 employing the experimental design tools presented in this chapter, and later adapts and extends the procedure in Sec. 5.2.2 to combine both benchmarking and confounding analysis.

6.4.1 Revisiting Case Study Experiments

The case studies in Ch. 4 and 5 provided compelling evidence of confounding and showcased the need for enhanced evaluation practices. The analysis approaches employed there, however, did not take full advantage of the tools presented in this chapter. The discussion here thus considers which improvements could be introduced to achieve a coherent analysis plan, focusing first on the deflation manipulations described in Sec. 4.2 and later on the targeted interventions from both Sec. 4.3 and 5.3.

Deflation Manipulations Strictly speaking, deflation manipulations do not adhere to the comparative experiment framework assumed in this chapter, at least when analysed as in Fig. 4.3 and 4.4. The performance measurements at the final iteration, such as the Final ER values in Table 4.3, could be used to compare methods, but that would not accurately reflect the purpose of the analysis. Alternatively, measurements could be ratios between final and original performance or, if the maximum number of iterations is not fixed, the iteration in which a particular performance threshold is reached. Nevertheless, multiple distinct measurements would be necessary for each method. Since the deflations in Ch. 4 focus on raw data manipulation, and not instance assignment, conventional resampling strategies would suffice to both increase the number of replicates and disentangle method and sample effects. The CBD model in Eqn (6.5) would be suitable in this case. If various combinations of feature extractors and learning algorithms are involved, a BFD such as in Eqn (6.13) (or a GBFD such as in Eqn (6.14) if component-sample interactions are of interest) would suit best.

Since deflations naturally involve several iterations per method, one might be tempted to use such iterations to replace those that resampling generates. In other words, one could use each step in the deflation process as level of a blocking factor taking the role of $\mathcal{K}$. Although this would generate multiple measurements per method, method and sample effects would remain confounded, with conclusions thus being at system level. Nevertheless, for exploratory analyses such as those in Ch. 4, system-level results often suffice to inform further experiments. A more systematic approach could involve considering resampling and deflation iterations as levels from separate blocking factors. Each system of method $m \in \mathcal{M}$ originally trained from the sample corresponding to $k \in \mathcal{K}$ will

undergo all deflation steps, hence yielding a measurement for every level of this hypothetical variable. Orthogonality would thus be preserved. It remains to be seen whether the information that this additional factor provides compensates the increased complexity that it would introduce in the analysis.

The framework presented in this chapter permits scaling the analysis even further and, for instance, including multiple manipulations. This might be useful when system analysis fails to clearly indicate which elements of the input signals are most likely exploited. The levels of a factor $\mathcal{Z}$ could each correspond to a different manipulation. The same structural models as those presented in Sec. 6.2.2 would then hold, with an additional factor for the deflation steps if deemed necessary. To avoid excessive complexity, however, it might be advisable to regard $\mathcal{Z}$ as the only treatment factor, ignoring all possible interactions. Noting the factor variable governing the deflation steps as $\mathcal{Q}$, this would correspond to a structural model such as:

$$y_i = \mu + \beta_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \beta_{\mathcal{Q}(i)} + \tau_{\mathcal{Z}(i)} + \varepsilon_i \tag{6.27}$$

with logistic version:

$$logit\big(\pi(i)\big) = \mu + \beta_{\mathcal{M}(i)} + \beta_{\mathcal{K}(i)} + \beta_{\mathcal{Q}(i)} + \tau_{\mathcal{Z}(i)}. \tag{6.28}$$

Future studies might benefit from comparisons between alternative manipulations in this way during their exploratory phases.

Targeted Interventions Both Ch. 4 and 5 largely focused on the concept of targeted interventions: alterations of the conventional classification experiment pipeline that create regulated and unregulated evaluation conditions according to a particular factor, often a potential confounder of interest. The structure of the corresponding experiments adheres to the models presented in Sec. 6.2. Some of their implementation details, however, may be suboptimal according to the practices illustrated throughout this chapter.

The interventions in Sec. 4.3 serve well as exploratory precursors to more systematic experiments, with a structure that adheres to common orthogonal designs. For instance, the measurements in Table 4.2 reflect a factorial design with $F = 3$ factors: the feature extractor $\mathcal{X}$, with $\mathcal{X} = 6$ levels; the partitioning condition $\mathcal{Z}$, with $Z = 2$ levels; and and the filtering condition $\mathcal{W}$, with $W = 2$ levels. As previously shown, structures of this kind permit

all pairwise interactions without becoming saturated, with $d_{\mathcal{E}} = (\mathcal{X}-1)(Z-1)(W-1) = 5$. Nevertheless, the evaluation conditions associated with the levels in $\mathcal{Z}$ rely on filtered partitioning, which simultaneously alters training and testing materials, impeding disentangling their effects. More importantly, $\mathcal{Z}$ and $\mathcal{C}_t$ (or, equivalently, $\mathcal{C}_p$) are aliased — for every observation i, the elements of $\mathcal{Z}[i]$ coincide with the elements of $\mathcal{C}_t[i]$ (and $\mathcal{C}_p[i]$). The effect of the partitioning condition and that of the particular recordings used for training and testing in each such condition are thus conflated. Since the training and testing collections result from applying the partitioning conditions themselves, their conflation is unlikely to be misleading in an exploratory setting. From a strict DoE perspective, however, the design choices in the targeted interventions of Sec. 4.3 are suboptimal.

The enhancements introduced in Ch. 5 tackle the drawbacks above to make conclusions from targeted interventions statistically valid. The regulated bootstrap resampling strategy is key to achieve this. First, as any resampling strategy, it creates multiple replicates for every combination of method and evaluation condition, which avoids aliasing between $\mathcal{Z}$ and $\mathcal{C}_t$. Instead, $\mathcal{C}_t$ becomes equivalent to the sampling factor $\mathcal{K}$, which is not part of the treatment set. Moreover, the particularities of bootstrap resampling result in two distinct test conditions per training collection (one regulated and another unregulated), meaning $\mathcal{C}_t$ and $\mathcal{C}_p$ are no longer aliased — $\mathcal{C}_p$ becomes the infimum between $\mathcal{C}_t$ and $\mathcal{Z}$ (or, equivalently, between $\mathcal{K}$ and $\mathcal{Z}$).

Although introducing a resampling strategy such as the regulated bootstrap addresses issues with the statistical validity of targeted interventions, their implementation and suggested analysis in Sec. 5.3 could be further enhanced. The regulation in the instance assignment intervention is applied marginally, such as in Fig. 6.5(b). A factorial approach combining regulated and unregulated versions of both training and testing collections would make the most of the limited resources in the study. This would facilitate inferences about the effects of potential confounders, such as artist information, in training as well as testing, similar to what Fig. 5.6 reflects for infrasonic content.

The analysis approach proposed in Sec. 5.2.2(f) provides an intuitive way of assessing confounding effects and their possible interactions. As showcased in this chapter, however, defining a suitable structural model and estimating its associated parameters achieves the same purpose through a unified methodology grounded on well-established

DoE practices. Instead of defining separate ad-hoc formulas to obtain estimates for confounding effects and their interactions, structural models cleanly integrate those within a broader comparative analysis. Benchmarking and confounding analysis can be conducted at once. For the particular case of Sec. 5.3, the model suggested in Eqn (6.21), or its logistic version:

$$\begin{aligned} logit\big(\pi(i)\big) = \mu + \tau_{\mathcal{X}(i)} + \tau_{\mathcal{L}(i)} + \beta_{\mathcal{K}(i)} + \tau_{\mathcal{M}(i)} + \tau_{\mathcal{Z}(i)} + \tau_{\mathcal{W}(i)} + \\ \tau_{\mathcal{Z}\mathcal{X}(i)} + \tau_{\mathcal{Z}\mathcal{L}(i)} + \tau_{\mathcal{Z}\mathcal{M}(i)} + \tau_{\mathcal{W}\mathcal{X}(i)} + \tau_{\mathcal{W}\mathcal{L}(i)} + \tau_{\mathcal{W}\mathcal{M}(i)} + \tau_{\mathcal{Z}\mathcal{W}(i)} \end{aligned} \tag{6.29}$$

would suit analyses similar to the case study presented there. This jointly provides estimates for method and confounding effects, as well as their mutual interactions. If aggregate levels are of interest, such as the "source of feature set" in Fig. 5.9, additional factors nesting current ones can be included. This would discriminate effects arising from a whole group, such as scattering-based extractors, from the particularities of each individual method. As mentioned before, increasing the granularity of the observations could even allow for the inclusion of further parameters, such as class-specific effects. The scope of each study determines the levels of aggregation and detail that should be considered for its analysis, always taking into account the trade-off between how exhaustive and interpretable results become.

6.4.2 Conducting Intervention-based Evaluation Studies

The structural models introduced and discussed in this chapter show that it is possible to combine within a single experiment conventional benchmarking and confounding analysis through interventions. The procedure in Sec. 5.2.2 was presented and exemplified as mainly targeting confounding effects, and some of its implementation details reflected that. As shown above, however, the structure of the derived experiments largely matches the general framework here. This section provides some guidelines on how researchers may apply a similar approach to more conventional evaluation scenarios. Although the structural modelling advocated in this chapter applies mainly within comparative experiments, preliminary analyses are fundamental to decide which factors and with which levels one should include in the experiments. The guidelines thus include the preliminary steps that inform the whole procedure.

Consider the following situation, akin to the work by Andén and Mallat (2014) that the case studies in this dissertation build upon and a myriad other MCA studies. A team of researchers intends to assess the suitability of a novel method they have developed for a problem with existing alternative approaches. They would like to know how their method compares against the alternatives, so they employ for their evaluation a widely adopted evaluation collection. In addition, since they wish their solution to be as generalisable as possible, they would like to determine whether extraneous factors affect the performance of their method. To address these goals, they could proceed as follows.

1. Problem formulation: Identifying goals and challenges

Although rarely considered part of the evaluation, acquiring domain knowledge about the targeted use case is essential to establish the suitability of any method. This includes determining which cues are considered legitimate and which transformations of the input should not affect predictions. Previous publications examining the chosen evaluation collection or related ones may help identifying potential confounders that should be taken into account in both exploratory and comparative analyses.

2. Exploratory analyses: Uncovering reasons behind performance

Before proceeding to more systematic comparisons against alternative approaches, it is advisable to gather as much information as possible about the proposed method's behaviour. Researchers may train one or at most a few systems using their method and attempt to explain the causes of the performance these achieve. As showcased in Ch. 4, diverse strategies serve to this end, with each complementing and informing the rest. In depth system analysis may reveal unexpected potential confounder candidates, as Sec. 4.1 demonstrated, which can then be checked alongside previously identified candidates using brute-force approaches, such as deflations. Other approaches not explored in this dissertation, such as interpretable explanations, can also be used in this step to determine which potential confounders to prioritise. These can be further narrowed down through interventions, each designed to alter the pipeline according to a specific potential confounder candidate. Targeted interventions can include alternative feature extractors and/or learning algorithms to assess the contribution of the original components, but are

best kept simple in this phase since this can be further emphasised through comparative experiments.

3. Comparative experiments: Benchmarking performance and confounding effects

Once the problem formulation and exploratory analyses have illuminated which information sources deserve further attention, researchers can incorporate interventions targeting those sources within a conventional benchmarking experiment. As discussed in this chapter, this first requires deciding which structural model best suits the desired analysis. If the study focuses on both the feature extractor and learning algorithm components of the proposed method, then both their contributions and their interaction should be modelled as fixed-effects parameters. Otherwise, if only one component is of interest, researchers may assume the other introduces random effects.

Regardless of the modelling choice, a wide range of alternative system-constructing methods should be considered. Unlike the procedure suggested in Sec. 5.2.2, which recommended including low-performing methods to cover a range of performance values as wide as possible, benchmarking should focus on methods whose comparisons are of interest. For instance, Andén and Mallat (2014) could have included in their study non-scattering feature extractors and learning algorithms other than SVM, even if their interest lay in the former and not the latter, to reach conclusions generalisable beyond the particular learning algorithm employed.

All combinations of feature extractor and learning algorithm should then be exposed to both regulated and unregulated conditions from each considered intervention for K samples generated from the evaluation collection. If possible, interventions should be applied factorially, which may require using a resampling strategy such as the regulated bootstrap if they include instance assignment interventions. Although using a factorial design permits scaling the experiment to an arbitrary number of interventions, in practice any analysis might become unbearable with more than two potential confounders unless devoted software tools emerge. Therefore, researchers should carefully consider which to include according to preliminary analyses.

Finally, successes or failures in prediction that each trained system makes can be used as performance measurements themselves or aggregated into sample-wise metrics de-

pending on the granularity of the structural model. Researchers can then fit the structural model using these measurements and obtain estimates for each parameter, suggesting the extent to which each factor contributes to the measurements. Appendix B presents an example of how such analysis could be conducted from hypothetical data, which may help readers implement a pipeline to obtain estimates of interest and reach conclusions about them on their own studies.

6.5 Discussion

The myriad structural models presented in this chapter illustrate the wide variety of approaches available for analysing experiments. Each model reflects a particular set of implicit assumptions, potentially leading to disparate conclusions. When planning a study, or when intending to analyse the results of an already conducted one, it may feel overwhelming to choose a particular approach among so many options. Unfortunately, there seems to be no one-size-fits-all solution. The analysis reported in this chapter intends to highlight the tools that researchers have at their disposal to make informed decisions.

The foremost decision that needs to be made is whether the goals of the study demand analyses at system, method and/or component aggregation level. The first option suits pre-deployment analyses, where a single fixed system needs to be selected. Method level may befit evaluation challenges in which organisers have no control over the compatibility between the different components proposed — i.e., each proposed method includes distinct feature extraction and learning algorithms, but it is infeasible to study all their combinations since they have been separately implemented. When a study focuses on a particular component, the latter level should always be the first choice, since it provides the most cost-effective amount of information. This choice informs which factors should be considered on the structural model employed.

It is also important to determine which factors are of interest to the study and which introduce nuisance variability. Modelling undesired effects explicitly as random variables instead of simply averaging them out has the distinct advantage of estimating the variability present in the data, as well as providing more accurate estimates of the effects of interest. Moreover, representing as fixed only the effects of those factors one intends to

compare within the study facilitates reaching valid conclusions about what the measurements actually imply.

It is good practice in statistical modelling to pursue model parsimony — i.e., proposing models with as little complexity as possible. This theoretically favours generalisation, since it introduces fewer restrictions. For the purposes of analysing the measurements from classification experiments, however, it may be argued that providing more conservative estimates of differences between effects can outweigh the benefits of parsimony. In other words, controlling for undesired variability through an exhaustive accounting of multiple potentially contributing factors, and therefore sacrificing parsimony, may yield more generalisable conclusions about differences in performance at the expense of the generalisability of the structural model itself. Such a structural model should be aimed at truthfully capturing the particular settings of the experiment. One should be aware, however, that the number of replications available limits which sources of variation can be included without saturating the model.

Introducing pipeline interventions not only facilitates uncovering reasons behind performance, as exemplified in previous chapters, but also provides additional replicates that enable analysing previously unreachable factors. Properly designed interventions could be used to generate further evaluation conditions similar, but not identical, to the unregulated conventional one to increase the degrees of freedom available for inference. Identical conditions, on the other hand, would not actually increase the degrees of freedom despite the number of observations apparently doubling, since the new ones would be false replications — i.e., the number of real observations would remain the same. Nevertheless, no amount of inferential power would be able to solve the issues in the evaluation of systems and methods that motivate this dissertation. Understanding why systems perform in the way they do arguably trumps introducing further parameters in the structural model if one aims to assess the success of such systems on a particular problem.

A major downside of exhaustive structural models, especially when incorporating interventions, is that they quickly become cumbersome, hampering their practicality. In addition to that, it is important to keep in mind the recommendation of authors such as Bailey (2008) against plot-treatment interactions. Therefore, it might be advisable to adopt simplified models such as the one in Eqn (6.20) unless there are specific reasons to target

the interactions ignored there.

Logistic models might better suit the type of data that classification experiments generate. Their widespread adoption could help improve the validity of the conclusions about the performance of systems and methods. It is important to keep in mind, however, that any adopted model is a simplification. As the aphorism attributed to Box (1976) states, "all models are wrong, but some are useful." A manageable model can often be more useful than a more precise but impractical one, as long as one is aware of the limitations that it introduces in the conclusions.

Applying the tools presented in this chapter to the previously presented case studies reveals possible improvements in their implementation that future studies could mirror. Many of the specific discussions within this chapter, however, arise largely from a frequentist perspective, such as the distinction between fixed and random effects or the calculation of degrees of freedom. The underlying issues, such as which factors to consider in the analysis and whether there are sufficient replicates, would remain pertinent under alternative approaches, such as Bayesian inference. Bayesian evaluation approaches are becoming increasingly popular (e.g., Benavoli et al., 2017), likely due to the controversy surrounding conventional frequentist statistics (Wasserstein and Lazar, 2016). Adapting the analysis presented in this chapter to a Bayesian context, for instance through the use of Multilevel Logistic Models (Sommet and Morselli, 2017), is a promising research avenue.

Part III

Conclusion

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

To conclude this dissertation, Sec. 7.1 overviews the main contributions of the presented research and their overarching implications. Next, Sec. 7.2 proposes future research paths that could be pursued to extend the work here. Finally, Sec. 7.3 provides some closing remarks.

7.1 Summary of Contributions

The research reported in this dissertation has addressed what is arguably the most pressing limitation of classification experiments: systems appearing successful by exploiting supposedly irrelevant information. This issue demands revisiting conventional evaluation practices, which has been tackled here through the incorporation of pipeline interventions and the adoption of principles and tools of the statistical Design of Experiments (DoE). Incorporating targeted interventions into the experimental pipeline in a factorial way facilitates obtaining additional evaluation information with minimal modifications to the conventional procedures.

The study reported in Ch. 4 illustrates the procedure one might undertake to not only understand the reasons behind some particular systems' behaviour but also to identify possible sources of confounding affecting similar evaluations. In-depth system analysis of scattering-based SVM systems highlights details of their implementation that were not evident from their theoretical description, informing potential interventions. Similar to

what other studies had reported before, the known faults of the *GTZAN* genre collection appear to affect the estimated performance of scattering-based systems. Furthermore, such systems appear to exploit previously unknown information at inaudible frequencies to predict genre annotations. Although this revelation arises from an analysis specific to scattering-based systems, the potential confounder that it illuminates can then be probed in more systematic comparisons between system-construction methods, providing information about their vulnerability to such a confounder.

Analyses similar to the one in Ch. 4 serve as exploratory precursors to benchmarking experiments that go beyond counting the number of reproduced annotations. The procedure presented in Ch. 5 relies on such exploratory analyses and domain knowledge to identify potential confounders and integrate interventions that target them. Factorially combining interventions that target various potential confounders simultaneously enables to compare performance of multiple system-construction methods and to assess the effects of those confounders on performance estimates, including possible interactions between confounders. The combination of multiple interventions is possible in part due to the regulated bootstrap resampling strategy, which addresses some of the limitations of filtered partitioning for designing valid data assignment interventions.

The case study included in Ch. 5 illustrates the proposed procedure, training several system-construction methods to assess the effect of artist replication and infrasonic information on performance estimates obtained on *GTZAN*. The results of the study suggest that each considered potential confounder impacts performance results distinctively, although they partially interact. Whereas artist replication appears to affect most performance estimates in a similar manner, inflating test results proportionally, the infrasonic information present on *GTZAN* only affects estimates from particular methods. This means that ignoring some potential confounders during evaluation might lead to overoptimistic expectations about the state-of-the-art solutions to a problem. More worryingly, ignoring other potential confounders could cause the community to prefer specific solutions largely due to them exploiting confounding information that others correctly dismiss. The widespread adoption of an evaluation procedure such as the one proposed would help to avoid such pitfalls.

Regardless of whether classification experiments incorporate pipeline interventions

or not, the use of structural models to express their measurements illuminates the often implicit presumed relationships between contributing factors and facilitates conducting statistically valid analyses. The study presented in Ch. 6 highlights the choices researchers have on which contributions to consider in their analyses, and illustrates the use of the Calculus of Factors approach to determine the suitability of a selected model. Although classification experiments often permit obtaining information about the contributions of multiple factors, excessively complex models consume all degrees of freedom and thus impede inference. Using the cascading procedure for calculating degrees of freedom from Hasse diagrams that is exemplified throughout the chapter helps avoid such an issue.

The use of experimental design tools to decompose performance measurements reveals a further benefit of introducing pipeline interventions in classification experiments beyond estimating confounding effects. The evaluation conditions that interventions add increase the total number of degrees of freedom, enabling estimates of contributions that would otherwise be unreachable. Furthermore, using each individual prediction as observation instead of summary metrics permits including further factors, such as the class or the artist, whose contributions to the overall results might be of interest. Due to the binary nature of the measurements in this case, it is strongly recommended to replace conventional linear models with logistic ones.

Overall, the methodological modifications proposed throughout this dissertation can be viewed as affecting the three phases of most empirical studies: exploration, experimentation and analysis. All three phases are fundamental to obtain valid and relevant evaluation information. Systematically introducing targeted interventions in the classification experiment pipeline following the fundamental principles of experimental design, as presented in this dissertation, is a promising avenue to address the most pressing challenges in the evaluation of Music Content Analysis systems and methods.

7.2 Future Research Directions

The contributions of this dissertation hopefully raise awareness of the pitfalls of conventional evaluation practices in Music Content Analysis and related disciplines, and provide ways forward for the community to adopt. Due to the complexity of the topic, however,

several research paths remain open, a few of which are described in the following paragraphs.

Analyse further systems and collections The illustrative examples presented in this dissertation focused on a particular evaluation collection (*GTZAN*) and largely on a particular family of systems (based on the scattering transform). Several music description problems beyond MGR rely on machine learning solutions and, as demonstrated in the reported studies, failing to understand the reasons behind the success of such solutions might be distorting which ones the community embraces. It is thus fundamental to conduct in the future similar analyses on a wider variety of systems and collections, to better gauge the actual state-of-the-art of the discipline and uncover yet unknown confounding factors.

Create a repository of interventions Identifying potential confounder candidates and implementing suitable pipeline interventions to assess their impact is often complicated and not necessarily rewarding according to common standards in academic research. Developing a system that achieves a higher performance than alternative approaches is more likely to have immediate recognition in the community than understanding the reasons behind such performance. In the long term, however, the community is likely to benefit from a deeper understanding of both the upsides and downsides of any possible approach. To encourage authors to include intervention analyses in their studies, as well as to facilitate the comparison between systems, the community could maintain a collaborative online repository of standardised interventions. Authors who identify potential confounders and implement interventions that target them could contribute to the repository, which would then be reused in future studies addressing similar problems. Some interventions, such as the pitch-preserving time-stretching of Sturm (2016b) or the high-pass filtering here, could be applicable in a wide variety of problems and become standard practice in future publications and evaluation exchanges.

Adapt regulated bootstrap for training One of the major limitations of the regulated bootstrap algorithm in the version introduced in Ch. 5 is that it intervenes solely on testing. This means that studies that adopt it can only assess whether trained systems predict differently when exposed to regulated and unregulated collections of recordings, but not

whether methods construct distinct systems, thus accounting only for half of the story. A modified version of the algorithm that creates the four possible combinations of levels of $\mathcal{C}_t$ and $\mathcal{C}_p$, as shown in Fig. 6.5(d), remains to be implemented. Such implementation would permit not only to assess the impact of potential confounders on training, but also the possible interactions between training and testing.

Incorporate hyperparameter tuning The case study included in Ch. 5 explicitly avoided the optimisation of hyperparameters to ensure that low performance measurements could be achieved. In most practical scenarios, however, authors are interested in comparing methods in their most suitable versions for the target problem. Therefore, benchmarking evaluations with pipeline interventions are likely to include hyperparameter tuning processes for each considered method and data partition. To gain further insights on the consequences of tuning, however, devoted studies could incorporate combinations of hyperparameters as levels of an additional factor variable (nested to the method factor) and explicitly assess how tuning affects the vulnerability of each approach to confounding. In other words, future research could address the question of whether tuning reduces or increases the dependency of systems on the presence of confounding information.

Implement automatic analysis tools The MIR community has at its disposal specialised software libraries such as `mir_eval` (Raffel et al., 2014) to facilitate and standardise evaluation in a variety of scenarios that differ from the conventional case. Such tools, however, are often focused on the calculation of performance estimates and not on the analysis of the resulting measurements. The complexity of the models presented in Ch. 6, particularly when incorporating interventions, may discourage some of their use. The implementation of software tools able to decompose measurements automatically into contributions and assess their effects would likely facilitate the widespread adoption of formal analyses. Großmann (2014) provides an algorithm for the automatic analysis of measurements from orthogonal designs with arbitrary structure, but it is implemented in `Mathematica`,[1] a language with which most researchers in the field would not be familiar,

[1] `https://www.wolfram.com/mathematica/`

and it is not optimised for the subset of structures that are feasible in classification experiments. `mir_eval` and similar libraries could incorporate tools tailored for the analysis of measurements from classification experiments.

Examine non-frequentist analysis approaches Much of the language and tools used throughout this dissertation come from the frequentist tradition of experimental design dominant since the seminal work of Fisher (1935). Recent trends, however, suggest a shift towards alternative approaches, especially in the form of Bayesian inference. Although the main ideas presented in Ch. 6 with regards to the modelling of measurements as the sum of factor effects would hold, some of the specific details, such as the distinction between fixed and random effects, would need to be revisited under a different paradigm. In the future, the community would benefit from devoted studies comparing frequentist and non-frequentist modelling approaches on the validity of the inferences they yield.

Integrate interventions and local explanations Aside from pipeline interventions, the use of evaluation methods based on local explanations has a strong potential to uncover reasons behind performance. They complement interventions by providing interpretable explanations for particular predictions instead of overall changes in performance. This is particularly helpful when attempting to identify potential confounders in the exploratory phase. Nonetheless, to date there have been no attempts to systematically join both perspectives into a unified methodology, which offers opportunities for future research.

Explore the use of IRT latent variables Adapting Item Response Theory (IRT) tools for the evaluation of Machine Learning systems shares, to a large extent, the underlying goals of this dissertation. The use of item-level IRT latent variables, such as difficulty and discrimination, can help uncover reasons behind performance and inform targets for interventions. For instance, systems appearing particularly successful on instances at the high-end of the difficulty scale, or instances with negative discrimination — i.e., on which supposedly poor systems perform better than those with high ability — suggest suspicious behaviours that deserve further analysis. To date, the MIR literature has not yet explored the use of IRT evaluation approaches. Incorporating them into a more comprehensive framework is a promising research avenue.

7.3 Closing Remarks

Evaluation in Music Content Analysis and other applied Machine Learning disciplines may appear to some researchers as a necessary burden, an inescapable formality with which they must comply to see the fruits of hard work finally published. Fortunately for them, the bar is not that high. If a well-established public collection for the problem they intend to tackle does not exist yet, gathering and annotating new data might become laborious; otherwise, the procedure is quite straightforward: they should get hold of evaluation data, chop it into smaller pieces, train their proposed method using some of the pieces, and ask each trained model to predict the annotations of the rest. If, on average, the number of predictions that match the annotations exceeds previously reported values (or is close enough), the likelihood of being accepted for publication increases dramatically; otherwise, they tweak their method and repeat the procedure until finally succeeding. As this dissertation has hopefully shown, however, evaluation is much more than such a necessary burden.

The main goal of evaluation practices is not gatekeeping (or, at least, it should not be). Proper evaluation provides feedback, informing the authors about the virtues and drawbacks of their proposed method. Classification experiments, in their conventional form, provide a single piece of information: how closely predictions match annotations. This feedback suffices for many. Computational disciplines are inherently deterministic, avoiding the uncertainty that hampers physical experiments. The estimates of performance that classification experiments yield can thus appear sufficient to reveal the success of evaluated methods. Examples here and elsewhere show otherwise. Black box models can often rely on unexpected (and undesirable) cues to predict, and blindly trusting that such cues will appear beyond the experimental setting can be extremely detrimental. Understanding why methods work is essential for the discipline to progress and avoid misleading research paths. The evaluation methodology presented in this dissertation aims to achieve precisely that.

The effort that identifying potential confounders, designing targeted interventions and formulating suitable structural models require undeniably exceeds what publications currently demand. Some can see such an effort as counterproductive, stealing time from developing and sharing novel solutions. This cannot be farther from the truth. Devoting

more effort in the short term pays off later, making it easier to improve solutions in the future. Moreover, the methodology proposed here is intended to be highly reusable, with much of the work requiring only minor modifications to be adapted to further studies. Collaborative efforts from the community, including the sharing of intervened pipelines, should facilitate the transition towards a paradigm in which more complex evaluation is not seen as a waste of resources but as the valuable tool that it can become.

In some respects, the approach to evaluation advocated in this dissertation follows the ideas of Popper (1959). Determining whether some lines of code, some bits in a computer, are able to achieve the extraordinary feat of understanding something so complex as a musical concept feels daunting, almost inconceivable. In truth, it is impossible to confirm; there can always be alternative explanations not yet considered. Interventions provide a falsificationist perspective: methods that survive falsification attempts are more likely to be successful. Hopefully, building upon these ideas leads to the development of methods that truly address the challenging and fascinating problems of Music Content Analysis.

Although much of the research reported in this dissertation has focused on Music Content Analysis problems, especially in the illustrative examples employed, nothing about its underlying principles is exclusive to the analysis of music data. Similar issues related to confounding information arise in other applied Machine Learning disciplines (e.g., Chen and Asch, 2017; Nguyen et al., 2015), recently leading to an increased awareness of its risks (e.g., Heaven, 2019; Hernández-Orallo, 2019; Lapuschkin et al., 2019). Apart from the design of specialised interventions, which require domain knowledge, the methodology developed here can prove useful for problems dealing with data other than music audio. Society increasingly relies on Machine Learning systems to break the boundaries of human capabilities. Adopting a systematic evaluation methodology incorporating pipeline interventions will help to understand how these systems behave. Such an understanding is fundamental to gain trust and avoid potentially harmful consequences, so that society can fully embrace the almost unbelievable opportunities that Machine Learning offers for the future.

Appendices

APPENDIX A

ILLUSTRATIVE EXAMPLES OF THE CALCULUS OF FACTORS

This appendix presents some hypothetical examples that might help the reader understand the Calculus of Factors approach to experimental design introduced in Sec. 3.2. Sec. A.1 illustrates the general concepts and methods of the Calculus of Factors, from the definition of factor variables and their relationships to the calculations that Hasse diagrams facilitate, while Sec. A.2 showcases the analysis of conventional designs according to such approach.

A.1 Factors and Subspaces

Factors and their Relationships

Imagine some researchers conduct an experiment with only 8 plots ($N = |\Omega| = 8$). ω indexes the plots, such that $1 \leq \omega \leq 8$. They identify two factors of interest, $\mathcal{F}$ and $\mathcal{G}$, with $N_{\mathcal{F}} = 4$ and $N_{\mathcal{G}} = 2$ respectively (i.e., $\mathcal{F}$ has 4 levels in Ω, while $\mathcal{G}$ has 2). In particular, $\mathcal{F}(\omega) = \lceil \omega/2 \rceil$ and $\mathcal{G}(\omega) = \lceil \omega/4 \rceil$ — i.e., plots 1 and 2 share the same level of $\mathcal{F}$, 3 and 4 another one and so on; $\mathcal{G}$ splits the plot space in half, with plots 1 to 4 sharing one level, and plots 5 to 8 the other. The *equivalence classes* of such factors are $\mathcal{F}(\omega) := \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$ and $\mathcal{G}(\omega) := \{\{1,2,3,4\},\{5,6,7,8\}\}$. All $\mathcal{F}$-classes are contained within $\mathcal{G}$-classes, because every plot contained in each of the classes of $\mathcal{F}$ belongs to the same class of $\mathcal{G}$. For instance,

plots 5 and 6 share the same level of $\mathcal{F}$ (i.e., $\mathcal{F}(5) = \mathcal{F}(6)$), and they also share the same level of $\mathcal{G}$ (i.e., $\mathcal{G}(5) = \mathcal{G}(6)$). The opposite does not hold, though. Plots 5 and 7 share the same level of $\mathcal{G}$ (i.e., $\mathcal{G}(5) = \mathcal{G}(7)$), but differ in $\mathcal{F}$ (i.e., $\mathcal{F}(5) \neq \mathcal{F}(7)$). Therefore, $\mathcal{F}$ is finer than $\mathcal{G}$.

The universal and equality factors of the experiment are quite easy to construct, since their structure is always the same. In particular, $\mathcal{U}(\omega) := \{1,2,3,4,5,6,7,8\}$ — a single class containing all eight plots — and $\mathcal{E}(\omega) := \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$ — one class per plot.

Since $\mathcal{F}$ is finer than $\mathcal{G}$, obtaining their infimum and supremum is immediate. $\mathcal{F} \wedge \mathcal{G}$ is equivalent to the finer factor $\mathcal{F}$, since its classes match the intersection between the $\mathcal{F}$-classes and the $\mathcal{G}$-classes. Take, for instance, the $\mathcal{F}$-class that includes plots 1 and 2. Intersecting it with each of the two $\mathcal{G}$-classes yields $\{1,2\}$ for the one including $\{1,2,3,4\}$, and $\emptyset$ for the other. Repeating this process over all four $\mathcal{F}$-classes confirms that $(\mathcal{F} \wedge \mathcal{G})(\omega) := \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$, so $\mathcal{F} \wedge \mathcal{G} \equiv \mathcal{F}$.

The supremum, on the other hand, is equivalent to the coarser factor $\mathcal{G}$, since its classes completely contain the $\mathcal{F}$-classes. Again, starting with the $\mathcal{F}$-class including plots 1 and 2, its matching class in $\mathcal{F} \vee \mathcal{G}$ also includes these same two plots, as well as any other plot within $\mathcal{G}[1]$ and $\mathcal{G}[2]$. $\mathcal{G}[1]$ and $\mathcal{G}[2]$ coincide and include plots 3 and 4, which should then be added to $(\mathcal{F} \vee \mathcal{G})[1]$. Adding new plots to a set entails returning to the former factor and checking whether it is necessary to include further plots that share an $\mathcal{F}$-class with any of these new ones. In the example, however, all plots in the $\mathcal{F}$-classes including plots 1, 2, 3, or 4 have been already covered, so no other plot needs to be added to $(\mathcal{F} \vee \mathcal{G})[1]$. To complete the classes in the supremum, one would iterate between factors until no further plot remained unassigned. In this case, one would create another class in $\mathcal{F} \vee \mathcal{G}$ containing all remaining plots. This process yields a supremum factor that partitions the plots as $(\mathcal{F} \vee \mathcal{G})(\omega) := \{\{1,2,3,4\},\{5,6,7,8\}\}$, so $\mathcal{F} \vee \mathcal{G} \equiv \mathcal{G}$.

Subspaces defined by Factors

The plot space $V_{\mathcal{E}}$ coincides with $\mathbb{R}^8$, since there are 8 plots in the experiment. $V_{\mathcal{U}}$ includes all vectors constant on all 8 plots, such as $(1,1,1,1,1,1,1,1)$.[1] Recall that the factors $\mathcal{F}$ and $\mathcal{G}$ partition the set of plots such that $\mathcal{F}(\omega) := \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$ and $\mathcal{G}(\omega) := \{\{1,2,3,4\},\{5,6,7,8\}\}$. The vector subspaces these factors define thus consist of:

[1] Parentheses indicate column vectors, so $(1,1,1,1,1,1,1,1) = [1,1,1,1,1,1,1,1]^{\mathrm{T}}$.

$$V_{\mathcal{F}} = span\{(1,1,0,0,0,0,0,0),(0,0,1,1,0,0,0,0),(0,0,0,0,1,1,0,0),(0,0,0,0,0,0,1,1)\}$$
$$V_{\mathcal{G}} = span\{(1,1,1,1,0,0,0,0),(0,0,0,0,1,1,1,1)\}$$

— i.e., as many vectors as classes, each one constant on the components associated with the plots in a class. Vectors in $V_{\mathcal{G}}$ can be expressed as a linear combination of the vectors generating $V_{\mathcal{F}}$, implying that $V_{\mathcal{G}}$ is a subspace of $V_{\mathcal{F}}$. Moreover, the intersection of both subspaces coincides with $V_{\mathcal{G}}$, so $V_{\mathcal{F}\vee\mathcal{G}} = V_{\mathcal{G}}$.

If two factors are not nested, finding the intersection between their subspaces becomes less trivial. For instance, consider the factor $\mathcal{H}$ over the same plots, such that $\mathcal{H}(\omega) := \{\{1,2,3\},\{4,5,6\},\{7,8\}\}$. Its associated subspace is:

$$V_{\mathcal{H}} = span\{(1,1,1,0,0,0,0,0),(0,0,0,1,1,1,0,0),(0,0,0,0,0,0,1,1)\}.$$

The subspace corresponding to the supremum of $\mathcal{F}$ and $\mathcal{H}$ is:

$$V_{\mathcal{F}\vee\mathcal{H}} = V_{\mathcal{F}} \cap V_{\mathcal{H}} = span\{(1,1,1,1,1,1,0,0),(0,0,0,0,0,0,1,1)\}$$

which has $dim(V_{\mathcal{F}\vee\mathcal{H}}) = 2$ and differs from any subspace considered so far.

The relation and projection matrices associated with factor $\mathcal{F}$ are:

$$\mathbf{R}_{\mathcal{F}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{P}_{\mathcal{F}} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}.$$

Since $\mathcal{F}$ is uniform with $K_{\mathcal{F}} = 2$, the relationship $\mathbf{R}_{\mathcal{F}} = K_{\mathcal{F}}\mathbf{P}_{\mathcal{F}} = 2\mathbf{P}_{\mathcal{F}}$ holds; this would not be the case for factor $\mathcal{H}$, for instance. For any arbitrary vector in $\mathbb{R}^{|\Omega|}$, such as $\mathbf{y} = (1,8,2,4,5,-4,2,8)$, applying the linear transformation defined by $\mathbf{P}_{\mathcal{F}}$ generates its projection into the subspace $V_{\mathcal{F}}$:

$$\mathbf{P}_{\mathcal{F}}(\mathbf{y}) = \mathbf{P}_{\mathcal{F}}\mathbf{y} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1 \\ 8 \\ 2 \\ 4 \\ 5 \\ -4 \\ 2 \\ 8 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 4.5 \\ 3 \\ 3 \\ 0.5 \\ 0.5 \\ 5 \\ 5 \end{bmatrix}.$$

Factor Orthogonality

Definition from Subspaces Since the factors $\mathcal{F}$ and $\mathcal{G}$ in the example form a chain, with $\mathcal{F} \prec \mathcal{G}$, their orthogonality is guaranteed. The product as in Eqn (3.15) confirms this. The matrices $\mathbf{P}_\mathcal{F}$ and $\mathbf{P}_\mathcal{G}$ are:

$$\mathbf{P}_\mathcal{F} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}, \mathbf{P}_\mathcal{G} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

with their product yielding:

$$\mathbf{P}_\mathcal{F}\mathbf{P}_\mathcal{G} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix} = \mathbf{P}_\mathcal{G}\mathbf{P}_\mathcal{F} = \mathbf{P}_{\mathcal{F}\vee\mathcal{G}} = \mathbf{P}_\mathcal{G}.$$

Exactly the same result is obtained regardless of the order in which one performs the product. Moreover, the result of the product matches $\mathbf{P}_\mathcal{G}$, the projection matrix associated with the subspace $V_{\mathcal{F}\vee\mathcal{G}}$.

Conversely, the factor $\mathcal{H}$ defined above does not form a chain with any other factor. Its projection matrix $\mathbf{P}_\mathcal{H}$ is:

$$\mathbf{P}_\mathcal{H} = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

whose two products with $\mathbf{P}_\mathcal{F}$ are:

$$\mathbf{P}_{\mathcal{F}}\mathbf{P}_{\mathcal{H}} = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}, \mathbf{P}_{\mathcal{H}}\mathbf{P}_{\mathcal{F}} = \begin{bmatrix} 1/3 & 1/3 & 1/6 & 1/6 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/6 & 1/6 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/6 & 1/6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/6 & 1/6 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}.$$

These products yield different results, implying that that $\mathcal{F}$ and $\mathcal{H}$ are not orthogonal. $\mathcal{G}$ leads to a similar conclusion.

Factors not forming chains can also be mutually orthogonal. Consider an additional factor $\mathcal{K}$ such that $\mathcal{K}(\omega) := \{\{1,3\},\{2,4\},\{5,7\},\{6,8\}\}$, with projection matrix:

$$\mathbf{P}_{\mathcal{K}} = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{bmatrix}.$$

Although $\mathcal{K}$ is neither finer nor coarser than $\mathcal{F}$, the product of their projection matrices is commutative and its result coincides with $\mathbf{P}_{\mathcal{G}}$:

$$\mathbf{P}_{\mathcal{F}}\mathbf{P}_{\mathcal{K}} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix} = \mathbf{P}_{\mathcal{K}}\mathbf{P}_{\mathcal{F}} = \mathbf{P}_{\mathcal{F}\vee\mathcal{K}} = \mathbf{P}_{\mathcal{G}}.$$

$\mathcal{F}$ and $\mathcal{K}$ are thus mutually orthogonal despite none being nested to the other. Moreover, they are both finer than $\mathcal{G}$, which is their supremum, so $\mathcal{K}$ is also orthogonal to $\mathcal{G}$.

Definition from Classes The definition in Eqn (3.17) is quite obscure, so its application to the example may clarify how it can be leveraged. Both $\mathcal{F}$ and $\mathcal{K}$ are uniform with $K_{\mathcal{F}} = K_{\mathcal{K}} = 2$, so $|\mathcal{F}[\omega]| = |\mathcal{K}[\omega]| = 2\ \forall\omega$. Their supremum is $\mathcal{G}$, which is also uniform with $K_{\mathcal{G}} = 4$, so $|(\mathcal{F}\vee\mathcal{K})[\omega]| = 4\ \forall\omega$. Only information about the infimum is missing. Since $\mathcal{F}$- and $\mathcal{K}$-classes share only one element each, the largest intersections between classes one can construct are the eight subsets containing a single plot. $\mathcal{F}\wedge\mathcal{K}$ is thus the equality

factor $\mathcal{E}$, so $|(\mathcal{F} \wedge \mathcal{K})[\omega]| = 1\ \forall\omega$. Since all four components of Eqn (3.17) are uniform, one only needs to check whether the equality holds once. $|\mathcal{F}[\omega]| \times |\mathcal{K}[\omega]| = 2 \times 2 = 4$ matches $|(\mathcal{F} \wedge \mathcal{K})[\omega]| \times |(\mathcal{F} \vee \mathcal{K})[\omega]| = 1 \times 4 = 4$ for any plot within any of the two $\mathcal{G}$-classes. Therefore, $\mathcal{F}$ and $\mathcal{K}$ are orthogonal according to the alternative definition of orthogonality.

Factors $\mathcal{F}$ and $\mathcal{H}$ are not mutually orthogonal, so one might wonder what the definition from classes yields in that case. Recall that $\mathcal{H}(\omega) := \{\{1,2,3\},\{4,5,6\},\{7,8\}\}$, which is evidently not uniform. This means it is necessary to distinguish the size of each $\mathcal{H}$-class with respect to ω:

$$|\mathcal{H}(\omega)| = \begin{cases} 3 & \forall\omega, 1 \leq \omega \leq 6 \\ 2 & \text{if } \omega = 7,8 \end{cases}$$

Computing the infimum and supremum factors of $\mathcal{F}$ and $\mathcal{H}$ is not as simple as previously, but following the procedure suggested in Sec. 3.2.1 generates $(\mathcal{F} \wedge \mathcal{H})(\omega) := \{\{1,2\},\{3\},\{4\},\{5,6\},\{7,8\}\}$, and $(\mathcal{F} \vee \mathcal{H})(\omega) := \{\{1,2,3,4,5,6\},\{7,8\}\}$. Neither the infimum nor the supremum are uniform, so it is again necessary to distinguish the sizes of their classes depending on ω:

$$|(\mathcal{F} \wedge \mathcal{H})(\omega)| = \begin{cases} 2 & \text{if } \omega = 1,2,5,6,7,8 \\ 1 & \text{if } \omega = 3,4 \end{cases}$$

$$|(\mathcal{F} \vee \mathcal{H})(\omega)| = \begin{cases} 6 & \forall\omega, 1 \leq \omega \leq 6 \\ 2 & \text{if } \omega = 7,8 \end{cases}$$

Consider the class of the supremum containing plots 1 to 6. For the factors to be orthogonal it is required that the proportionality ratio remains constant for any plot belonging to this class. For instance, for $\omega = 1$, $|\mathcal{F}[\omega]| \times |\mathcal{H}[\omega]| = 2 \times 3 = 6$ and $|(\mathcal{F} \wedge \mathcal{H})[\omega]| \times |(\mathcal{F} \vee \mathcal{H})[\omega]| = 2 \times 6 = 12$. The proportionality ratio for class $j = 1$ obtained from plot 1 is then $c_1 = 2$. For $\omega = 3$, on the other hand, $|\mathcal{F}[\omega]| \times |\mathcal{H}[\omega]| = 2 \times 3 = 6$ and $|(\mathcal{F} \wedge \mathcal{H})[\omega]| \times |(\mathcal{F} \vee \mathcal{H})[\omega]| = 1 \times 6 = 6$, so $c_1 = 1$ instead. This mismatch suffices to ensure that factors $\mathcal{F}$ and $\mathcal{H}$ are not mutually orthogonal.

Non-orthogonality

Imagine factor $\mathcal{H}$ represents the height of the participants in a study, and its three classes group `short` ($\{1,2,3\}$), `medium` ($\{4,5,6\}$), and `tall` ($\{7,8\}$) individuals, respectively. Realising this split causes the factors in the experiment to become non-orthogonal, researchers could examine the heights in their original continuous scale, finding that the individuals

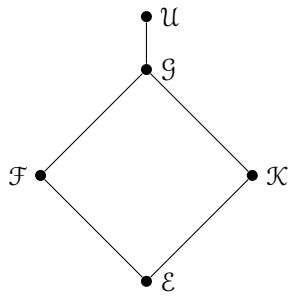


Figure A.1: Hasse diagram showing relationships between factors in the example.

are ordered, from shortest to tallest, in the following way: $1,3,2,4,6,5,8,7$. Many alternative ways exist to set level boundaries, with some of them allowing $\mathcal{H}$ to become orthogonal to all other factors in the experiment. Fixing a single boundary between plots 4 and 6, for instance, would split all individuals into two categories: `short` and `tall`. Regardless of which concepts they represent, however, this splitting forces $\mathcal{H} \equiv \mathcal{G}$ in the experiment. Setting boundaries after plots 3, 4, and 5, on the other hand, would make $\mathcal{H}$ orthogonal with all other factors without aliasing. In what follows, however, it is assumed that researchers have decided to ignore $\mathcal{H}$ in their analysis.

Orthogonal Decomposition

Let $\mathscr{F}$ be a set that includes all factors in the experiment other than the non-orthogonal $\mathcal{H}$. Figure A.1 shows the relationships between such factors using a Hasse diagram, with factors coarser than another placed above and connected to that other one. No factor is thus coarser than $\mathcal{U}$, only $\mathcal{U}$ is coarser than $\mathcal{G}$, both $\mathcal{U}$ and $\mathcal{G}$ are coarser than $\mathcal{F}$ and $\mathcal{K}$, and all other factors are coarser than $\mathcal{E}$. Using the definition in Eqn (3.18), the following calculations then yield the W-subspaces:

$$
\begin{aligned}
W_{\mathcal{U}} &= V_{\mathcal{U}} \\
W_{\mathcal{G}} &= V_{\mathcal{G}} \cap V_{\mathcal{U}}^{\perp} \\
W_{\mathcal{F}} &= V_{\mathcal{F}} \cap (V_{\mathcal{U}}^{\perp} \cap V_{\mathcal{G}}^{\perp}) \\
W_{\mathcal{K}} &= V_{\mathcal{K}} \cap (V_{\mathcal{U}}^{\perp} \cap V_{\mathcal{G}}^{\perp}) \\
W_{\mathcal{E}} &= V_{\mathcal{K}} \cap (V_{\mathcal{U}}^{\perp} \cap V_{\mathcal{G}}^{\perp} \cap V_{\mathcal{F}}^{\perp} \cap V_{\mathcal{K}}^{\perp})
\end{aligned}
$$

Recall the V-subspaces associated with each factor are obtained through:

$$
\begin{aligned}
V_{\mathcal{U}} &= span\{(1,1,1,1,1,1,1,1)\}\\
V_{\mathcal{G}} &= span\{(1,1,1,1,0,0,0,0),(0,0,0,0,1,1,1,1)\}\\
V_{\mathcal{F}} &= span\{(1,1,0,0,0,0,0,0),(0,0,1,1,0,0,0,0),(0,0,0,0,1,1,0,0),(0,0,0,0,0,0,1,1)\}\\
V_{\mathcal{K}} &= span\{(1,0,1,0,0,0,0,0),(0,1,0,1,0,0,0,0),(0,0,0,0,1,0,1,0),(0,0,0,0,0,1,0,1)\}\\
V_{\mathcal{E}} &= span\{(1,0,0,0,0,0,0,0),(0,1,0,0,0,0,0,0),(0,0,1,0,0,0,0,0),(0,0,0,1,0,0,0,0),\\
&\qquad (0,0,0,0,1,0,0,0),(0,0,0,0,0,1,0,0),(0,0,0,0,0,0,1,0),(0,0,0,0,0,0,0,1)\}
\end{aligned}
$$

Computing the W-subspaces requires the orthogonal complement vector spaces to $V_{\mathcal{U}}$, $V_{\mathcal{G}}$, $V_{\mathcal{F}}$, and $V_{\mathcal{K}}$.[2] The orthogonal complement of $V_{\mathcal{E}}$ is not necessary since, by definition, $\mathcal{E}$ is finer than every other factor in the experiment.

$$
\begin{aligned}
V_{\mathcal{U}}^{\perp} &= span\{(-1,1,0,0,0,0,0,0),(-1,0,1,0,0,0,0,0),(-1,0,0,1,0,0,0,0),(-1,0,0,0,1,0,0,0),\\
&\qquad (-1,0,0,0,0,1,0,0),(-1,0,0,0,0,0,1,0),(-1,0,0,0,0,0,0,1)\}\\
V_{\mathcal{G}}^{\perp} &= span\{(-1,1,0,0,0,0,0,0),(-1,0,1,0,0,0,0,0),(-1,0,0,1,0,0,0,0),(0,0,0,0,-1,1,0,0),\\
&\qquad (0,0,0,0,-1,0,1,0),(0,0,0,0,-1,0,0,1)\}\\
V_{\mathcal{F}}^{\perp} &= span\{(-1,1,0,0,0,0,0,0),(0,0,-1,1,0,0,0,0),(0,0,0,0,-1,1,0,0),(0,0,0,0,0,0,-1,1)\}\\
V_{\mathcal{K}}^{\perp} &= span\{(-1,0,1,0,0,0,0,0),(0,-1,0,1,0,0,0,0),(0,0,0,0,-1,0,1,0),(0,0,0,0,0,-1,0,1)\}
\end{aligned}
$$

The following intersections are necessary:[3]

$$
\begin{aligned}
V_{\mathcal{U}}^{\perp} \cap V_{\mathcal{G}}^{\perp} &= span\{(1,0,0,-1,0,0,0,0),(0,1,0,-1,0,0,0,0),(0,0,1,-1,0,0,0,0),\\
&\qquad (0,0,0,0,1,0,0,-1),(0,0,0,0,0,1,0,-1),(0,0,0,0,0,0,1,-1)\}\\
V_{\mathcal{U}}^{\perp} \cap V_{\mathcal{G}}^{\perp} \cap V_{\mathcal{F}}^{\perp} \cap V_{\mathcal{K}}^{\perp} &= span\{(1,-1,-1,1,0,0,0,0),(0,0,0,0,1,-1,-1,1)\}
\end{aligned}
$$

These complete the necessary pieces to calculate the W-subspaces:

$$
\begin{aligned}
W_{\mathcal{U}} &= span\{(1,1,1,1,1,1,1,1)\}\\
W_{\mathcal{G}} &= span\{(1,1,1,1,-1,-1,-1,-1)\}\\
W_{\mathcal{F}} &= span\{(1,1,-1,-1,0,0,0,0),(0,0,0,0,1,1,-1,-1)\}\\
W_{\mathcal{K}} &= span\{(1,-1,1,-1,0,0,0,0),(0,0,0,0,1,-1,1,-1)\}\\
W_{\mathcal{E}} &= span\{(1,-1,-1,1,0,0,0,0),(0,0,0,0,1,-1,-1,1)\}
\end{aligned}
$$

The factors considered in the experiment satisfy both conditions for orthogonal decomposition: they are all mutually orthogonal and their pairwise supremums belong to

[2] The orthogonal complement of the associated subspaces is computed using `Matlab`'s native `null` function (`http://uk.mathworks.com/help/matlab/ref/null.html`), with the `'r'` option enabled to obtain a "rational" basis suitable for pedagogical purposes.

[3] The function `findIntersect` (downloaded from `https://www.mathworks.com/matlabcentral/fileexchange/32060-intersection-of-linear-subspaces/content/findIntersect.m`) is used to intersect linear subspaces.

	$\mathcal{E}$			
$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$		
$\mathcal{K}$	$\mathcal{K}$	$\mathcal{G}$	$\mathcal{K}$	
$\mathcal{G}$	$\mathcal{G}$	$\mathcal{G}$	$\mathcal{G}$	$\mathcal{G}$
$\mathcal{U}$	$\mathcal{U}$	$\mathcal{U}$	$\mathcal{U}$	$\mathcal{U}$

Table A.1: Supremum factors of all possible combinations of non-equivalent factors in our example.

the set $\mathscr{F}$ (see Table A.1). The implications (i) and (ii) from those conditions mentioned in Sec. 3.2.4 thus hold:

(i) Two vector spaces are mutually orthogonal if the scalar product of any vector from one with any from the other is zero. To check if two W-subspaces are orthogonal, then, one can multiply the matrices formed by their bases; a resulting matrix with zeroes in all cells entails the subspaces are indeed orthogonal. For instance, in the case of $\mathcal{F}$ and $\mathcal{K}$:

$$\mathbf{W}_{\mathcal{F}}^{\mathrm{T}}\mathbf{W}_{\mathcal{K}} = \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

which ensures their orthogonality. Any other combination of W-subspaces yields a similar result.

(ii) The direct sum of orthogonal subspaces can be obtained by spanning the union of their basis, such as:

$$W_{\mathcal{G}} \oplus W_{\mathcal{F}} = span\{(1,1,1,1,-1,-1,-1,-1),(1,1,1,1,1,1,1,1)\}.$$

This subspace should coincide with $V_{\mathcal{G}}$, since $\mathcal{G}$ and $\mathcal{U}$ are the factors coarser or equivalent to $\mathcal{G}$ in the experiment. The vectors that form the bases just obtained, however, differ from those presented previously. On the other hand, the reduced echelon form[4] of both bases coincides. More precisely,

$$\begin{aligned} W_{\mathcal{G}} \oplus W_{\mathcal{F}} &= span\{(1,1,1,1,-1,-1,-1,-1),(1,1,1,1,1,1,1,1)\} \\ &= span\{(1,1,1,1,0,0,0,0),(0,0,0,0,1,1,1,1)\} = V_{\mathcal{G}}. \end{aligned}$$

[4] Matlab provides the function `rref` (`http://uk.mathworks.com/help/matlab/ref/rref.html`) for this purpose.

The remaining direct sums also match the associated vectors. The direct sum of all W-subspaces yields:

$$\begin{aligned} W_{\mathcal{E}} \oplus W_{\mathcal{F}} \oplus W_{\mathcal{K}} \oplus W_{\mathcal{G}} \oplus W_{\mathcal{U}} = span\{&(1,-1,-1,1,0,0,0,0),(0,0,0,0,1,-1,-1,1),\\ &(1,1,-1,-1,0,0,0,0),(0,0,0,0,1,1,-1,-1),\\ &(1,-1,1,-1,0,0,0,0),(0,0,0,0,1,-1,1,-1),\\ &(1,1,1,1,-1,-1,-1,-1),(1,1,1,1,1,1,1,1)\} \end{aligned}$$

whose reduced echelon form matches the canonical basis of $\mathbb{R}^8$.

The projection matrix of the W-subspace associated with $\mathcal{F}$ can be obtained as follows:

$$(\mathbf{W}_{\mathcal{F}}^{\mathrm{T}}\mathbf{W}_{\mathcal{F}})^{-1} = \left(\begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ -1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix} \right)^{-1} = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix}$$

$$\mathbf{P}_{W_{\mathcal{F}}} = \mathbf{W}_{\mathcal{F}}(\mathbf{W}_{\mathcal{F}}^{\mathrm{T}}\mathbf{W}_{\mathcal{F}})^{-1}\mathbf{W}_{\mathcal{F}}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ -1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1/4 & 1/4 & -1/4 & -1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & -1/4 & -1/4 & 0 & 0 & 0 & 0 \\ -1/4 & -1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ -1/4 & -1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & -1/4 & -1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & -1/4 & -1/4 \\ 0 & 0 & 0 & 0 & -1/4 & -1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & -1/4 & -1/4 & 1/4 & 1/4 \end{bmatrix}.$$

Recall the data vector that the researchers obtain is $\mathbf{y} = (1,8,2,4,5,-4,2,8)$, whose orthogonal projection onto $V_{\mathcal{F}}$ is $\mathbf{P}_{V_{\mathcal{F}}}\mathbf{y} = (4.5,4.5,3,3,0.5,0.5,5,5)$. Hence, the projection of $\mathbf{y}$ onto the $W_{\mathcal{F}}$ subspace is:

$$\mathbf{P}_{W_{\mathcal{F}}}\mathbf{y} = \begin{bmatrix} 1/4 & 1/4 & -1/4 & -1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & -1/4 & -1/4 & 0 & 0 & 0 & 0 \\ -1/4 & -1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ -1/4 & -1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & -1/4 & -1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & -1/4 & -1/4 \\ 0 & 0 & 0 & 0 & -1/4 & -1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & -1/4 & -1/4 & 1/4 & 1/4 \end{bmatrix} \begin{bmatrix} 1 \\ 8 \\ 2 \\ 4 \\ 5 \\ -4 \\ 2 \\ 8 \end{bmatrix} = \begin{bmatrix} 0.75 \\ 0.75 \\ -0.75 \\ -0.75 \\ 2.25 \\ 2.25 \\ -2.25 \\ -2.25 \end{bmatrix}.$$

Similarly, one can project $\mathbf{y}$ onto the remaining W:

$$\begin{aligned} \mathbf{P}_{W_{\mathcal{U}}}\mathbf{y} &= (3.25, 3.25, 3.25, 3.25, 3.25, 3.25, 3.25, 3.25) \\ \mathbf{P}_{W_{\mathcal{G}}}\mathbf{y} &= (0.5, 0.5, 0.5, 0.5, -0.5, -0.5, -0.5, -0.5) \\ \mathbf{P}_{W_{\mathcal{K}}}\mathbf{y} &= (-2.25, 2.25, -2.25, 2.25, 0.75, -0.75, 0.75, -0.75) \\ \mathbf{P}_{W_{\mathcal{E}}}\mathbf{y} &= (-1.25, 1.25, 1.25, -1.25, 3.75, -3.75, -3.75, 3.75) \end{aligned}$$

Computing all pairwise scalar products between the projection vectors confirms all are mutually orthogonal. Moreover, summing all projections into W-subspaces associated with factors coarser or equivalent to one particular factor yields the projection into the V-subspace of that same factor. For instance, for $\mathcal{F}$:

$$\mathbf{P}_{\mathbf{W}_{\mathcal{F}}}\mathbf{y} + \mathbf{P}_{\mathbf{W}_{\mathcal{G}}}\mathbf{y} + \mathbf{P}_{\mathbf{W}_{\mathcal{U}}}\mathbf{y} = \begin{bmatrix} 0.75 \\ 0.75 \\ -0.75 \\ -0.75 \\ 2.25 \\ 2.25 \\ -2.25 \\ -2.25 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 4.5 \\ 3 \\ 3 \\ 0.5 \\ 0.5 \\ 5 \\ 5 \end{bmatrix} = \mathbf{P}_{V_{\mathcal{F}}}\mathbf{y}$$

Summing the projections onto all W-subspaces generates:

$$\mathbf{P}_{W_{\mathcal{E}}}\mathbf{y} + \mathbf{P}_{W_{\mathcal{F}}}\mathbf{y} + \mathbf{P}_{W_{\mathcal{K}}}\mathbf{y} + \mathbf{P}_{W_{\mathcal{G}}}\mathbf{y} + \mathbf{P}_{W_{\mathcal{U}}}\mathbf{y} = \begin{bmatrix} -1.25 \\ 1.25 \\ 1.25 \\ -1.25 \\ 3.75 \\ -3.75 \\ -3.75 \\ 3.75 \end{bmatrix} + \begin{bmatrix} 0.75 \\ 0.75 \\ -0.75 \\ -0.75 \\ 2.25 \\ 2.25 \\ -2.25 \\ -2.25 \end{bmatrix} + \begin{bmatrix} -2.25 \\ 2.25 \\ -2.25 \\ 2.25 \\ 0.75 \\ -0.75 \\ 0.75 \\ -0.75 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \\ 3.25 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 2 \\ 4 \\ 5 \\ -4 \\ 2 \\ 8 \end{bmatrix}$$

which matches the original data vector $\mathbf{y}$. The projections onto W-subspaces thus decompose measurements into contributions for each factor.

Connection with ANOVA The results obtained previously lead directly to both the degrees of freedom and the sum of squares for each factor. Counting the number of vectors

in the basis of each W-subspace, then:

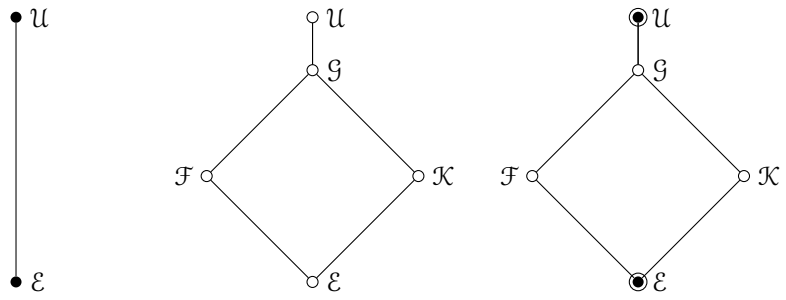
$$
\begin{aligned}
d_{\mathcal{U}} &= dim(W_{\mathcal{U}}) = 1 \\
d_{\mathcal{G}} &= dim(W_{\mathcal{G}}) = 1 \\
d_{\mathcal{F}} &= dim(W_{\mathcal{F}}) = 2 \\
d_{\mathcal{K}} &= dim(W_{\mathcal{K}}) = 2 \\
d_{\mathcal{E}} &= dim(W_{\mathcal{E}}) = 2
\end{aligned}
$$

and the projections of $\mathbf{y}$ onto each W-subspace yield:

$$
\begin{aligned}
SS_{\mathcal{U}}(\mathbf{y}) &= \left\|\mathbf{P}_{W_{\mathcal{U}}}\mathbf{y}\right\|^2 = 84.5 \\
SS_{\mathcal{G}}(\mathbf{y}) &= \left\|\mathbf{P}_{W_{\mathcal{G}}}\mathbf{y}\right\|^2 = 2 \\
SS_{\mathcal{F}}(\mathbf{y}) &= \left\|\mathbf{P}_{W_{\mathcal{F}}}\mathbf{y}\right\|^2 = 22.5 \\
SS_{\mathcal{K}}(\mathbf{y}) &= \left\|\mathbf{P}_{W_{\mathcal{K}}}\mathbf{y}\right\|^2 = 22.5 \\
SS_{\mathcal{E}}(\mathbf{y}) &= \left\|\mathbf{P}_{W_{\mathcal{E}}}\mathbf{y}\right\|^2 = 62.5
\end{aligned}
$$

Calculations on the Hasse Diagram

For the sake of simplicity, plot and treatment factors have not been distinguished so far, grouping them all together in a single set. Assume now that the plots are *unstructured* — only $\mathcal{U}$ and $\mathcal{E}$ relate with the plot structure —, and $\mathcal{G}$, $\mathcal{F}$, and $\mathcal{K}$ describe the treatments. Figure. A.2 shows the resulting Hasse diagrams.



(a) Plot structure (b) Treatment structure (c) Combined structure

Figure A.2: Plot, treatment and combined factor structures in the example.

The number of classes of each factor are $N_{\mathcal{U}} = 1$, $N_{\mathcal{G}} = 2$, $N_{\mathcal{F}} = 4$, $N_{\mathcal{K}} = 4$, and $N_{\mathcal{E}} = 8$. Starting from the top of the diagram, one can then calculate the degrees of freedom for each factor in cascade:

$$
\begin{aligned}
d_{\mathcal{U}} &= N_{\mathcal{U}} = 1 \\
d_{\mathcal{G}} &= N_G - d_{\mathcal{U}} = 2 - 1 = 1 \\
d_{\mathcal{F}} &= N_F - (d_{\mathcal{U}} + d_{\mathcal{G}}) = 4 - (1 + 1) = 2 \\
d_{\mathcal{K}} &= N_K - (d_{\mathcal{U}} + d_{\mathcal{G}}) = 4 - (1 + 1) = 2 \\
d_{\mathcal{E}} &= N_{\mathcal{E}} - (d_{\mathcal{U}} + d_{\mathcal{G}} + d_{\mathcal{F}} + d_{\mathcal{K}}) = 8 - (1 + 1 + 2 + 2) = 2
\end{aligned}
$$

As Fig. A.3 shows, both the number of classes and the degrees of freedom are often written next to their corresponding factors in the diagram, separated by a comma. As expected, the degrees of freedom coincide with the dimensionality of the W-subspace associated with each factor.

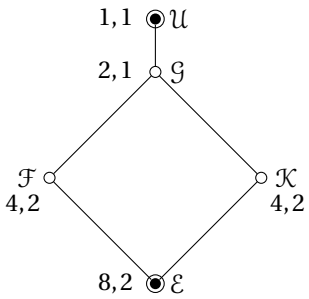


Figure A.3: Hasse diagram of the factor structure in the example including class sizes and degrees of freedom.

The crude sums of squares for each factor are as follows:

$$
\begin{aligned}
CSS_{\mathcal{U}} &= \frac{(1+8+2+4+5+(-4)+2+8)^2}{8} = \frac{26^2}{8} = 84.5 \\
CSS_{\mathcal{G}} &= \frac{(1+8+2+4)^2}{4} + \frac{(5+(-4)+2+8)^2}{4} = \frac{15^2}{4} + \frac{11^2}{4} = \frac{346}{4} = 86.5 \\
CSS_{\mathcal{F}} &= \frac{(1+8)^2}{2} + \frac{(2+4)^2}{2} + \frac{(5+(-4))^2}{2} + \frac{(2+8)^2}{2} = \frac{9^2}{2} + \frac{6^2}{2} + \frac{1^2}{2} + \frac{10^2}{2} = \frac{218}{2} = 109 \\
CSS_{\mathcal{K}} &= \frac{(1+2)^2}{2} + \frac{(8+4)^2}{2} + \frac{(5+2)^2}{2} + \frac{((-4)+8)^2}{2} = \frac{3^2}{2} + \frac{12^2}{2} + \frac{7^2}{2} + \frac{4^2}{2} = \frac{218}{2} = 109 \\
CSS_{\mathcal{E}} &= 1^2 + 8^2 + 2^2 + 4^2 + 5^2 + (-4)^2 + 2^2 + 8^2 = 194
\end{aligned}
$$

which can then be used to calculate the sums of squares:

$$SS_{\mathcal{U}} = CSS_{\mathcal{U}} = 84.5$$
$$SS_{\mathcal{G}} = CSS_{\mathcal{G}} - CSS_{\mathcal{U}} = 86.5 - 84.5 = 2$$
$$SS_{\mathcal{F}} = CSS_{\mathcal{F}} - (CSS_{\mathcal{G}} + CSS_{\mathcal{U}}) = 109 - (84.5 + 2) = 22.5$$
$$SS_{\mathcal{K}} = CSS_{\mathcal{K}} - (CSS_{\mathcal{G}} + CSS_{\mathcal{U}}) = 109 - (84.5 + 2) = 22.5$$
$$SS_{\mathcal{E}} = CSS_{\mathcal{E}} - (CSS_{\mathcal{G}} + CSS_{\mathcal{U}} + CSS_{\mathcal{F}} + CSS_{\mathcal{K}}) = 194 - (84.5 + 2 + 22.5 + 22.5) = 62.5$$

These values coincide with those we obtained before using projection matrices, as expected. The Hasse diagram thus provides a shortcut for calculating the values necessary to perform inferential analysis based on the variance ratios.

A.2 Analysis of Conventional Designs

The example analyses in this section are all based on the same hypothetical situation, which is adapted slightly to the particularities of each of the three conventional experimental designs.

Completely Randomised Design

Consider a wine tasting contest in which $N = 8$ judges are asked to rate the quality of $T = 4$ wines from a minimum of 0 (disgusting) to a maximum of 5 (magnificient). The organisers of the contest decide that each judge will taste a single wine, so each wine will get exactly two scores. More precisely, the random assignment of wines to judges is $\mathcal{T}(\omega) := \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$. The organisers also decide that no particular characteristic of the judges or the wines will be taken into consideration for the analysis of the results. The experiment matches the characteristics of a CRD, so its Hasse diagrams correspond to those shown in Fig. 3.3.

The organisers are mainly interested in answering two questions. Due to the small budget available, they are concerned about the overall quality of the selected wines. More precisely, they want to make sure that they are not completely disgusting — i.e., they want to check that the mean effect μ is significantly higher than 0. Since all selected wines belong to the same (low) price range, the organisers are also interested in knowing if the judges perceive differences in quality. They also agree that they will use a significance level of $\alpha = 0.05$ to test the hypotheses.

Experienced in experimental design, the organisers check before the tasting starts that they have enough degrees of freedom to perform the analysis, obtaining:

$$
\begin{aligned}
d_{\mathcal{U}} &= N_{\mathcal{U}} = 1 \\
d_{\mathcal{T}} &= N_{\mathcal{T}} - d_{\mathcal{U}} = T - 1 = 4 - 1 = 3 \\
d_{\mathcal{E}} &= N_{\mathcal{E}} - (d_{\mathcal{U}} + d_{\mathcal{T}}) = N - (1 + (T - 1)) = N - T = 8 - 4 = 4
\end{aligned}
$$

This ensures that the analysis is feasible, since all values are larger than 0.

Aware that projection matrices simplify the computation of the sums of squares, the organisers obtain the following:

$$
\mathbf{P}_{V_{\mathcal{T}}} = \begin{bmatrix}
1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2
\end{bmatrix}.
$$

They also know that $\mathbf{P}_{V_{\mathcal{U}}} = \mathbf{J}_8/8$ and $\mathbf{P}_{V_{\mathcal{E}}} = \mathbf{I}_8$.

They then conduct the experiment, obtaining the following scores from the judges: $\mathbf{y} = (0,4,1,2,2,4,1,4)$. This allows them to compute the sums of squares corresponding to each factor:

$$
\begin{aligned}
SS_{\mathcal{U}} &= \left\|\mathbf{P}_{W_{\mathcal{U}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 = 40.5 \\
SS_{\mathcal{T}} &= \left\|\mathbf{P}_{W_{\mathcal{T}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} = 43 - 40.5 = 2.5 \\
SS_{\mathcal{E}} &= \left\|\mathbf{P}_{W_{\mathcal{E}}}\mathbf{y}\right\|^2 = \left\|\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 = 58 - 43 = 15
\end{aligned}
$$

so the mean squares are:

$$
\begin{aligned}
MS_{\mathcal{U}} &= {}^{SS_{\mathcal{U}}}/_{d_{\mathcal{U}}} = {}^{40.5}/_{1} = 40.5 \\
MS_{\mathcal{T}} &= {}^{SS_{\mathcal{T}}}/_{d_{\mathcal{T}}} = {}^{2.5}/_{3} = 0.8\overline{3} \\
MS_{\mathcal{E}} &= {}^{SS_{\mathcal{E}}}/_{d_{\mathcal{E}}} = {}^{15}/_{4} = 3.75
\end{aligned}
$$

Last but not least, the organisers compute the necessary variance ratios:

$$
\begin{aligned}
VR_{\mathcal{U}} &= {}^{MS_{\mathcal{U}}}/_{MS_{\mathcal{E}}} = {}^{40.5}/_{3.75} = 10.8 \\
VR_{\mathcal{T}} &= {}^{MS_{\mathcal{T}}}/_{MS_{\mathcal{E}}} = {}^{0.8\overline{3}}/_{3.75} = 0.\overline{2}
\end{aligned}
$$

With these calculations completed, they can finally test their hypotheses of interest. First, they compare the variance ratio of the global mean ($VR_{\mathcal{U}}$) with an F distribution of 1

and 4 degrees of freedom with $\alpha = 0.05$. Looking at the tables, they see that $VR_{\mathcal{U}} = 10.8 > F_{0.05;1;4} = 7.71$. They thus reject the null hypothesis that the global mean equals 0. Second, they compare the variance ratio of the treatment factor ($VR_{\mathcal{T}}$) with an F distribution of 3 and 4 degrees of freedom with $\alpha = 0.05$. Again, they look at the tables and see that $VR_{\mathcal{T}} = 0.\overline{2} < F_{0.05;3;4} = 6.59$. This indicates that they cannot reject the null hypothesis of equal treatment effects — the wines appear of similar quality.

Complete Block Design

Imagine now that the organisers of the wine tasting contest realise that the judges they selected come from two schools with very different tasting traditions. Concerned that this difference might affect the conclusions they extract from the experiment, they decide to introduce a blocking factor in the analysis. More precisely, $\mathcal{B}(\omega) := \{\{1,3,5,7\},\{2,4,6,8\}\}$ to ensure that each kind of wine is tasted by one member of each school. The Hasse diagrams of the experiment thus match those shown in Fig. 3.4.

The organisers recompute the degrees of freedom in this new situation, obtaining:

$$
\begin{aligned}
d_{\mathcal{U}} &= N_{\mathcal{U}} = 1 \\
d_{\mathcal{B}} &= N_{\mathcal{B}} - d_{\mathcal{U}} = B - 1 = 2 - 1 = 1 \\
d_{\mathcal{T}} &= N_{\mathcal{T}} - d_{\mathcal{U}} = T - 1 = 4 - 1 = 3 \\
d_{\mathcal{E}} &= N_{\mathcal{E}} - (d_{\mathcal{U}} + d_{\mathcal{T}}) = N - T - B + 1 = 8 - 4 - 2 + 1 = 3
\end{aligned}
$$

After including a blocking factor, all values are still larger than 0, so the analysis is still feasible. They also need to compute the projection matrix associated with $\mathcal{B}$:

$$
\mathbf{P}_{V_{\mathcal{B}}} = \begin{bmatrix}
1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\
1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\
1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\
1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4
\end{bmatrix}.
$$

Assuming that the response vector is the same as previously ($\mathbf{y} = (0,4,1,2,2,4,1,4)$), the sums of squares for each factor in the experiment are now:

$$
\begin{aligned}
SS_{\mathcal{U}} &= \left\|\mathbf{P}_{W_{\mathcal{U}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 = 40.5 \\
SS_{\mathcal{B}} &= \left\|\mathbf{P}_{V_{\mathcal{B}}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} = 53 - 40.5 = 12.5 \\
SS_{\mathcal{T}} &= \left\|\mathbf{P}_{W_{\mathcal{T}}}\mathbf{y}\right\|^2 = \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 - SS_{\mathcal{U}} = 43 - 40.5 = 2.5 \\
SS_{\mathcal{E}} &= \left\|\mathbf{y}\right\|^2 + \left\|\mathbf{P}_{V_{\mathcal{U}}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{B}}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_{\mathcal{T}}}\mathbf{y}\right\|^2 = 58 + 40.5 - 53 - 43 = 2.5
\end{aligned}
$$

so the mean squares are:

$$
\begin{aligned}
MS_{\mathcal{U}} &= {}^{SS_{\mathcal{U}}}/_{d_{\mathcal{U}}} = {}^{40.5}/_{1} = 40.5 \\
MS_{\mathcal{T}} &= {}^{SS_{\mathcal{B}}}/_{d_{\mathcal{B}}} = {}^{12.5}/_{1} = 12.5 \\
MS_{\mathcal{B}} &= {}^{SS_{\mathcal{T}}}/_{d_{\mathcal{T}}} = {}^{2.5}/_{3} = 0.8\overline{3} \\
MS_{\mathcal{E}} &= {}^{SS_{\mathcal{E}}}/_{d_{\mathcal{E}}} = {}^{2.5}/_{3} = 0.8\overline{3}
\end{aligned}
$$

The variance ratios necessary for testing the hypotheses all need to be recomputed as well:

$$
\begin{aligned}
VR_{\mathcal{U}} &= {}^{MS_{\mathcal{U}}}/_{MS_{\mathcal{E}}} = {}^{40.5}/_{0.8\overline{3}} = 48.5 \\
VR_{\mathcal{B}} &= {}^{MS_{\mathcal{B}}}/_{MS_{\mathcal{E}}} = {}^{12.5}/_{0.8\overline{3}} = 15 \\
VR_{\mathcal{T}} &= {}^{MS_{\mathcal{T}}}/_{MS_{\mathcal{E}}} = {}^{0.8\overline{3}}/_{0.8\overline{3}} = 1
\end{aligned}
$$

which they then compare with the appropriate values in the tables for the F distribution:

Overall mean $VR_{\mathcal{U}} = 48.5 > F_{0.05;1;3} = 10.13$

School $VR_{\mathcal{B}} = 15 > F_{0.05;1;3} = 10.13$

Wine $VR_{\mathcal{T}} = 1 < F_{0.05;3;3} = 9.28$

They thus find significant differences between the scores granted by the members of the two schools. The possible differences in the quality of the wines, however, are still not significant.

Factorial Design

The judges now have a more detailed information about the wines they bought for the contest, and they are interested in extracting as many conclusions as they can with the material they have currently available. They have completely consumed the whole budget, however, so they are not able to acquire more bottles. They identify two factors of interest: the brand of the wine and its type. More precisely, two of their bottles are from "`A`", while the other two are from "`B`", being one of each brand "`red`" and the other "`white`".

Hence, $\mathcal{F} = \{\mathtt{A}, \mathtt{B}\}$ and $\mathcal{G} = \{\mathtt{red}, \mathtt{white}\}$. This reflects an orthogonal factorial design with 2 factors of 2 levels each (often expressed in the literature as a 2^2 factorial design).

After assigning randomly bottles to judges, the equivalence classes of the treatment factors are:

$$\mathcal{F}(\omega) := \{\{1,2,3,4\},\{5,6,7,8\}\}$$
$$\mathcal{G}(\omega) := \{\{1,2,5,6\},\{3,4,7,8\}\}$$

which implies $\mathcal{T}(\omega) = (\mathcal{F} \wedge \mathcal{G})(\omega) := \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$. The projection matrices associated which each factor are:

$$\mathbf{P}_{V_\mathcal{F}} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}, \quad \mathbf{P}_{V_\mathcal{G}} = \begin{bmatrix} 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 \end{bmatrix}.$$

The Hasse diagrams of the factor sets match perfectly those shown in Fig. 3.5, so the organisers calculate the degress of freedom using the cascading process:

$$\begin{aligned} d_\mathcal{U} &= N_\mathcal{U} = 1 \\ d_\mathcal{F} &= N_\mathcal{F} - 1 = 2 - 1 = 1 \\ d_\mathcal{G} &= N_\mathcal{G} - 1 = 2 - 1 = 1 \\ d_\mathcal{T} &= (N_\mathcal{F} - 1)(N_\mathcal{G} - 1) = (2-1)(2-1) = 1 \\ d_\mathcal{E} &= N - (d_\mathcal{U} + d_\mathcal{F} + d_\mathcal{G} + d_\mathcal{T}) = 8 - (1+1+1+1) = 8 - 4 = 4 \end{aligned}$$

Assuming that the measurements are still $\mathbf{y} = (0,4,1,2,2,4,1,4)$, the sums of squares corresponding to each factor in the experiment are:

$$\begin{aligned} SS_\mathcal{U} &= \left\|\mathbf{P}_{V_\mathcal{U}}\mathbf{y}\right\|^2 = 40.5 \\ SS_\mathcal{F} &= \left\|\mathbf{P}_{V_\mathcal{F}}\mathbf{y}\right\|^2 - SS_\mathcal{U} = 42.5 - 40.5 = 2 \\ SS_\mathcal{G} &= \left\|\mathbf{P}_{V_\mathcal{G}}\mathbf{y}\right\|^2 - SS_\mathcal{U} = 41 - 40.5 = 0.5 \\ SS_\mathcal{T} &= \left\|\mathbf{P}_{V_\mathcal{T}}\mathbf{y}\right\|^2 + \left\|\mathbf{P}_{V_\mathcal{U}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_\mathcal{F}}\mathbf{y}\right\|^2 - \left\|\mathbf{P}_{V_\mathcal{G}}\mathbf{y}\right\|^2 = 43 - 40.5 = 2.5 \\ SS_\mathcal{E} &= \left\|\mathbf{y}\right\|^2 - (SS_\mathcal{U} + SS_\mathcal{F} + SS_\mathcal{G} + SS_\mathcal{T}) = 58 - 40.5 - 2 - 0.5 - 2.5 = 12.5 \end{aligned}$$

leading to the mean squares:

$$MS_{\mathcal{U}} = {}^{SS_{\mathcal{U}}}/_{d_{\mathcal{U}}} = {}^{40.5}/_{1} = 40.5$$
$$MS_{\mathcal{F}} = {}^{SS_{\mathcal{F}}}/_{d_{\mathcal{F}}} = {}^{2}/_{1} = 2$$
$$MS_{\mathcal{G}} = {}^{SS_{\mathcal{G}}}/_{d_{\mathcal{G}}} = {}^{0.5}/_{1} = 0.5$$
$$MS_{\mathcal{J}} = {}^{SS_{\mathcal{J}}}/_{d_{\mathcal{J}}} = {}^{2.5}/_{1} = 2.5$$
$$MS_{\mathcal{E}} = {}^{SS_{\mathcal{E}}}/_{d_{\mathcal{E}}} = {}^{12.5}/_{4} = 3.125$$

whose corresponding variance ratios are:

$$VR_{\mathcal{U}} = {}^{MS_{\mathcal{U}}}/_{MS_{\mathcal{E}}} = {}^{40.5}/_{3.125} = 12.96$$
$$VR_{\mathcal{F}} = {}^{MS_{\mathcal{F}}}/_{MS_{\mathcal{E}}} = {}^{2}/_{3.125} = 0.64$$
$$VR_{\mathcal{G}} = {}^{MS_{\mathcal{G}}}/_{MS_{\mathcal{E}}} = {}^{0.5}/_{3.125} = 0.16$$
$$VR_{\mathcal{J}} = {}^{MS_{\mathcal{J}}}/_{MS_{\mathcal{E}}} = {}^{2.5}/_{3.125} = 0.8$$

Comparing these values with the appropriate values in the tables for the F distribution, they observe:

Overall mean $VR_{\mathcal{U}} = 12.96 > F_{0.05;1;4} = 7.71$

Brand $VR_{\mathcal{B}} = 0.64 < F_{0.05;1;4} = 7.71$

Type $VR_{\mathcal{J}} = 0.16 < F_{0.05;1;4} = 7.71$

Wine $VR_{\mathcal{J}} = 0.8 < F_{0.05;1;4} = 7.71$

Therefore, the only hypothesis the organisers can reject from the judges' scores is that all wines are "disgusting".

APPENDIX B

EXAMPLE ANALYSIS OF MEASUREMENTS FROM AN INTERVENTION-BASED STUDY

This appendix exemplifies proposed steps for the analysis of performance measurements from a hypothetical study involving targeted interventions with simulated data. The code for the example is available online.[1] This will hopefully help readers implement their own analysis pipelines.

Figure B.1 shows the distribution of simulated performance measurements for a classification experiment with $X = 2$ feature extractors ($\mathcal{X}(i) \in \{e_1, e_2\}$) and $L = 2$ learning algorithms ($\mathcal{L}(i) \in \{\ell_1, \ell_2\}$) on a collection from which $K = 40$ train/test sample pairs have been generated. The pipeline for this experiment has been factorially subjected to two interventions, identified by the factor variables $\mathcal{Z}$ and $\mathcal{W}$. The combination of levels (z, w) at the top left corner of the figure reflects the completely unregulated measurements, with the combination (z', w') at the bottom right corresponding to the measurements under simultaneous regulation. This hypothetical example thus resembles the case study in Sec. 5.3 but with fewer evaluated methods.

Assume the structural model in Eqn (6.21) — or its logistic version in Eqn (6.29) — is adopted to analyse the measurements. To this end, one can employ the `R` package `lme4` suggested in Sec. 6.3, since it can deal with both linear and logistic mixed-effects models. The procedure below illustrates its usage in both cases. Nevertheless, the `lme4` package

[1] `https://github.com/franrodalg/logistic_struct_model`

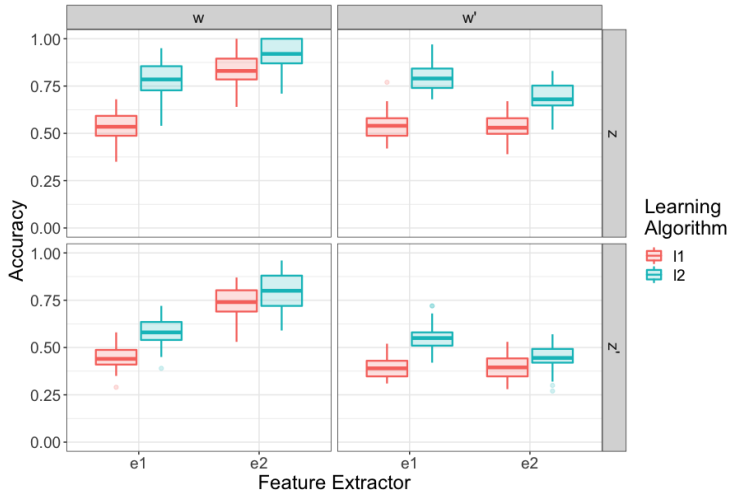


Figure B.1: Distribution of simulated performance measurements from a hypothetical study with two feature extractors, two learning algorithms, and two interventions, each with regulated and unregulated evaluation conditions ($\mathcal{Z}(i) \in \{z, z'\}$ and $\mathcal{W}(i) \in \{w, w'\}$).

is not entirely consistent in how it presents its output, lacking frequentist statistical significance analysis for its linear estimates. If such information is of interest, the output of `lme4` can be extended using the `lmerTest` package (Kuznetsova et al., 2017). Since the data have been simulated with the sole purpose of illustrating the procedure and not to adhere to any real data generating process, no conclusion should be taken from this example on whether linear or logistic approaches are more suitable for inferential analyses from classification experiments. Devoted studies are necessary to this end.

a) Format measurements

In order to use the model-fitting functions in `lme4`, the input data should be stored in a data frame with one observation per row and columns for each factor variable of interest, plus one for the response. Although logistic estimation should accept probability values — ranging from 0 to 1 — as response, its implementation in `lme4` behaves unexpectedly unless response values are exactly 0 or 1. Hence, if one intends to use a logistic approach in `lme4`, each observation should correspond to an individual success or failure in prediction. In the example, the data frame containing these binary values is named `observations`. Conversely, for the linear approach, continuous-valued aggregates work best. In the example, the data frame containing such aggregates is named `measurements`.

Both `observations` and `measurements` contain the following columns: `samp` refers

to the train/test pair $\mathcal{K}(i)$, with values 1 to 40, `feat_ext` refers to the feature extractor $\mathcal{X}(i)$, with values `e1` or `e2`, `learn_alg` refers to the learning algorithm $\mathcal{L}(i)$, with values `l1` or `l2`, `conf_1` refers to the intervention $\mathcal{Z}(i)$, with values `z` or `z'`, `conf_2` refers to the intervention $\mathcal{W}(i)$, with values `w` or `w'`, and `y` refers to the response y_i, with either binary values in `observations` or their aggregate mean per sample in `measurements`. Moreover, `observations` contains three additional columns (`ability`, `difficulty` and `prob`) used to generate the simulated success or failure values for each observation. These would not be available on a real study, but here serve to assess the estimates `lme4` calculates.

b) Define model formula

The structural model intended for the analysis can be defined in `R` using a "formula" — an unevaluated expression that relates symbols. Regardless of whether one intends to use the linear or logistic capabilities of the `lme4` package, the formulae capturing the underlying structural models are identical. In particular, a complete formula for the models in Eqn (6.21) and (6.29) would be expressed as:

```
model_complete <- y ~ feat_ext + learn_alg +
    feat_ext*learn_alg + conf_1 + conf_2 + conf_1*conf_2 +
    feat_ext*conf_1 + feat_ext*conf_2 +
    learn_alg*conf_1 + learn_alg*conf_2 +
    learn_alg*feat_ext*conf_1 + learn_alg*feat_ext*conf_2 +
    (1 | samp)
```

All terms from the structural models are included in the formula separated by plus signs, except for the benchmark parameter μ and residual ε_i, which are implicit, and the response y_i, whose corresponding column name is written on the left-hand side of a tilde sign to mark it as the dependent variable. Fixed-effect parameters are represented by the name of the column that encodes its associated variable, with asterisks representing their mutual interactions. Random-effect parameters, on the other hand, follow the more complex notation `( expr | factor )`. As Bates et al. (2015) illustrate, `expr` can be used to define relationships between intercepts and slopes for the levels of random-effect factors. In the simplest case, `expr` is set to 1, indicating that the intercepts for each level are random with a fixed mean to be estimated. More complex expressions could be employed in the future to represent presumed hierarchical relationships between factors.

The model fitting functions in the `lme4` package make some implicit assumptions that make several terms in the formula above redundant. In particular, for every single high-order interaction term in a formula, it is assumed that all its partial interactions and in-

dividual constitutive elements are also part of the underlying model and thus need to be estimated, regardless of whether they are explicitly expressed or not. For instance, from the three-way interaction `learn_alg*feat_ext*conf_1`, the `lme4` functions will infer that estimates should be calculated for `learn_alg`, `feat_ext` and `conf_1`, as well as for their mutual pair-wise interactions. This means that the same underlying model can be expressed using a much more concise formula:

```
model <- y ~ conf_1*conf_2 +
    learn_alg*feat_ext*conf_1 +
    learn_alg*feat_ext*conf_2 +
    (1 | samp)
```

c) Fit model

Once measurements are properly formatted and the intended structural model has been defined as an `R` formula, functions from `lme4` provide parameter estimates. These functions differ for linear and logistic approaches, so they are illustrated separately below.

Linear fit The `lmer` function from the `lme4` package enables fitting linear mixed-effects models, with the package `lmerTest` offering a version of this function with the same signature but with extended output. To use `lmer`, one simply needs to run the following:

```
linear_fit <- lmerTest::lmer(model, data=measurements)
```

To inspect the result of the model fitting, one can then use the `summary` function:

```
summary(linear_fit)
```

which, for the example data, produces the following output:

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3901 -0.5891  0.0618  0.5991  2.8693

Random effects:
 Groups   Name        Variance Std.Dev.
 samp     (Intercept) 0.003732 0.06109
 Residual             0.002313 0.04809
Number of obs: 640, groups:  samp, 40

Fixed effects:
                                Estimate Std. Error          df t value Pr(>|t|)
(Intercept)                     0.535016   0.011844  80.648050  45.173  < 2e-16 ***
learn_algl2                     0.242500   0.009313 588.000000  26.038  < 2e-16 ***
feat_exte2                      0.297062   0.009313 588.000000  31.897  < 2e-16 ***
conf_1z'                       -0.086531   0.008502 588.000000 -10.178  < 2e-16 ***
conf_2w'                        0.004219   0.008502 588.000000   0.496   0.6199
learn_algl2:feat_exte2         -0.142437   0.013171 588.000000 -10.815  < 2e-16 ***
learn_algl2:conf_1z'           -0.099250   0.010754 588.000000  -9.229  < 2e-16 ***
feat_exte2:conf_1z'            -0.001375   0.010754 588.000000  -0.128   0.8983
learn_algl2:conf_2w'            0.016500   0.010754 588.000000   1.534   0.1255
feat_exte2:conf_2w'            -0.294625   0.010754 588.000000 -27.397  < 2e-16 ***
conf_1z':conf_2w'              -0.060937   0.007604 588.000000  -8.014 6.02e-15 ***
learn_algl2:feat_exte2:conf_1z' 0.032625   0.015209 588.000000   2.145   0.0323 *
learn_algl2:feat_exte2:conf_2w' 0.020625   0.015209 588.000000   1.356   0.1756
```

The `lmer` function thus estimates all parameters from the structural model, even if they were not explicitly included in the formula, separating those with random and fixed effects. The two right-most columns for fixed-effects parameters are added by `lmerTest` and indicate the level of statistical significance of the corresponding parameter — three asterisks represent p-values smaller than 0.001, whereas a single asterisk indicates a p-value between 0.01 and 0.05. The output also includes a correlation matrix for fixed-effect parameters, but the matrix does not appear in here due to the high number of parameters in the model. To retrieve it, one can use `vcov(linear_fit)`.

Logistic fit The `lme4` package provides the `glmer` function to fit Generalised Linear Mixed-effects Models (GLMMs), including logistic ones. The call is similar to `lmer`, only requiring to specify the random variable family assumed for the data, which, in turn, sets the link function that will be employed. For a logistic analysis, the `family` argument should be set to `"binomial"`. In addition, to ensure convergence of the estimates, it might be necessary to replace the default optimiser and increase the maximum number of iterations. The `glmerControl` function enables users to make such changes. The following instructions were used to fit the model in the example:

```
logistic_fit <- lme4::glmer(
  model,
  family="binomial",
  data=observations,
  control=lme4::glmerControl(optimizer="bobyqa",
                             optCtrl=list(maxfun=1e6)))
```

whose `summary` produces the following output:

```
     AIC      BIC   logLik deviance df.resid
 75984.3  76111.3 -37978.2  75956.3    63986

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4503 -0.9309  0.4476  0.7646  1.7443

Random effects:
 Groups Name        Variance Std.Dev.
 samp   (Intercept) 0.08743  0.2957
Number of obs: 64000, groups:  samp, 40

Fixed effects:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                      0.18430    0.05510   3.345 0.000824 ***
learn_algl2                      1.15023    0.04291  26.803  < 2e-16 ***
feat_exte2                       1.43976    0.04436  32.459  < 2e-16 ***
conf_1z'                        -0.43668    0.03686 -11.849  < 2e-16 ***
conf_2w'                        -0.06446    0.03680  -1.752 0.079856 .
learn_algl2:feat_exte2          -0.38817    0.06887  -5.636 1.74e-08 ***
learn_algl2:conf_1z'            -0.56059    0.04810 -11.654  < 2e-16 ***
feat_exte2:conf_1z'             -0.09470    0.04800  -1.973 0.048514 *
learn_algl2:conf_2w'             0.06956    0.04779   1.455 0.145544
feat_exte2:conf_2w'             -1.38597    0.04838 -28.646  < 2e-16 ***
conf_1z':conf_2w'               -0.08993    0.03602  -2.496 0.012546 *
learn_algl2:feat_exte2:conf_1z'  0.09791    0.07132   1.373 0.169833
learn_algl2:feat_exte2:conf_2w' -0.15012    0.07210  -2.082 0.037328 *
```

This approach also provides estimates for all parameters in the structural model, both for fixed and random effects. Note that residuals are not included as parameters to estimate. As mentioned in Sec. 6.3, this is conventional in logistic models. Goodness of fit metrics are also included to facilitate comparisons between alternative models.

As expected, the computational complexity of logistic estimates is substantially higher than in the linear case. From 10 replications, it took an average of 0.57 seconds to fit the model using `lmer` and 375.89 seconds using `glmer`, almost 700 times longer.

d) Interpret estimates

Both linear and logistic approaches generate a single estimate for each fixed-effect parameter in the example because, for simplicity, all factors have exactly 2 levels. In general, one would obtain $F-1$ estimates per parameter, where F represents the number of levels of the corresponding factor variable. If, for instance, a third learning algorithm ℓ_3 was included, an additional estimate `learn_algl3` would be computed. In addition, every single interaction involving $\mathcal{L}$ would also feature an additional estimate, such as `learn_algl3:feat_exte2` or `conf_1z':learn_algl3:feat_exte2`. The level from each factor not receiving an estimate acts as baseline, with the $F-1$ estimates directly or indirectly expressing differential effects against such a baseline.

Average performance estimates The `(Intercept)` estimate in both linear and logistic cases reflects the estimated average performance of systems with $\mathcal{X}(i) = e_1$ and $\mathcal{L}(i) = \ell_1$ under completely unregulated conditions (i.e., $\mathcal{Z}(i) = z$ and $\mathcal{W}(i) = w$). The linear estimate directly matches such value, meaning the accuracy estimated for (e_1, ℓ_1) systems under unregulated conditions is around 53.5%. To obtain the corresponding value from the logistic estimate, however, it is necessary to transform the estimate using the inverse *logit* operation in Eqn (6.24). In particular, $logit^{-1}(0.184) \approx 0.546$. Linear and logistic estimates of performance are thus quite similar to each other as well as to the "ability" value of 0.55 set in the simulation as base probability for this combination of factor levels.

To obtain average performance estimates for level combinations other than the baseline, it is necessary to reconstruct them adding parameter estimates to the `(Intercept)` according to the levels on which they differ. For instance, for (e_1, ℓ_2) systems under unregulated conditions, performance estimates require adding the `learn_algl2` parameter estimate only, since they only differ from the baseline in that $\mathcal{L}(i) = \ell_2$. In the lin-

ear case, this yields an average performance estimate for these systems of approximately $0.535 + 0.243 = 0.778$. Adding a further level change, such as $\mathcal{X}(i) = e_2$, would require not only to add the `feat_exte2` parameter estimate but also the `learn_algl2:feat_exte2` interaction parameter, thus yielding approximately $0.535 + 0.243 + 0.297 - 0.142 = 0.933$. The `emmeans` function, from the package of the same name,[2] provides a simple interface to obtain all performance estimates:

```
emmeans::emmeans(linear_fit,
  c('feat_ext', 'learn_alg', 'conf_1', 'conf_2'))
```

```
feat_ext learn_alg conf_1 conf_2 emmean     SE   df lower.CL upper.CL
e1       l1        z       w       0.535 0.0118 80.7    0.511    0.559
e2       l1        z       w       0.832 0.0118 80.7    0.809    0.856
e1       l2        z       w       0.778 0.0118 80.7    0.754    0.801
e2       l2        z       w       0.932 0.0118 80.7    0.909    0.956
e1       l1        z'      w       0.448 0.0118 80.7    0.425    0.472
e2       l1        z'      w       0.744 0.0118 80.7    0.721    0.768
e1       l2        z'      w       0.592 0.0118 80.7    0.568    0.615
e2       l2        z'      w       0.778 0.0118 80.7    0.754    0.801
e1       l1        z       w'      0.539 0.0118 80.7    0.516    0.563
e2       l1        z       w'      0.542 0.0118 80.7    0.518    0.565
e1       l2        z       w'      0.798 0.0118 80.7    0.775    0.822
e2       l2        z       w'      0.679 0.0118 80.7    0.655    0.702
e1       l1        z'      w'      0.392 0.0118 80.7    0.368    0.415
e2       l1        z'      w'      0.393 0.0118 80.7    0.369    0.416
e1       l2        z'      w'      0.552 0.0118 80.7    0.528    0.575
e2       l2        z'      w'      0.463 0.0118 80.7    0.440    0.487

Degrees-of-freedom method: kenward-roger
Confidence level used: 0.95
```

The logistic performance estimates are obtained similarly, summing the corresponding parameter estimates and, in this case, computing the inverse *logit* of the sum. For instance, the logistic performance estimate for (e_1, ℓ_2) systems under unregulated conditions are calculated as $logit^{-1}\big(\texttt{(Intercept)} + \texttt{learn\_algl2}\big)$, which is approximately $logit^{-1}(0.184 + 1.15) = logit^{-1}(1.334) \approx 0.792$. The remaining performance estimates are calculated in the same way, and can also be obtained using the `emmeans` function:

```
emmeans::emmeans(logistic_fit,
  c('feat_ext', 'learn_alg', 'conf_1', 'conf_2'),
  type='response')
```

```
feat_ext learn_alg conf_1 conf_2  prob       SE  df asymp.LCL asymp.UCL
e1       l1        z       w       0.546 0.01366 Inf     0.519     0.573
e2       l1        z       w       0.835 0.00816 Inf     0.819     0.851
e1       l2        z       w       0.792 0.00957 Inf     0.772     0.810
e2       l2        z       w       0.916 0.00499 Inf     0.905     0.925
e1       l1        z'      w       0.437 0.01357 Inf     0.411     0.464
e2       l1        z'      w       0.749 0.01075 Inf     0.727     0.769
e1       l2        z'      w       0.584 0.01347 Inf     0.557     0.610
e2       l2        z'      w       0.801 0.00939 Inf     0.782     0.819
e1       l1        z       w'      0.530 0.01372 Inf     0.503     0.557
e2       l1        z       w'      0.543 0.01373 Inf     0.516     0.570
e1       l2        z       w'      0.792 0.00955 Inf     0.773     0.811
e2       l2        z       w'      0.702 0.01188 Inf     0.678     0.724
e1       l1        z'      w'      0.400 0.01327 Inf     0.374     0.426
e2       l1        z'      w'      0.390 0.01322 Inf     0.364     0.416
e1       l2        z'      w'      0.563 0.01362 Inf     0.536     0.589
e2       l2        z'      w'      0.443 0.01372 Inf     0.416     0.470

Confidence level used: 0.95
Intervals are back-transformed from the logit scale
```

[2] https://github.com/rvlenth/emmeans

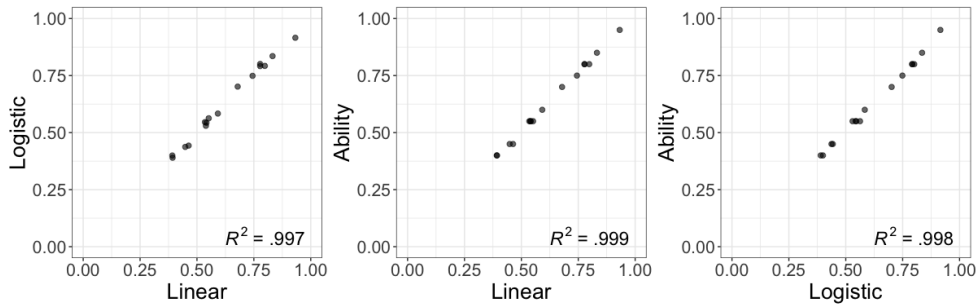


Figure B.2: Correlation between average performance estimates from a hypothetical study using linear and logistic models (Left), as well as correlations between "ability" values used for data simulation and such linear (Middle) and logistic (Right) estimates.

The `type='response'` argument causes the estimates to be automatically converted from the *logit* scale to probabilities, facilitating their interpretation. The `Inf` values in the "degrees of freedom" (`df`) column of `emmeans` for logistic models is expected, indicating that estimates have been tested against a standard normal distribution.[3]

Performance estimates in this example — the `emmean` and `prob` columns for the linear and logistic cases, respectively — are almost identical. As Fig. B.2 shows, their estimates correlate almost perfectly with each other ($R^2 = 0.997$) and with the "ability" values used to seed the simulated observations ($R^2 = 0.999$ in the linear case and $R^2 = 0.997$ in the logistic case). Nevertheless, there is no guarantee that such concordances will happen on real data, since they might be artefacts derived from the simplistic simulation procedure.

Although presented in columns with different names, `emmeans` provides boundaries for 95% confidence intervals of the estimates from both linear and logistic cases (`lower.CL` and `upper.CL` in the former, and `asymp.LCL` and `asymp.UCL` in the latter). Confidence intervals are often regarded as more suitable than raw p-values to report frequentist inferential analyses (Cormack and Lynam, 2006; Urbano et al., 2013), and might be used as bases for ranking benchmarked methods.

Factor relevance A major motivation for expressing measurements as structural models is to determine which contributions to the measurements appear more relevant. The output tables that `summary` generates from the fitted models include significance levels

[3] See `https://cran.r-project.org/web/packages/emmeans/vignettes/FAQs.html` for further information.

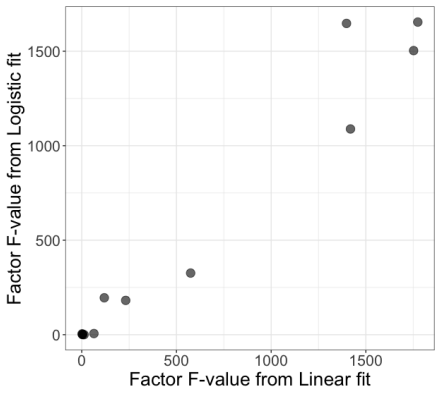


Figure B.3: Comparison between factor F-values obtained from fitting linear and logistic models with observations from a hypothetical study.

for the estimated parameters, but these do not necessarily reflect the relevance of the parameter as a whole. As mentioned in Ch. 3, the usual approach to this end is to conduct an Analysis of Variance (ANOVA). Base R provides the `anova` function to this end, which produces the following from the linear fit:

```
anova(linear_fit)
```

```
Type III Analysis of Variance Table with Satterthwaite's method
                          Sum Sq Mean Sq NumDF DenDF   F value    Pr(>F)
learn_alg                 3.2819  3.2819     1   588 1418.8823 < 2.2e-16 ***
feat_ext                  1.3295  1.3295     1   588  574.8037 < 2.2e-16 ***
conf_1                    4.0529  4.0529     1   588 1752.2417 < 2.2e-16 ***
conf_2                    4.1040  4.1040     1   588 1774.3301 < 2.2e-16 ***
learn_alg:feat_ext        0.5365  0.5365     1   588  231.9517 < 2.2e-16 ***
learn_alg:conf_1          0.2751  0.2751     1   588  118.9566 < 2.2e-16 ***
feat_ext:conf_1           0.0089  0.0089     1   588    3.8587 0.0499588 *
learn_alg:conf_2          0.0288  0.0288     1   588   12.4326 0.0004548 ***
feat_ext:conf_2           3.2333  3.2333     1   588 1397.9078 < 2.2e-16 ***
conf_1:conf_2             0.1485  0.1485     1   588   64.2179 6.018e-15 ***
learn_alg:feat_ext:conf_1 0.0106  0.0106     1   588    4.6018 0.0323474 *
learn_alg:feat_ext:conf_2 0.0043  0.0043     1   588    1.8391 0.1755724
```

and from the logistic fit:

```
anova(logistic_fit)
```

```
Analysis of Variance Table
                          npar  Sum Sq Mean Sq   F value
learn_alg                    1 1088.42 1088.42 1088.4204
feat_ext                     1  325.96  325.96  325.9619
conf_1                       1 1503.10 1503.10 1503.0966
conf_2                       1 1654.05 1654.05 1654.0480
learn_alg:feat_ext           1  181.45  181.45  181.4490
learn_alg:conf_1             1  195.22  195.22  195.2159
feat_ext:conf_1              1    0.35    0.35    0.3527
learn_alg:conf_2             1    0.08    0.08    0.0849
feat_ext:conf_2              1 1646.94 1646.94 1646.9395
conf_1:conf_2                1    5.90    5.90    5.8958
learn_alg:feat_ext:conf_1    1    2.24    2.24    2.2409
learn_alg:feat_ext:conf_2    1    4.33    4.33    4.3293
```

Th latter table not only does not include significance values but also populates the `Sum Sq` and `Mean Sq` columns with the test statistic values. At the moment of writing,

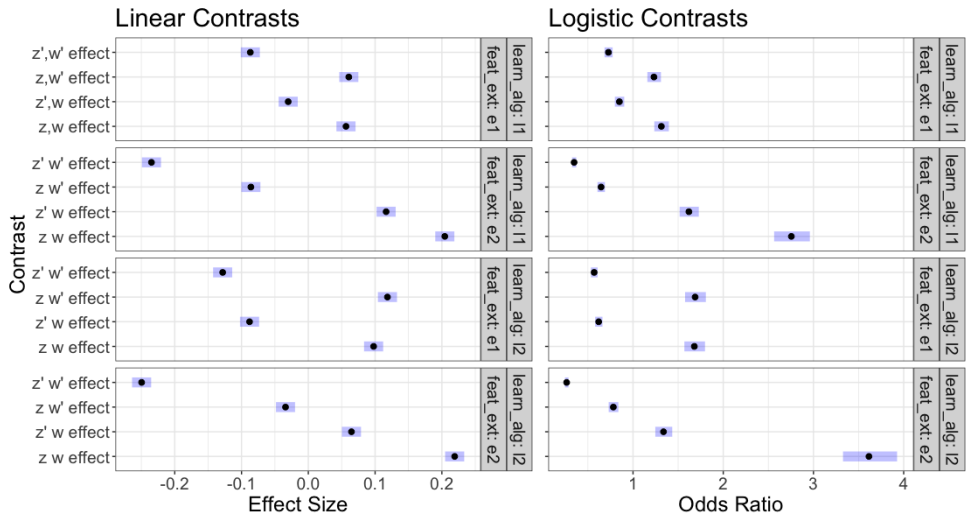


Figure B.4: Contrasts between combinations of potential confounder factor levels for each method from a hypothetical study, expressed in terms of Effect Size for those from fitting a linear model (Left) and Odds Ratio for those from a logistic model (Right).

no widely adopted solution exists in `R` for these kinds of analyses using logistic models, so its development might be worth considering in the future if such analyses are deemed necessary. Moreover, for the linear case, `anova` relies on standard ANOVA templates, such as the Type III mentioned in the output. A more generic approach based on the Calculus of Factors, such as the one implemented in `Mathematica` by Großmann (2014), might also prove useful. Nevertheless, the test statistics in the column named "`F value`" can serve to rank the relevance of contributions and, for instance, discern whether confounding factors affect the results. As Fig. B.3 shows, the test statistics generated from the example data largely agree on which factors appear most relevant (i.e., those that contribute the most to the variability within the measurements), with an $R^2 = 0.978$.

Contrasts Aside from the global relevance of factors, it might be useful to compare particular combinations of factor levels. The `contrasts` function from the `emmeans` package computes such comparisons. With no arguments other than the output of `emmeans` (such as `linear_means` above), it provides effect sizes for all combinations of fixed-effects factor levels. These are deviations from a global mean, expressed in terms of odds ratio for logistic models. Otherwise, including the `method='pairwise'` argument causes `contrasts` to compute relative differences for every combination of level pairs, 120 in the example.

In an intervention-based study, it might be relevant to determine whether potential confounders affect, and to which extent, different system-constructing methods. This can be achieved using the `emmeans` functions as follows:

```
emmeans::contrast(
  emmeans::emmeans(linear_fit, c('conf_1', 'conf_2'),
                   by=c('learn_alg', 'feat_ext')))
```

```
learn_alg = l1, feat_ext = e1:
 contrast      estimate      SE  df t.ratio p.value
 z,w effect     0.0564 0.0057 588   9.888 <.0001
 z',w effect   -0.0301 0.0057 588  -5.285 <.0001
 z,w' effect    0.0606 0.0057 588  10.627 <.0001
 z',w' effect  -0.0869 0.0057 588 -15.230 <.0001

learn_alg = l2, feat_ext = e1:
 contrast      estimate      SE  df t.ratio p.value
 z w effect     0.0978 0.0057 588  17.142 <.0001
 z' w effect   -0.0880 0.0057 588 -15.433 <.0001
 z w' effect    0.1185 0.0057 588  20.775 <.0001
 z' w' effect  -0.1282 0.0057 588 -22.485 <.0001

learn_alg = l1, feat_ext = e2:
 contrast      estimate      SE  df t.ratio p.value
 z w effect     0.2044 0.0057 588  35.838 <.0001
 z' w effect    0.1165 0.0057 588  20.424 <.0001
 z w' effect   -0.0860 0.0057 588 -15.082 <.0001
 z' w' effect  -0.2349 0.0057 588 -41.180 <.0001

learn_alg = l2, feat_ext = e2:
 contrast      estimate      SE  df t.ratio p.value
 z w effect     0.2191 0.0057 588  38.424 <.0001
 z' w effect    0.0646 0.0057 588  11.329 <.0001
 z w' effect   -0.0341 0.0057 588  -5.986 <.0001
 z' w' effect  -0.2496 0.0057 588 -43.767 <.0001

Degrees-of-freedom method: kenward-roger
P value adjustment: fdr method for 4 tests
```

and similarly for the logistic case. Figure B.4 represents these contrasts, and suggest that, in this hypothetical study, the potential confounders affect the considered methods differently, with the largest differences occurring when both are regulated in systems using feature extractor e_2.

REFERENCES

Agresti, A. (2002). *Categorical Data Analysis*. 2nd edition. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley and Sons (cited on p. 181).

Alpaydin, E. (2014). *Introduction to Machine Learning*. 3rd edition. Cambridge, MA, USA: The MIT Press (cited on pp. 43, 44, 89, 92, 114).

Andén, J., Lostanlen, V., and Mallat, S. (2015). "Joint Time-frequency Scattering for Audio Classification". In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. Boston, MA, USA (cited on p. 49).

Andén, J. and Mallat, S. (2011). "Multiscale Scattering for Audio Classification". In *Proc. 12th International Society for Music Information Retrieval Conference (ISMIR'11)*. Miami, FL, USA, pp. 657–662 (cited on pp. 49, 105).

Andén, J. and Mallat, S. (2014). "Deep Scattering Spectrum". *IEEE Transactions on Signal Processing*, 62(16), pp. 4114–4128 (cited on pp. 30, 33, 47, 49, 50, 98, 99, 103, 105, 106, 108, 110, 112, 121, 122, 139, 164, 165, 186, 187).

Aucouturier, J.-J. and Bigand, E. (2013). "Seven Problems that Keep MIR from Attracting the Interest of Cognition and Neuroscience". *Journal of Intelligent Information Systems*, 41(3), pp. 483–497 (cited on p. 59).

Aucouturier, J.-J. and Pachet, F. (2003). "Representing Musical Genre: A State of the Art". *Journal of New Music Research*, 32(1), pp. 83–93 (cited on p. 58).

Bailey, R. A. (2008). *Design of Comparative Experiments*. Cambridge Series in Statistical and Probabilistic Mathematics vol. 25. Cambridge University Press (cited on pp. 70, 76, 79, 169, 170, 176, 189).

Bailey, R. A. (2015). “Structures Defined by Factors”. In *Handbook of Design and Analysis of Experiments*. Ed. by A. Dean, M. Morris, J. Stufken, and D. Bingham. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press. Chap. 10, pp. 371–414 (cited on pp. 76, 79, 80).

Baker, F. B. and Kim, S.-H. (2017). *The Basics of Item Response Theory Using R*. Springer International Publishing AG (cited on p. 64).

Bartlett, J. W. and Frost, C. (2008). “Reliability, Repeatability and Reproducibility: Analysis of Measurement Errors in Continuous Variables”. *Ultrasound in Obstetrics and Gynecology*, 31(4), pp. 466–475 (cited on p. 53).

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). “Fitting Linear Mixed-Effects Models Using `lme4`”. *Journal of Statistical Software*, 67(1), pp. 1–48 (cited on pp. 180, 222).

Benavoli, A., Corani, G., Demšar, J., and Zaffalon, M. (2017). “Time for a Change: A Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis”. *Journal of Machine Learning Research*, 18, pp. 1–36 (cited on pp. 66, 190).

Berrar, D. and Dubitzky, W. (2018). “Should Significance Testing Be Abandoned in Machine Learning?” *International Journal of Data Science and Analytics*, 7(4), pp. 247–257 (cited on p. 66).

Bertin-Mahieux, T., Eck, D., Maillet, F., and Lamere, P. (2008). “Autotagger: A Model for Predicting Social Tags from Acoustic Features on Large Music Databases”. *Journal of New Music Research*, 37(2), pp. 115–135 (cited on p. 31).

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B. A., and Lamere, P. (2011). “The Million Song Dataset”. In *Proc. 12th International Society for Music Information Retrieval Conference (ISMIR'11)*, pp. 591–596 (cited on p. 41).

Bogdanov, D., Porter, A., Herrera, P., and Serra, X. (2016). “Cross-Collection Evaluation for Music Classification Tasks”. In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 379–385 (cited on p. 59).

Bogdanov, D., Porter, A., Urbano, J., and Schreiber, H. (2018). “The MediaEval 2018 AcousticBrainz Genre Task: Content-based Music Genre Recognition from Multiple

Sources". In *Proc. MediaEval 2018 Multimedia Benchmark Workshop*. Sophia Antipolis, France (cited on p. 41).

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). "Essentia: An Audio Analysis Library for Music Information Retrieval". In *Proc. 14th International Society for Music Information Retrieval Conference (ISMIR'13)*. Curitiba, Brazil, pp. 493–498 (cited on pp. 33, 139).

Box, G. E. P. (1976). "Science and Statistics". *Journal of the American Statistical Association*, 71, pp. 791–799 (cited on p. 190).

Bruna, J. and Mallat, S. (2013). "Invariant Scattering Convolution Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), pp. 1872–1886 (cited on p. 48).

Campbell, D. T. (1957). "Factors Relevant to the Validity of Experiments in Social Settings". *Psychological Bulletin*, 54(4), pp. 297–312 (cited on p. 53).

Cano, P., Gómez, E., Gouyon, F., Herrera, P., Koppenberger, M., Ong, B., Serra, X., Streich, S., and Wack, N. (2006). *ISMIR 2004 Audio Description Contest*. Tech. rep. Music Technology Group, Universitat Pompeu Fabra (cited on p. 36).

Carterette, B. A. (2011). "System Effectiveness, User Models, and User Utility: a Conceptual Framework for Investigation". In *Proc. 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*. Bejing, China, pp. 903–912 (cited on p. 55).

Carterette, B. A. (2012). "Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments". *ACM Transactions on Information Systems*, 30(1), pp. 1–34 (cited on pp. 153, 179).

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. (2008). "Content-Based Music Information Retrieval: Current Directions and Future Challenges". *Proc. IEEE*, 96(4), pp. 668–696 (cited on pp. 28, 29).

Celma, Ò. (2010). *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer-Verlag Berlin Heidelberg (cited on p. 29).

Charalambous, C. C. and Bharath, A. A. (2016). “A Data Augmentation Methodology for Training Machine/Deep Learning Gait Recognition Algorithms”. In *British Machine Vision Conference* (cited on pp. 55, 126, 127).

Chen, J. H. and Asch, S. M. (2017). “Machine Learning and Prediction in Medicine. Beyond the Peak of Inflated Expectations”. *New England Journal of Medicine*, 376(26), pp. 2507–2509 (cited on pp. 126, 199).

Cheng, C.-S. (2014). *Theory of Factorial Design. Single- and Multi-Stratum Experiments*. CRC Press (cited on p. 76).

Chi, T., Ru, P., and Shamma, S. (2005). “Multiresolution Spectrotemporal Analysis of Complex Sounds”. *Journal of the Acoustical Society of America*, 118(2), pp. 887–906 (cited on p. 49).

Choi, K., Fazekas, G., and Sandler, M. (2016). “Explaining Deep Convolutional Neural Networks on Music Classification”. *arXiv preprint arXiv:1607.02444* (cited on p. 63).

Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2017). “Transfer Learning for Music Classification and Regression Tasks”. In *Proc. 18th International Society for Music Information Retrieval Conference (ISMIR'17)*. Suzhou, China, pp. 141–149 (cited on pp. 47, 48).

Cleverdon, C. W. (1991). “The Significance of the Cranfield Tests on Index Languages”. In *Proc. 14th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1991)*. Chicago, IL, USA, pp. 9–12 (cited on p. 35).

Cobb, G. W. (1998). *Design and Analysis of Experiments*. Springer-Verlag (cited on pp. 55, 67, 68, 70).

Cohen, P. R. (1995). *Empirical Methods for Artificial Intelligence*. Cambridge, MA, USA: The MIT Press (cited on pp. 34, 35, 89).

Coolican, H. (2017). *Research Methods and Statistics in Psychology*. 6th edition. Psychology Press (cited on p. 55).

Cormack, G. V. and Lynam, T. R. (2006). “Statistical Precision of Information Retrieval Evaluation”. In *Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 533–540 (cited on p. 227).

Costa, Y. M. G., Oliveira, L. S., and Silla Jr, C. N. (2017). "An Evaluation of Convolutional Neural Networks for Music Classification using Spectrograms". *Applied Soft Computing*, 52, pp. 28–38 (cited on p. 33).

Craft, A. J. D., Wiggins, G. A., and Crawford, T. (2007). "How Many Beans Make Five? The Consensus Problem in Music-Genre Classification and a New Evaluation Method for Single-Genre Categorisation Systems". In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 73–76 (cited on pp. 42, 58).

Cunningham, S. J., Bainbridge, D., and Downie, J. S. (2012). "The Impact of MIREX on Scholarly Research". In *Proc. 13th International Society for Music Information Retrieval Conference (ISMIR'12)*. Porto, Portugal, pp. 259–264 (cited on p. 36).

Davies, M. and Böck, S. (2014). "Evaluating the Evaluation Measures for Beat Tracking". In *Proc. 15th International Society for Music Information Retrieval Conference (ISMIR'14)*. Taipei, Taiwan, pp. 637–642 (cited on p. 57).

Davis, S. B. and Mermelstein, P. (1980). "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences". *IEEE Transactions on Audio, Speech, and Language Processing*, 28(4), pp. 357–366 (cited on p. 146).

de Ayala, R. J. (2009). *The Theory and Practice of Item Response Theory*. Guilford Publications (cited on p. 64).

Dietterich, T. G. (1998). "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms". *Neural Computation*, 10, pp. 1895–1923 (cited on pp. 44, 89, 92).

Dittmar, C., Cano, E., Abeßer, J., and Grollmisch, S. (2012). "Music Information Retrieval Meets Music Education". In *Multimodal Music Processing*. Ed. by M. Müller, M. Goto, and M. Schedl. Dagstuhl Follow-Ups vol. 3. Dagstuhl Publishing, pp. 95–120 (cited on p. 29).

Dixon, S., Gouyon, F., and Widmer, G. (2004). "Towards Characterisation of Music via Rhythmic Patterns". In *Proc. 5th International Society for Music Information Retrieval Conference (ISMIR'04)*. Barcelona, Spain, pp. 509–517 (cited on pp. 62, 126, 129).

Doshi-Velez, F. and Kim, B. (2017). "Towards a Rigorous Science of Interpretable Machine Learning". arXiv preprint *arXiv:1702.08608* (cited on pp. 34, 63).

Downie, J. S., ed. (2003a). *The MIR/MDL Evaluation Project White Paper Collection*. 3rd edition. Available online at `http://www.music-ir.org/evaluation/wp.html` (cited on p. 35).

Downie, J. S. (2003b). "Toward the Scientific Evaluation of Music Information Retrieval Systems". In *Proc. 4th International Society for Music Information Retrieval Conference (ISMIR'03)*. Baltimore, MD, USA (cited on pp. 34, 35, 40).

Downie, J. S. (2004). "The Scientific Evaluation of Music Information Retrieval Systems: Foundations and Future". *Computer Music Journal*, 28(2), pp. 12–23 (cited on p. 35).

Downie, J. S., Ehmann, A. F., Bay, M., and Jones, M. C. (2010). "The Music Information Retrieval Evaluation eXchange: Some Observations and Insights". In *Advances in Music Information Retrieval*. Ed. by Z. W. Raś and A. A. Wieczorkowska. Studies in Computational Intelligence vol. 274. Springer, pp. 93–115 (cited on pp. 36, 54).

Drummond, C. (2006). "Machine Learning as an Experimental Science (Revisited)". In *Proc. AAAI'06 Workshop on Evaluation for Machine Learning*. Boston, MA, USA (cited on pp. 54, 89).

Drummond, C. (2008). "Finding a Balance Between Anarchy and Orthodoxy". In *Proc. Evaluation Methods for Machine Learning Workshop at the 25th International Conference on Machine Learning (ICML'08)*. Helsinki, Finland (cited on p. 89).

Drummond, C. (2009). "Replicability is not Reproducibility: Nor is it Good Science". In *Proc. Evaluation Methods for Machine Learning Workshop at the 26th International Conference on Machine Learning (ICML'09)*. Montreal, Canada (cited on p. 53).

Drummond, C. and Japkowicz, N. (2010). "Warning: Statistical Benchmarking Is Addictive. Kicking the Habit in Machine Learning". *Journal of Experimental & Theoretical Artificial Intelligence*, 22(1), pp. 67–80 (cited on p. 41).

Efron, B. (1977). "Bootstrap Methods: Another Look at the Jackknife". *The Annals of Statistics*, 7(1), pp. 1–26 (cited on pp. 44, 133).

Efron, B. (1983). “Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation”. *Journal of the American Statistical Association*, 78(382), pp. 316–331 (cited on pp. 45, 133).

Efron, B. and Tibshirani, R. (1997). “Improvements on Cross-Validation: The .632+ Bootstrap Method”. *Journal of the American Statistical Association*, 92(438), pp. 548–560 (cited on pp. 45, 133, 137).

Eugster, M. J. A. (2011). “Benchmark Experiments. A Tool for Analyzing Statistical Learning Algorithms”. PhD thesis. München, Germany: Ludwig-Maximilians-Universität München (cited on pp. 70, 89, 94, 161, 169).

Fisher, R. A. (1935). *The Design of Experiments*. Oliver & Boyd (cited on pp. 60, 66, 68, 197).

Flach, P. (2012). *Machine Learning*. Cambridge University Press (cited on p. 44).

Flexer, A. (2006). “Statistical Evaluation of Music Information Retrieval Experiments”. *Journal of New Music Research*, 32(2), pp. 113–120 (cited on pp. 54, 57).

Flexer, A. (2007). “A Closer Look on Artist Filters for Musical Genre Classification”. In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 341–344 (cited on p. 61).

Flexer, A. and Grill, T. (2016). “The Problem of Limited Inter-rater Agreement in Modelling Music Similarity”. *Journal of New Music Research*, 45(3), pp. 239–251 (cited on pp. 42, 58).

Flexer, A. and Schnitzer, D. (2010). “Effects of Album and Artist Filters in Audio Similarity Computed for Very Large Music Databases”. *Computer Music Journal*, 34(3), pp. 20–28 (cited on pp. 59, 60, 126).

Gómez, E. (2006). “Tonal Description of Polyphonic Audio for Music Content Processing”. *INFORMS Journal on Computing*, 18(3), pp. 294–304 (cited on pp. 32, 33).

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). “Explaining and Harnessing Adversarial Examples”. In *Proc. International Conference on Learning Representations (ICLR) 2015* (cited on p. 61).

Gouyon, F., Sturm, B. L., Oliveira, J. L., Hespanhol, N., and Langlois, T. (2014). "On Evaluation Validity in Music Autotagging". *arXiv preprint arXiv:1410.0001* (cited on p. 51).

Großmann, H. (2014). "Automating the Analysis of Variance of Orthogonal Designs". *Computational Statistics and Data Analysis*, 40, pp. 1–18 (cited on pp. 196, 229).

Gu, S. and Rigazio, L. (2014). "Towards Deep Neural Network Architectures Robust to Adversarial Examples". In *NIPS Workshop on Deep Learning and Representation Learning*. Montreal, Canada (cited on p. 61).

Guaus, E. (2009). "Audio Content Processing for Automatic Music Genre Classification: Descriptors, Databases, and Classifiers". PhD thesis. Universitat Pompeu Fabra, Barcelona, Spain (cited on p. 48).

Gupta, C., Li, H., and Wang, Y. (2018). "A Technical Framework for Automatic Perceptual Evaluation of Singing Quality". *APSIPA Transactions on Signal and Information Processing*, 7(e10) (cited on p. 34).

Hand, D. J. (1994). "Deconstructing Statistical Questions". *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 157(3), pp. 317–356 (cited on p. 60).

Hand, D. J. (2006). "Classifier Technology and the Illusion of Progress". *Statistical Science*, 21(1), pp. 1–15 (cited on p. 89).

Hand, D. J. (2012). "Assessing the Performance of Classification Methods". *International Statistical Review*, 80(3), pp. 400–414 (cited on pp. 55, 89).

Hand, D. J. (2018). "Aspects of Data Ethics in a Changing World: Where Are We Now?" *Big Data*, 6(3), pp. 176–190 (cited on p. 59).

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. 2nd edition. Springer (cited on pp. 39, 43, 44, 56, 133).

Haunschmid, V., Chowdhury, S., and Widmer, G. (2019). "Two-level Explanations in Music Emotion Recognition". *arXiv preprint arXiv:1905.11760* (cited on p. 63).

Heaven, D. (2019). "Why Deep-Learning AIs Are So Easy to Fool". *Nature*, 574, pp. 163–166 (cited on pp. 21, 199).

Hernández-Orallo, J. (2017). "Evaluation in Artificial Intelligence: From Task-Oriented to Ability-Oriented Measurement". *Artificial Intelligence Review*, 48(3), pp. 397–447 (cited on pp. 34, 35, 64).

Hernández-Orallo, J. (2019). "Gazing into Clever Hans Machines". *Nature Machine Intelligence*, 1(4), pp. 172–173 (cited on pp. 59, 199).

Herrera, P., Serrà, J., Laurier, C., Guaus, E., Gómez, E., and Serra, X. (2009). "The Discipline Formerly Known as MIR". In *10th International Society for Music Information Retrieval Conference (ISMIR'09). Special Session on The Future of MIR (fMIR)*. Kobe, Japan (cited on p. 28).

Hinkelmann, K. (2015). "History and Overview of Design and Analysis of Experiments". In *Handbook of Design and Analysis of Experiments*. Ed. by A. Dean, M. Morris, J. Stufken, and D. Bingham. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press. Chap. 1, pp. 3–62 (cited on pp. 67, 81).

Horton, R. L. (1978). *The General Linear Model*. McGraw-Hill (cited on p. 69).

Hothorn, T., Leisch, F., Zeileis, A., and Hornik, K. (2005). "The Design and Analysis of Benchmark Experiments". *Journal of Computational and Graphical Statistics*, 14(3), pp. 675–699 (cited on pp. 44, 89, 94, 133).

Hu, X. and Kando, N. (2012). "User-centered Measures vs. System Effectiveness in Finding Similar Songs". In *Proc. 13th International Society for Music Information Retrieval Conference (ISMIR'12)*. Porto, Portugal, pp. 331–336 (cited on p. 57).

Hu, X., Lee, J. H., Bainbridge, D., Choi, K., Organisciak, P., and Downie, J. S. (2017). "The MIREX Grand Challenge: A Framework of Holistic User-experience Evaluation in Music Information Retrieval". *Journal of the Association for Information Science and Technology*, 68(1), pp. 97–112 (cited on p. 34).

Hu, X. and Liu, J. (2010). "Evaluation of Music Information Retrieval: Towards a User-centered Approach". In *4th Workshop on Human-Computer Interaction and Information Retrieval (HCIR'10)*. New Brunswick, NJ, USA (cited on p. 58).

Humphrey, E. J. and Bello, J. P. (2012). "Rethinking Automatic Chord Recognition with Convolutional Neural Networks". In *Proc. 2012 11th International Conference on Ma-*

chine Learning and Applications (ICMLA). Vol. 2. Boca Raton, Florida, USA, pp. 357–362 (cited on p. 61).

Humphrey, E. J., Bello, J. P., and Lecun, Y. (2013). "Feature Learning and Deep Architectures: New Directions for Music Informatics". *Journal of Intelligent Information Systems*, 41(3), pp. 461–481 (cited on p. 33).

Ioannidis, J. P. A. (2005). "Why Most Published Research Findings Are False". *PLoS Medicine*, 2(8), pp. 696–701 (cited on p. 57).

Jamain, A. and Hand, D. J. (2008). "Mining Supervised Classification Performance Studies: A Meta-Analytic Investigation". *Journal of Classification*, 25, pp. 87–112 (cited on p. 55).

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms. A Classification Perspective*. Cambridge University Press (cited on pp. 45, 54, 56, 92, 94).

Jillings, N., Moffat, D., De Man, B., Reiss, J. D., and Stables, R. (2016). "Web Audio Evaluation Tool: A Framework for Subjective Assessment of Audio". In *Web Audio Conference (WAC-2016)*. Atlanta, GA, USA (cited on p. 34).

Jones, K. S., ed. (1981). *Information Retrieval Experiment*. Newton, MA, USA: Butterworth-Heinemann (cited on pp. 35, 54, 89).

Kaufman, S., Rosset, S., and Perlich, C. (2011). "Leakage in Data Mining: Formulation, Detection, and Avoidance". In *Proc. 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'11)*. San Diego, CA, USA, pp. 556–563 (cited on pp. 126, 127).

Kendall, M. G. (1938). "A New Measure of Rank Correlation". *Biometrica*, 30(1–2), pp. 81–89 (cited on p. 150).

Kereliuk, C., Sturm, B. L., and Larsen, J. (2015). "Deep Learning and Music Adversaries". *IEEE Transactions on Multimedia*, 17(11), pp. 2059–2071 (cited on pp. 61, 104, 108).

Kim, J. W. and Bello, J. P. (2019). "Adversarial Learning for Improved Onsets and Frames Music Transcription". In *Proc. 20th International Society for Music Information Retrieval Conference (ISMIR'19)*. Delft, The Netherlands, pp. 620–627 (cited on p. 61).

Klapuri, A. (2009). “A Method for Visualizing the Pitch Content of Polyphonic Music Signals”. In *Proc. 10th International Society for Music Information Retrieval Conference (ISMIR'09)*. Kobe, Japan, pp. 615–620 (cited on p. 41).

Korzeniowski, F. and Widmer, G. (2017). “End-to-end Musical Key Estimation using a Convolutional Neural Network”. In *Proc. 25th European Signal Processing Conference (EUSIPCO 2017)*, pp. 966–970 (cited on p. 33).

Kuznetsova, A., Brockhoff, P., and Christensen, R. B. (2017). “lmerTest Package: Tests in Linear Mixed Effects Models”. *Journal of Statistical Software*, 82(13), pp. 1–26 (cited on p. 221).

Lalor, J. P., Wu, H., and Yu, H. (2016). “Building an Evaluation Scale using Item Reponse Theory”. In *Proc. 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, TX, USA, pp. 648–657 (cited on p. 64).

Lalor, J. P., Wu, H., and Yu, H. (2019). “Learning Latent Parameters without Human Reponse Patterns: Item Response Theory with Artificial Crowds”. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing*. Hong Kong, China, pp. 4240–4250 (cited on p. 64).

Langley, P. (1988). “Machine Learning as an Experimental Science”. *Machine Learning*, 3(1), pp. 5–8 (cited on pp. 54, 55, 89).

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). “Unmasking Clever Hans Predictors and Assessing what Machines Really Learn”. *Nature Communications*, 10(1). Article number: 1096 (cited on pp. 59, 199).

Law, E. L. M. (2008). “The Problem of Accuracy as an Evaluation Criterion”. In *Proc. Evaluation Methods for Machine Learning Workshop at the 25th International Conference on Machine Learning (ICML'08)*. Helsinki, Finland (cited on p. 55).

Law, E. L. M., Von Ahn, L., Dannenberg, R. B., and Crawford, M. (2007). “TagATune: A Game for Music and Sound Annotation”. In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 361–364 (cited on p. 42).

Lee, C.-H., Shih, J.-L., Yu, K.-M., and Lin, H.-S. (2009). "Automatic Music Genre Classification Based on Modulation Spectral Analysis of Spectral and Cepstral Features". *IEEE Transactions on Multimedia*, 11(4), pp. 670–682 (cited on pp. 49, 51).

Lee, J. H. (2010). "Crowdsourcing Music Similarity Judgments using Mechanical Turk". In *Proc. 11th International Society for Music Information Retrieval Conference (ISMIR'10)*. Utrecht, The Netherlands, pp. 183–188 (cited on p. 42).

Lee, J. H., Hu, X., Choi, K., and Downie, J. S. (2015). "MIREX Grand Challenge 2014 User Experience: Qualitative Analysis of User Feedback". In *Proc. 16th International Society for Music Information Retrieval Conference (ISMIR'15)*. Málaga, Spain, pp. 779–785 (cited on p. 35).

Lehner, B., Sonnleitner, R., and Widmer, G. (2013). "Towards Light-Weight, Real-Time-Capable Singing Voice Detection". In *Proc. 14th International Conference on Music Information Retrieval (ISMIR'13)*. Curitiba, Brazil, pp. 53–58 (cited on p. 63).

Li, T. L. H. and Chan, A. B. (2011). "Genre Classification and the Invariance of MFCC Features to Key and Tempo". In *International Conference on MultiMedia Modeling*. Taipei, Taiwan, pp. 317–327 (cited on p. 61).

Lipton, Z. C. (2016). "The Mythos of Model Interpretability". In *Proc. ICML Workshop on Human Interpretability in Machine Learning* (cited on p. 34).

Lipton, Z. C. and Steinhardt, J. (2018). "Troubling Trends in Machine Learning Scholarship". *arXiv preprint arXiv:1807.03341* (cited on p. 63).

Lubberhuizen, W. (2010). *Near Perfect Reconstruction Polyphase Filterbank*. Available online at `https://www.mathworks.com/matlabcentral/fileexchange/15813-near-perfect-reconstruction-polyphase-filterbank` (cited on p. 104).

Mallat, S. (2012). "Group Invariant Scattering". *Communications on Pure and Applied Mathematics*, LXV, pp. 1331–1398 (cited on pp. 30, 49, 98).

Mandel, M. I. and Ellis, D. P. W. (2007). "A Web-Based Game for Collecting Music Metadata". In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 365–366 (cited on p. 42).

Marques, G., Domingues, M. A., Langlois, T., and Gouyon, F. (2011). "Three Current Issues in Music Autotagging". In *Proc. 12th International Society for Music Information Retrieval Conference (ISMIR'11)*. Miami, FL, USA, pp. 795–800 (cited on p. 128).

Martínez-Plumed, F., Prudêncio, R. B. C., Martínez-Usó, A., and Hernández-Orallo, J. (2016). "Making Sense of Item Response Theory in Machine Learning". In *Proc.European Conference on Artificial Intelligence*. The Hague, Netherlands (cited on p. 64).

Martínez-Plumed, F., Prudêncio, R. B. C., Martínez-Usó, A., and Hernández-Orallo, J. (2019). "Item Response Theory in AI: Analysing Machine Learning Classifiers at the Instance Level". *Artificial Intelligence*, 271, pp. 18–42 (cited on p. 64).

Mason, R. L., Gunst, R. F., and Hess, J. L. (2003). *Statistical Design and Analysis of Experiments with Applications to Engineering and Science*. Wiley (cited on pp. 67, 68).

Mauch, M. and Ewert, S. (2013). "The Audio Degradation Toolbox and its Application to Robustness Evaluation", pp. 83–88 (cited on p. 62).

McFee, B., Humphrey, E. J., and Bello, J. P. (2015a). "A Software Framework for Musical Data Augmentation". In *Proc. 16th International Society for Music Information Retrieval Conference (ISMIR'15)*. Málaga, Spain, pp. 248–254 (cited on p. 61).

McFee, B., Humphrey, E. J., and Urbano, J. (2016). "A Plan for Sustainable MIR Evaluation". In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 285–291 (cited on pp. 36, 42, 59).

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015b). "`librosa`: Audio and Music Signal Analysis in Python". In *Proc. 14th Python in Science Conference*. Austin, TX, USA, pp. 18–25 (cited on p. 33).

McKay, C. and Fujinaga, I. (2006). "Musical Genre Classification: Is It Worth Pursuing and How Can It Be Improved?" In *Proc. 7th International Conference on Music Information Retrieval (ISMIR'06)*. Victoria, BC, Canada, pp. 101–106 (cited on p. 58).

McNemar, Q. (1947). "Note on the Sampling Error of the Difference between Correlated Proportions or Percentages". *Psychometrika*, 12(2), pp. 153–157 (cited on p. 92).

Mendelson, A. F., Zuluaga, M. A., Lorenzi, M., Hutton, B. F., and Ourselin, S. (2017). "Selection Bias in the Reported Performances of AD Classification Pipelines". *NeuroImage: Clinical*, 14, pp. 400–416 (cited on p. 126).

Mishra, S., Sturm, B. L., and Dixon, S. (2017). "Local Interpretable Model-Agnostic Explanations for Music Content Analysis". In *Proc. 18th International Society for Music Information Retrieval Conference (ISMIR'17)*. Suzhou, China, pp. 537–543 (cited on pp. 34, 63).

Mishra, S., Sturm, B. L., and Dixon, S. (2018a). ""What are You Listening to?": Explaining Predictions of Deep Machine Listening Systems". In *Proc. 26th European Signal Processing Conference (EUSIPCO'18)*. Rome, Italy, pp. 2274–2278 (cited on p. 63).

Mishra, S., Sturm, B. L., and Dixon, S. (2018b). "Understanding a Deep Learning Machine Listening Model through Feature Inversion". In *Proc. 19th International Society for Music Information Retrieval Conference (ISMIR'18)*. Paris, France, pp. 755–762 (cited on p. 63).

Montgomery, D. C. (2013). *Design and Analysis of Experiments*. 8th edition. John Wiley and Sons (cited on pp. 55, 60, 67, 70, 130, 155).

Müller, M., Ellis, D. P. W., Klapuri, A., and Richard, G. (2011). "Signal Processing for Music Analysis". *IEEE Journal of Selected Topics in Signal Processing*, 5(6), pp. 1088–1110 (cited on p. 30).

Nelder, J. A. and Wedderburn, R. W. M. (1972). "Generalized Linear Models". *Journal of the Royal Statistical Society. Series A*, 135, pp. 370–384 (cited on p. 71).

Nembrini, S., König, I. R., and Wright, M. N. (2018). "The Revival of the Gini Importance?" *Bioinformatics*, 34(21), pp. 3711–3718 (cited on p. 115).

Nguyen, A., Yosinski, J., and Clune, J. (2015). "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images". In *Proc. 2015 Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, MA, USA (cited on pp. 56, 199).

Niedermayer, B., Böck, S., and Widmer, G. (2011). "On the Importance of "Real" Audio Data for MIR Algorithm Evaluation at the Note-Level: A Comparative Study". In

Proc. 13th International Society for Music Information Retrieval Conference (ISMIR'12). Porto, Portugal, pp. 543–548 (cited on p. 41).

Page, K. R., Fields, B., De Roure, D., Crawford, T., and Downie, J. S. (2013). "Capturing the Workflows of Music Information Retrieval for Repeatability and Reuse". *Journal of Intelligent Information Systems*, 41(3), pp. 435–459 (cited on p. 53).

Page, K. R., Nurmikko-Fuller, T., Rindfleisch, C., Weigl, D. M., Lewis, R., Dreyfus, L., and De Roure, D. (2015). "A Toolkit for Live Annotation of Opera Performance: Experiences Capturing Wagner's Ring Cycle". In *Proc. 16th International Conference on Music Information Retrieval (ISMIR'15)*. Málaga, Spain, pp. 211–217 (cited on p. 42).

Pálmason, H., Jónsson, B. Þ., Schedl, M., and Knees, P. (2017). "Music Genre Classification Revisited: An In-depth Examination Guided by Music Experts". In *International Symposium on Computer Music Multidisciplinary Research (CMRR'17)*, pp. 49–62 (cited on p. 58).

Pampalk, E., Flexer, A., and Widmer, G. (2005). "Improvements of Audio-Based Similarity and Genre Classification". In *Proc. 6th International Society for Music Information Retrieval Conference (ISMIR'05)*. London, UK, pp. 628–633 (cited on pp. 59, 60, 108, 127, 161).

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. 2nd edition. Cambridge University Press (cited on pp. 55, 128).

Pearl, J. (2014). "Comment: Understanding Simpson's Paradox". *The American Statistician*, 68(1), pp. 8–13 (cited on p. 153).

Peeters, G. (2004). *A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project*. Tech. rep. IRCAM (cited on p. 33).

Peeters, G. and Port, K. (2012). "Towards a (Better) Definition of the Description of Annotated MIR Corpora". In *Proc. 13th International Society for Music Information Retrieval Conference (ISMIR'12)*. Porto, Portugal, pp. 25–30 (cited on p. 40).

Peng, R. D. (2011). "Reproducible Research in Computational Science". *Science*, 334(6060), pp. 1226–1227 (cited on p. 53).

Perner, P. (2011). "How to Interpret Decision Trees?" In *11th International Conference on Advances in Data Mining: Applications and Theoretical Aspects*. New York City, NY, USA, pp. 40–55 (cited on p. 115).

Pfungst, O., Stumpf, C., Rahn, C. L., and Angell, J. R. (1911). "Clever Hans (the Horse of Mr. von Osten): A Contribution to Experimental, Animal, and Human Psychology". *Journal of Philosophy, Psychology and Scientific Methods*, 8(24), pp. 663–666 (cited on p. 59).

Pikrakis, A. (2013). "A Deep Learning Approach to Rhythm Modeling with Applications". In *Proc. International Workshop Machine Learning and Music of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013)*. Prague, Czech Republic (cited on p. 62).

Pons, J., Nieto, O., Prockup, M., Schmidt, E. M., Ehmann, A. F., and Serra, X. (2018). "End-to-end Learning for Music Audio Tagging at Scale". In *Proc. 19th International Society for Music Information Retrieval Conference (ISMIR'18)*. Paris, France, pp. 637–644 (cited on p. 33).

Popper, K. (1959). *The Logic of Scientific Discovery*. Hutchinson (cited on p. 199).

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W. (2014). "`mir_eval`: A Transparent Implementation of Common MIR Metrics". In *Proc. 15th International Society for Music Information Retrieval Conference (ISMIR'14)*. Taipei, Taiwan, pp. 367–372 (cited on pp. 45, 57, 59, 60, 196).

Ramona, M., Richard, G., and David, B. (2008). "Vocal Detection in Music with Support Vector Machines". In *Proc. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Las Vegas, NV, USA, pp. 1885–1888 (cited on p. 41).

Reichardt, C. S. (2011). "Criticisms of and an Alternative to the Shadish, Cook, and Campbell Validity Typology". In *Advancing Validity in Outcome Evaluation: Theory and Practice. New Directions for Evaluation*. Ed. by H. T. Chen, S. I. Donaldson, and M. M. Mark. Vol. 130, pp. 43–53 (cited on p. 53).

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In *Proc. 22nd ACM SIGKDD International Conference*

on Knowledge Discovery and Data Mining (KDD'16). San Francisco, CA, USA, pp. 1135–1144 (cited on p. 63).

Roads, C. (1996). *The Computer Music Tutorial*. Cambridge, MA, USA: MIT Press (cited on p. 28).

Rodríguez-Algarra, F., Sturm, B. L., and Dixon, S. (2019). "Characterising Confounding Effects in Music Classification Experiments through Interventions". *Transactions of the International Society for Music Information Retrieval*, 2(1), pp. 52–66 (cited on pp. 30, 125).

Rodríguez-Algarra, F., Sturm, B. L., and Maruri-Aguilar, H. (2016). "Analysing Scattering-Based Music Classification Systems: Where's the Music?" In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 344–350 (cited on pp. 31, 98, 130, 138, 144, 152, 161).

Salzberg, S. L. (1999). "On Comparing Classifiers: A Critique of Current Research and Methods". *Data Mining and Knowledge Discovery*, 1(1), pp. 1–12 (cited on p. 89).

Schedl, M., Flexer, A., and Urbano, J. (2013). "The Neglected User in Music Information Retrieval". *Journal of Intelligent Information Systems*, 41(3), pp. 523–539 (cited on pp. 29, 34, 58).

Schedl, M., Gómez, E., and Urbano, J. (2014). "Music Information Retrieval; Recent Developments and Applications". *Foundations and Trends in Information Retrieval*, 8(2-3), pp. 127–261 (cited on pp. 28, 29, 39, 41).

Schlüter, J. (2016). "Learning to Pinpoint Singing Voice from Weakly Labeled Examples". In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 44–50 (cited on p. 63).

Schlüter, J. and Grill, T. (2015). "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks". In *Proc. 16th International Society for Music Information Retrieval Conference (ISMIR'15)*. Málaga, Spain, pp. 114–120 (cited on p. 61).

Serrà, J. (2007). "A Qualitative Assessment of Measures for the Evaluation of a Cover Song Identification System". In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 319–322 (cited on p. 57).

Serra, X., Magas, M., Benetos, E., Chudy, M., Dixon, S., Flexer, A., Gómez, E., Gouyon, F., Herrera, P., Jordá, S., Paytuvi, O., Peeters, G., Schlüter, J., Vinet, H., and Widmer, G. (2013). *Roadmap for Music Information Research*. The MIReS Consortium (cited on pp. 28–30, 33, 34).

Seyerlehner, K., Widmer, G., and Knees, P. (2010). "A Comparison of Human, Automatic and Collaborative Music Genre Classification and User Centric Evaluation of Genre Classification Systems". In *International Workshop on Adaptive Multimedia Retrieval (AMR'10)*. Linz, Austria, pp. 118–131 (cited on p. 57).

Shadish, W. R., Cook, T. D., and Campbell, D. T. (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Boston, MA, USA: Houghton Mifflin Company (cited on pp. 53–56, 155).

Shuster, J. and Eys, J. van (1983). "Interaction Between Prognostic Factors and Treatment". *Controlled Clinical Trials*, 4(3), pp. 209–214 (cited on p. 169).

Siedenburg, K., Fujinaga, I., and McAdams, S. (2016). "A Comparison of Approaches to Timbre Descriptors in Music Information Retrieval and Music Psychology". *Journal of New Music Research*, 45(1), pp. 1–15 (cited on pp. 49, 59).

Sigtia, S. and Dixon, S. (2014). "Improved Music Feature Learning with Deep Neural Networks". In *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy, pp. 6959–6963 (cited on p. 33).

Simpson, E. H. (1951). "The Interpretation of Interaction in Contingency Tables". *Journal of the Royal Statistical Society, Series B*, 13, pp. 238–241 (cited on p. 153).

Six, J., Bressan, F., and Leman, M. (2018). "A Case for Reproduciblity in MIR: Replication of 'A Highly Robust Audio Fingerprinting System'". *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 1(1), pp. 56–67 (cited on p. 53).

Skowronek, J., McKinney, M. F., and Van De Par, S. (2007). "A Demonstrator for Automatic Music Mood Estimation". In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 345–346 (cited on p. 44).

Sommet, N. and Morselli, D. (2017). “Keep Calm and Learn Multilevel Logistic Modeling: A Simplified Three-Step Procedure Using Stata, R, Mplus, and SPSS”. *International Review of Social Psychology*, 30(1), pp. 203–218 (cited on p. 190).

Stober, S. (2013). “Adaptive Methods for User-Centered Organization of Music Collections”. PhD thesis. Magdeburg, Germany: Otto-von-Guericke University (cited on p. 29).

Stoller, D., Ewert, S., and Dixon, S. (2018). “Adversarial Semi-supervised Audio Source Separation Applied to Singing Voice Extraction”. In *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada, pp. 2391–2395 (cited on p. 61).

Stowell, D., Petrusková, T., Šálek, M., and Linhart, P. (2019). “Automatic Acoustic Identification of Individuals in Multiple Species: Improving Identification across Recording Conditions”. *Journal of the Royal Society Interface*, 16(153). Article number: 20180940 (cited on p. 128).

Sturm, B. L. (2012a). “A Survey of Evaluation in Music Genre Recognition”. In *International Workshop on Adaptive Multimedia Retrieval*. Copenhagen, Denmark, pp. 29–66 (cited on p. 40).

Sturm, B. L. (2012b). “An Analysis of the GTZAN Music Genre Dataset”. In *Proc. 2nd International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies (MIRUM’12)*. Nara, Japan (cited on p. 47).

Sturm, B. L. (2013a). “Classification Accuracy Is Not Enough”. *Journal of Intelligent Information Systems*, 41(3), pp. 371–406 (cited on p. 58).

Sturm, B. L. (2013b). “Evaluating Music Emotion Recognition: Lessons from Music Genre Recognition?” In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. San Jose, CA, USA, pp. 1–6 (cited on pp. 36, 58).

Sturm, B. L. (2013c). “The GTZAN Dataset: Its Contents, its Faults, their Effects on Evaluation, and its Future Use”. *arXiv preprint arXiv:1306.1461* (cited on pp. 47, 48, 51, 58, 108, 109, 123).

Sturm, B. L. (2014a). "A Simple Method to Determine if a Music Information Retrieval System Is a "Horse"". *IEEE Transactions on Multimedia*, 16(6), pp. 1636–1644 (cited on pp. 59, 62, 98, 104, 126).

Sturm, B. L. (2014b). "A Survey of Evaluation in Music Genre Recognition". In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*. Ed. by A. Nürnberger, S. Stober, B. Larsen, and M. Detyniecki. Lecture Notes in Computer Science vol. 8382. Springer, pp. 29–66 (cited on p. 36).

Sturm, B. L. (2014c). "Making Explicit the Formalism Underlying Evaluation in Music Information Retrieval Research: A Look at the MIREX Automatic Mood Classification Task". In *Sound, Music, and Motion*. Ed. by M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad. Lecture Notes in Computer Science vol. 8905. Springer, pp. 89–104 (cited on p. 58).

Sturm, B. L. (2014d). "The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval". *Journal of New Music Research*, 43(2), pp. 147–172 (cited on pp. 30, 36, 47, 58, 61, 98, 108, 139, 141, 150).

Sturm, B. L. (2016a). "Revisiting Priorities: Improving MIR Evaluation Practices". In *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR'16)*. New York City, NY, USA, pp. 1–32 (cited on pp. 33, 59, 60, 126).

Sturm, B. L. (2016b). "The "Horse" Inside: Seeking Causes of the Behaviours of Music Content Analysis Systems". *Computers in Entertainment, Special Issue on Musical Metacreation*, 14(2), pp. 488–494 (cited on pp. 33, 59, 62, 98, 105, 130, 195).

Sturm, B. L., Bardeli, R., Langlois, T., and Emiya, V. (2014). "Formalizing the Problem of Music Description". In *Proc. 15th International Society for Music Information Retrieval Conference (ISMIR'14)*. Taipei, Taiwan, pp. 89–94 (cited on pp. 29, 30).

Sturm, B. L. and Collins, N. (2014). "The Kiki-Bouba Challenge: Algorithmic Composition for Content-Based MIR Research and Development". In *Proc. 15th International Society for Music Information Retrieval Conference (ISMIR'14)*. Taipei, Taiwan, pp. 21–26 (cited on p. 41).

Sturm, B. L., Kereliuk, C., and Larsen, J. (2015). "*'El Caballo Viejo'*: Latin Genre Recognition with Deep Learning and Spectral Periodicity". In *Mathematics and Computation in*

Music. Ed. by T. Collins, D. Meredith, and A. Volk. Lecture Notes in Computer Science vol. 9119. Springer (cited on p. 59).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). "Intriguing Properties of Neural Networks". In *Proc. International Conference on Learning Representations (ICLR) 2014*. Banff, Canada (cited on p. 61).

Tague-Sutcliffe, J. (1992). "The Pragmatics of Information Retrieval Experimentation, Revisited". *Information Processing & Management*, 28(4), pp. 467–490 (cited on pp. 35, 54).

Taylor, B. N. and Kuyatt, C. E. (1994). *NIST Technical Note 1297: Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results*. Available online at `https://www.nist.gov/pml/nist-technical-note-1297`. Last accessed: 17-07-2019 (cited on p. 53).

Trochim, W. M. K. and Donnelly, J. P. (2007). *The Research Methods Knowledge Base*. 3rd edition. Atomic Dog (cited on pp. 51, 53, 56).

Turnbull, D., Liu, R., Barrington, L., and Lanckriet, G. R. (2007). "A Game-Based Approach for Collecting Semantic Annotations of Music". In *Proc. 8th International Conference on Music Information Retrieval (ISMIR'07)*. Vienna, Austria, pp. 535–538 (cited on p. 42).

Tzanetakis, G. and Cook, P. (2002). "Musical Genre Classification of Audio Signals". *IEEE Transactions on Speech and Audio Processing*, 10(5), pp. 293–301 (cited on pp. 33, 41, 44, 47, 48, 98, 116, 123, 126, 136).

Urbano, J. (2011). "Information Retrieval Meta-Evaluation: Challenges and Opportunities in the Music Domain". In *Proc. 12th International Society for Music Information Retrieval Conference (ISMIR'12)*. Miami, FL, USA, pp. 609–614 (cited on p. 54).

Urbano, J., McFee, B., Downie, J. S., and Schedl, M. (2012). "How Significant is Statistically Significant? The Case of Audio Music Similarity and Retrieval". In *Proc. 13th International Society for Music Information Retrieval Conference (ISMIR'12)*. Porto, Portugal, pp. 181–186 (cited on p. 57).

Urbano, J. and Schedl, M. (2013). “Minimal Test Collections for Low-cost Evaluation of Audio Music Similarity and Retrieval Systems”. *International Journal of Multimedia Information Retrieval*, 2(1), pp. 59–70 (cited on p. 57).

Urbano, J., Schedl, M., and Serra, X. (2013). “Evaluation in Music Information Retrieval”. *Journal of Intelligent Information Systems*, 41(3), pp. 345–369 (cited on pp. 34, 36, 40, 41, 51, 54, 57, 227).

Vorhees, E. M. (2007). “TREC: Continuing Information Retrieval’s Tradition of Experimentation”. *Communications of the ACM*, 50(11), pp. 51–54 (cited on p. 35).

Wasserstein, R. L. and Lazar, N. A. (2016). “The ASA’s Statement on p-Values: Context, Process, and Purpose”. *The American Statistician*, 70(2), pp. 129–133 (cited on pp. 57, 190).

Weihs, C., Jannach, D., Vatolkin, I., and Rudolph, G., eds. (2017). *Music Data Analysis. Foundations and Applications*. CRC Press (cited on pp. 43, 131).

Widmer, G. (2016). “Getting Closer to the Essence of Music: The *Con Espressione* Manifesto”. *ACM Transactions on Intelligent Systems and Technology*, 8(2), Article 19 (cited on pp. 21, 58).

Wiering, F. (2009). “Meaningful Music Retrieval”. In *10th International Society for Music Information Retrieval Conference (ISMIR’09). Special Session on The Future of MIR (fMIR)*. Kobe, Japan (cited on p. 58).

Wierstorf, H., Ward, D., Mason, R., Grais, E. M., Hummersone, C., and Plumbley, M. D. (2017). “Perceptual Evaluation of Source Separation for Remixing Music”. In *143th Audio Engineering Society Convention*. Article 9880. New York City, NY, USA (cited on p. 34).

Wiggins, G. A. (2009). “Semantic Gap?? Schemantic Schmap!! Methodological Considerations in the Scientific Study of Music”. In *11th IEEE International Symposium on Multimedia (ISM’09)*. San Diego, CA, USA, pp. 477–482 (cited on pp. 21, 58).

Wolpert, D. H. and Macready, W. G. (1997). “No Free Lunch Theorems for Optimization”. *IEEE Transactions on Evolutionary Computation*, 1(1), pp. 67–82 (cited on p. 56).

Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). “Understanding Data Augmentation for Classification: When to Warp?” In *2016 International Conference*

on Digital Image Computing: Techniques and Applications (DICTA). Queensland, Australia (cited on p. 61).

Yang, Y.-H. and Chen, H. H. (2012). "Machine Recognition of Music Emotion: A Review". *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3), Article No. 40 (cited on p. 30).