Department of Computer Science

# Exploring classifier attribute interactions and time series using constrained randomisations

Andreas Henelius

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
**DOCTORAL DISSERTATIONS** 58/2017

# Exploring classifier attribute interactions and time series using constrained randomisations

**Andreas Henelius**

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall AS2 of the school on 5th May 2017 at 12 noon.

**Aalto University**
**School of Science**
**Department of Computer Science**

**Supervising professor**
Professor Aristides Gionis, Aalto University

**Thesis advisor**
Docent Kai Puolamäki, Finnish Institute of Occupational Health

**Preliminary examiners**
Prof. Dr. Stephan Günnemann, Technische Universität München, Germany
Docent Pauli Miettinen, Max-Planck-Institut für Informatik, Germany

**Opponent**
Professor Matthijs van Leeuwen, Leiden University, the Netherlands

Aalto University publication series
**DOCTORAL DISSERTATIONS** 58/2017



ISBN 978-952-60-7361-3 (printed)
ISBN 978-952-60-7360-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-7360-6

Unigrafia Oy
Helsinki 2017

Finland

441 697
Printed matter

**Author**
Andreas Henelius

**Name of the doctoral dissertation**
Exploring classifier attribute interactions and time series using constrained randomisations

**Publisher** School of Science

**Unit** Department of Computer Science

**Series** Aalto University publication series DOCTORAL DISSERTATIONS 58/2017

**Field of research** Computer and Information Science

**Date of the defence** 5 May 2017

**Permission to publish granted (date)** 6 March 2016 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

**Abstract**
Gaining insight into structures and properties in data is a central problem in data mining and knowledge discovery. This is essential when the data is to be used, e.g., in decision-making. In this thesis we consider investigating the structure of data in two cases: temporal structures in time series, and attribute interactions utilised by classifiers.

Time series are ubiquitous and represent an important type of data. We investigate temporal structures in time series, focusing on interval sequences. We seek explanations for observed properties by constructing and evaluating null hypotheses describing the internal properties of the time series. We approach this as a hypothesis testing problem where observed time series are compared to randomly generated instances. The properties being investigated are modelled in terms of constraints on the randomisations, allowing complex relationships to be examined and explained. Furthermore, we apply computational methods in the analysis of a sleep study to explain the relationship between time series representing heart rate variability and performance on a psychomotor vigilance test.

Classification has wide applicability in multiple domains, however, many high-performing classifiers are essentially opaque, black-box algorithms, making it difficult to gain insight into the basis for predictions. In classifier analysis we consider attribute interactions utilised by classifiers. An interaction means that two or more attributes jointly carry information with respect to, e.g., a class label. We study two different types of interactions. Firstly, we investigate relationships between attributes in a dataset and show how this is related to factorising the class-conditional joint data distribution, such that attributes in the same factor are interacting while attributes in different factors are independent, given the class. We devise a method for testing the hypothesis that a dataset originates from a generating distribution with a particular factorised form. Secondly, we investigate how classifiers exploit attribute interactions in making predictions and develop a novel framework based on constrained randomisations for partitioning the attributes of a dataset into groups based on how they are jointly exploited by the classifier. The methods developed here are useful in several data analysis applications, e.g., in enhancing the interpretability of opaque classifiers, detecting adverse drug interactions in pharmacovigilance, anonymising data and gaining insight into the structure of datasets.

**Keywords** Constrained randomisations, significance testing, time series, modelling, classifiers, classifier analysis, attribute interactions

**ISBN (printed)** 978-952-60-7361-3 **ISBN (pdf)** 978-952-60-7360-6

**ISSN-L** 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

**Location of publisher** Helsinki **Location of printing** Helsinki **Year** 2017

**Pages** 205 **urn** http://urn.fi/URN:ISBN:978-952-60-7360-6

Aalto-universitetet

# Sammandrag

**Aalto-universitetet, PB 11000, 00076 Aalto** www.aalto.fi

**Författare**
Andreas Henelius

**Doktorsavhandlingens titel**
Begränsade randomiseringar för undersökning av attributinteraktioner i klassificerare samt av tidsserier

**Utgivare** Högskolan för teknikvetenskaper

**Enhet** Institutionen för datavetenskap

**Seriens namn** Aalto University publication series DOCTORAL DISSERTATIONS 58/2017

**Forskningsområde** Informationsteknik

**Datum för disputation** 05.05.2017

**Beviljande av publiceringstillstånd (datum)** 06.03.2016 **Språk** Engelska

☐ **Monografi** ☒ **Artikelavhandling** ☐ **Essäavhandling**

**Sammandrag**

Ett centralt problem inom data- och kunskapsutvinning är hur man skall få insikt i datans egenskaper och struktur. Detta är viktigt då datan används t.ex. för beslutsfattande. I denna avhandling granskas strukturer i data i två fall: temporala strukturer i tidsserier, samt hur klassificerare använder sig av samverkan mellan attribut i datan.

Tidsserier är typiskt förekommande och utgör en viktig typ av data. Vi undersöker temporala strukturer i tidsserier, med fokus på intervallsekvenser. Vi skapar nollhypoteser som beskriver tidsseriens interna egenskaper och söker med hjälp av dessa förklaringar för observerade egenskaper. Vi närmar oss detta som ett hypotestestningsproblem, där den observerade tidsserien jämförs mot slumpmässigt skapade instanser. Egenskaperna vi undersöker beskrivs genom begränsade randomiseringar, vilket gör det möjligt att undersöka och hitta förklaringar till komplexa förhållanden. Vidare använder vi beräkningsmetoder i analysen av data från en sömnstudie, för att förklara förhållandet mellan tidsserier beskrivande hjärtfrekvensvariabilitet och psykomotorisk vaksamhet.

Klassificering har ett brett tillämpningsområde inom olika områden, men ett problem är att många av de mest effektiva klassificerarna är opaka svarta lådor, vilket gör det är svårt att få inblick i hur förutsägelserna görs. Vi studerar här hur klassificerare utnyttjar attributinteraktioner i datan. En interaktion innebär att två eller fler attribut samverkar med avseende på t.ex. en klassvariabel. Vi betraktar två typer av interaktioner. Det första fallet utgörs av förhållandet mellan attribut i datamängden. Vi visar hur detta problem är relaterat till en faktorisering av attributens klassberoende simultanfördelning, så att attribut i samma faktor samverkar, medan attribut i olika faktorer är oberoende, med hänsyn till klassvariabeln. Vi presenterar en ny metod för att undersöka hypotesen att datan härstammar från en distribution med en särskild faktoriserad form. Det andra fallet gäller hur klassificerare utnyttjar attributinteraktioner för förutsägelser och vi presenterar en ny metod för att dela in attributen i grupper på basen av hur algoritmen utnyttjar deras samverkan. Metoden bygger på begränsad randomisering av datan. Metoderna vi utvecklat är generella och användbara inom analys av data och de möjliggör t.ex. förståelse av svårtolkade klassificerare, undersökning av samverkan mellan mediciner inom farmakovigilans, anonymisering av data samt bättre insikt i datans struktur.

**Nyckelord** Begränsade randomiseringar, signifikanstestning, tidsserier, modellering, klassificerare, analys av klassificerare, attributinteraktioner

| | | |
|---|---|---|
| **ISBN (tryckt)** 978-952-60-7361-3 | **ISBN (pdf)** 978-952-60-7360-6 | |
| **ISSN-L** 1799-4934 | **ISSN (tryckt)** 1799-4934 | **ISSN (pdf)** 1799-4942 |
| **Utgivningsort** Helsingfors | **Tryckort** Helsingfors | **År** 2017 |
| **Sidantal** 205 | **urn** http://urn.fi/URN:ISBN:978-952-60-7360-6 | |

# Preface

This work was carried out at the Finnish Institute of Occupational Health in Helsinki, Finland, in projects funded by Tekes (SalWe Research Programme for Mind and Body, and the Revolution of Knowledge Work project) and the Academy of Finland (Human Guided Data Analysis).

I would like to express my deepest gratitude to my thesis advisor, Dr. Kai Puolamäki, who has taught me a lot about data mining and statistics, introduced me to the world of randomisations and provided guidance throughout this work. I am also grateful to Prof. Aristides Gionis for supervising the thesis and giving excellent comments. I also thank the pre-examiners Dr. Pauli Miettinen and Prof. Dr. Stephan Günnemann for valuable and insightful comments.

I am grateful to my former boss Dr. Kiti Müller for giving me the chance to work in a cross-discplinary environment with motivated people. This work could not have been carried out without the help and collaboration of a large group of people, who have contributed with their knowledge.

I like to thank my colleague Jussi Korpela, together with whom we have navigated through the murky waters of data analysis and had many good conversations on various technical and non-technical issues over the years. I also like to thank Dr. Antti Ukkonen, working with whom I have learned a lot.

In the ReKnow project I had the opportunity to spend three months at Stockholm University and I want to express my gratitude to Prof. Panagiotis Papapetrou for hosting me. I wish to thank Prof. Henrik Boström and Prof. Lars Asker for many good ideas and discussions. I also thank my other co-authors at Stockholm University; Isak Karlsson and Jing Zhao.

A large thanks goes to Dr. Jussi Virkkala and Dr. Mikael Sallinen, who have shared their knowledge and domain expertise on sleep research, and to Dr. Minna Huotilainen who has always provided good ideas and comments.

I also want to thank all of my past and present colleagues at the Finnish Institute of Occupational Health. It is a privilege to work with so motivated and enthusiastic people from different fields and you all contribute to a nice working atmosphere. I like to thank Kristian Lukander, Kati Pettersson, Teppo Valtonen, Satu Pakarinen, Laura Sokka, Lauri Ahonen,

Jani Lukander, Sharman Jagadeesan, Marianne Leinikka, Miika Toivanen, Matti Gröhn, Jari Torniainen, Emilia Oikarinen, Benjamin Cowley and Virpi Kalakoski.

I wish to thank my friends, parents-in-law and sisters-in-law with families for many fun moments over the years.

My warmest thanks go to my parents Ghitta and Viking for encouraging me throughout the years.

Most of all, I want to thank my wife Kajsa for being supportive and understanding and our children Amanda and Alvar for providing joy and distraction.

Espoo, 27.03.2017

Andreas Henelius

# Contents

# List of Publications

This thesis consists of this introductory part and the following five publications.

**I** Henelius, A., Korpela, J., and Puolamäki, K. Explaining Interval Sequences by Randomization. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2013)*, pages 337–352. Springer, 2013

**II** Henelius, A., Sallinen, M., Huotilainen, M., Müller, K., Virkkala, J., and Puolamäki, K. Heart Rate Variability for Evaluating Vigilant Attention in Partial Chronic Sleep Restriction. *Sleep*, 37(7):1257–1267, 2014b

**III** Henelius, A., Puolamäki, K., Boström, H., Asker, L., and Papapetrou, P. A peek into the black box: Exploring classifiers by randomization. *Data Mining and Knowledge Discovery*, 28(5–6): 1503–1529, 2014a

**IV** Henelius, A., Puolamäki, K., Karlsson, I., Zhao, J., Asker, L., Boström, H., and Papapetrou, P. GoldenEye++: A Closer Look into the Black Box. In *Proceedings of the Third International Symposium on Statistical Learning and Data Sciences (SLDS 2015)*, pages 96–105. Springer, 2015

**V** Henelius, A., Puolamäki, K., and Ukkonen, A. Finding Statistically Significant Attribute Interactions. *Submitted to ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2017), arXiv e-prints, arXiv:1612.07597*, 2017

# 1. Introduction

## 1.1 Motivation

In this thesis we explore the structure of data in two particular cases: (i) structures in time series and (ii) attribute interactions utilised by classifiers.

The goal of knowledge discovery is to extract patterns from the data and finding structure from data is one aspect of this (Fayyad et al., 1996). Consequently a wide variety of data analysis methods have been developed for the purpose of inferring structures in different types of data in different fields. Examples of methods in machine learning include unsupervised techniques such as clustering for assessing the similarity of data items and supervised techniques such as classification for learning from examples to assign labels to data items. Other techniques, such as time series modelling, are used to study time-dependent relationships in the data.

Although the type of structures considered by these various techniques are different, the goal is the same: to gain insight into the data. This is important for instance in order to describe the underlying process or phenomenon generating the data. Understanding patterns in the data is also essential if the data is used as part of a decision-making process (e.g., Kleinberg et al., 1998). We now consider the two areas on which we focus in this thesis.

### Exploring Structural Properties in Time Series

Time series are ubiquitous in almost any field, such as economy, medicine and physics and hence there exists a large number of methods for their analysis. One way of gaining insight into the structural properties of

time series is through modelling. An important concept in modelling is the philosophical principle of parsimony, i.e., what is the simplest model that captures the essence of the data. In practice this means that when analysing the data we do not start by using complex models but instead gradually build up our understanding of the structure of the time series. A further aspect of this is to avoid overfitting the model to the data. Here structural property is used to mean any aspect of the time series that depends on the time order of the samples, e.g., the Fourier coefficients of the time series.

**Contributions** The main contributions in this thesis in this area are as follows. In Publication I we consider ways of representing event sequences as interval sequences, in the time, frequency and autocorrelation spaces. We devise and apply different constrained randomisation methods in the analysis of such sequences to find explanations for observed patterns in the data. We apply the methods to, e.g., biomedical signals in the form of heart interbeat-interval (IBI) sequences. In Publication II we apply computational methods in the analysis of a dataset from a study on cumulative sleep restriction. Specifically, we study the correlation between heart rate variability (HRV) and performance on the psychomotor vigilance task (PVT) (Dinges and Powell, 1985) in subjects undergoing sleep restriction and in control subjects. Using this approach we gain insight into the association between HRV, PVT and the factors in a three-process model of alertness. We now consider the second area in this thesis.

### Attribute Interactions Utilised by Classifiers

Prediction is an important part of machine learning and a wide variety of algorithms exist for solving classification problems. Much focus has been on developing algorithms that perform well, but a drawback of many modern high-performing machine learning algorithms such as, e.g., support vector machine (Cortes and Vapnik, 1995) or random forests (Breiman, 2001) is that they are complex models and essentially opaque *black boxes*. In many applications it is essential that the models used are transparent, i.e., interpretable. The notion of interpretability is complex, see, e.g., Lipton (2016), but one aspect of a transparent model is the possibility to gain insight into the basis for the predictions. This is especially important in fields such as medicine (Caruana et al., 2015; Krause et al., 2016) or

genomics (Jalali and Pfeifer, 2016). One way of describing how a classifier arrives at the predictions is by considering how the algorithm *exploits interactions* between attributes. We here use the term *interaction* to denote that two or more attributes carry complementary information regarding the class and are jointly needed for predicting the class of a data item. As an example, for a dataset with five attributes $a, b, c, d, e$, we want to find a partition of the attributes of the form $\mathcal{S} = \{\{a, b, c\}, \{d, e\}\}$. This grouping then reveals that the classifier exploits attributes $a, b, c$ jointly and similarly attributes $d, e$. A partition of the attributes of this form describes the structure of the data in terms of how a classifier perceives and exploits attribute relationships.

A second aspect of attribute interactions is to understand which interactions *are needed* in order to train a good classifier. In this case we are interested in the structure of the attributes in the dataset, rather than in how a classifier utilises attribute interactions for making predictions. Although these two types of attribute interaction structures are related, they are not always identical as a classifier might, e.g., overlearn structures which are not representative of the true interactions in the data. We here develop a novel method for exploring interaction structures in datasets in terms of finding a factorisation of the class-conditional joint distribution of the attributes with the help of a classifier. The method we develop is based on the observation that a classifier must internally at least implicitly model the joint data distribution in order to make predictions. Given a data matrix $X$ and an associated vector of class labels $C$, the goal of a classifier is to model the classes based on the data, i.e., the classifier considers a joint probability distribution of the form $\Pr(C \mid X) = \Pr(X \mid C) \Pr(C) / \Pr(X)$. Here $\Pr(X \mid C)$ is the class-conditional distribution of the attributes, i.e., how the values of each attribute are distributed with respect to the class label. Our goal is to find a factorisation of the joint data distribution such that attributes in the same factor are interacting, while attributes in different factors are independent.

Assume now that in the above example of a dataset with five attributes the classifier found and exploited the true interactions in the data, meaning that there are interactions between attributes $a, b, c$, and between $d, e$. In this case we want to find a factorisation of the form

$$\Pr(C \mid X) = \Pr(X(\cdot, S_1) \mid C) \Pr(X(\cdot, S_2) \mid C) \Pr(C) / \Pr(X), \qquad (1.1)$$

where $X(\cdot, S_1)$ and $X(\cdot, S_2)$ denote the part of $X$ corresponding to attributes $a, b, c$ and $d, e$, respectively. Also here we hence seek a disjoint partition of the attributes compactly describing the structure of the dataset. We further approach the finding of this factorisation as a significance testing problem with the goal of investigating the hypothesis that an observed dataset has been sampled from a particular factorised distribution. We now exemplify the importance of attribute interactions as a means of interpreting the data. We first consider attribute interactions exploited by a classifier.

**Example 1.** Consider a dataset containing data measured from sensors connected to a machine on a factory production line. This dataset contains operational metrics such as temperature and pressure readings of various components. We also have a binary indicator added by the machine operator signalling whether the machine is operating correctly or if it is stopped due to a failure. We want to train a classifier that can predict when the machine is about to fail, so that the machine can be shut down in time. However, we would also like to *understand why* the machine needs to be shut down, i.e., we would like to know how the classifier perceives that different sensor readings are related to machine failure. Assume now that we have a method that outputs the attributes used *jointly* by the classifier in predicting failure. We then understand, e.g., that higher than average values of pressure of valve 1, temperature of gearbox 2 and rotational speed of shaft 2 *together* lead to failure, but not if the components on their own exceed some threshold value. We can also, e.g., notice that certain metrics are unnecessary as they are not indicative of failure and are consequently not used by the classifier. By gaining insight into the structure of the data in terms of determining which factors are important for the classification we might here, e.g., better understand the underlying causes for machine failure.

In the next example we consider the usefulness of attribute interactions in terms of factorising the joint data distribution.

**Example 2.** The goal of anonymisation is to ensure privacy, meaning that individual data items in the original dataset should not be present in the anonymised dataset. However, the anonymised data should share

properties with the original dataset such that it can be used in a similar manner. One method of data anonymisation is based on permuting the data. Assume that we have knowledge of the factorised form of the class-conditional joint data distribution as in Equation (1.1), where attributes in the same factor are dependent and all factors are independent. This allows us to permute the data in a principled way, taking the known attribute interactions into account. In Equation (1.1) we can anonymise the data by breaking the interaction between the two independent attribute groups $S_1 = \{a, b, c\}$ and $S_2 = \{d, e\}$ without affecting the class-conditional joint distribution, in which case the classification performance on the anonymised data will also be similar.

**Contributions** The main contributions in this thesis in this area are as follows. In Publication III we introduce the novel `GoldenEye` algorithm for exploring attribute interactions used by opaque classifiers. The `GoldenEye` algorithm is the first generic method for exploring attribute interactions used by classifiers. As noted by Duivesteijn and Thaele (2014) regarding Publication III: "*A very recent first inroad towards peeking into the classifier black box is the method by Henelius et al. [31], who strive to find groups of attributes whose interactions affect the predictive performance of a given classifier.*" We extend the work in Publication IV in the form of the `GoldenEye++` algorithm applied in pharmacovigilance to detect drug-drug interactions. In Publication V we present the novel ASTRID method for automatically finding a factorisation of the class-conditional joint distribution of a dataset. The method approaches the problem from a significance testing perspective and uses classifiers to investigate attribute interactions.

The two main research topics in this thesis are hence concerned with exploring the structure of data. Significance testing is important in both these topics and we hence briefly consider this next.

### Significance Testing of Structures

Empirical significance testing is a fundamental method for exploring structures in data and essentially consists of three steps: (1) a null hypothesis regarding the structure, (2) choice of a discriminating statistic, and (3) sampling values of the discriminating statistic under the null hypothesis.

The value of the statistic on the original data is compared to an ensemble of values sampled from the distribution under the null hypothesis, providing an empirical $p$-value describing the significance of the structure being investigated.

It is, in the general case, difficult to analytically obtain the distribution of the discriminating statistic under the null hypothesis, especially if the null hypothesis is complex, which is the case in the themes considered here. Example of a relevant complex null hypotheses is the sampling of time series with a particular autocorrelation structure or sampling of datasets following a certain class-conditional joint distribution. Direct sampling of data with these properties is not possible since the distributions are unknown. We hence need alternative methods to solve the sampling problem.

Different resampling and randomisation schemes have been widely used in data analysis (e.g., Davison and Hinkley, 1997; Efron and Tibshirani, 1994; Good, 2005, 2006). Sampling methods based on randomisation and resampling can also be used to generate data with the desired properties, such that the value of the discriminating statistic evaluated from the synthetic data follows the distribution under the null hypothesis. In this thesis we consider the use of *constrained randomisation* to generate such data for the purpose of exploring the structure of time series and attribute interactions.

### Constrained Randomisation

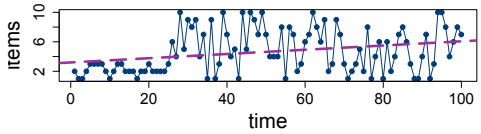
Constrained randomisation means that some property of an existing dataset is randomly permuted subject to specific constraints. The constraints are used to impose particular structural properties on the data. Different sets of constraints hence represent different null hypotheses regarding the data.

Constrained randomisation methods are appealing for the generation of synthetic data since they do not require that a model be fit to the existing data and in many cases it is convenient to parametrise the structural properties in terms of the constraints. Constrained randomisation schemes are hence flexible in terms of describing different types of data structures and are applicable in the analysis of various types of data.

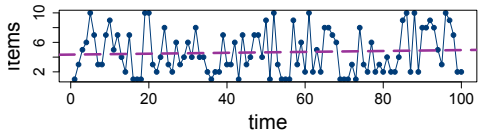
We now present an example of how the method of constrained ran-

domisation is used to investigate the presence of interesting structures in time series. We here define an *interesting structure* as a structure that is significantly different from structures found in random data, i.e., it is unlikely that the structure could occur randomly. An interesting structure is hence *non-random*.

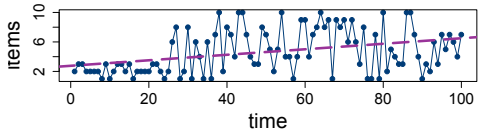
**Example 3.** Imagine that the machine discussed in Example 1 is producing items with a maximum capacity of 10 items per minute. As a means of quality control we record the number of units produced for the first 100 minutes the machine is running, shown in Figure 1.1a together with the linear trend shown in lilac. We assume that the number of units produced randomly varies between 1 and 10, depending on supply of raw materials. If the number of units is uniformly distributed we expect the slope of the linear trend to be close to zero. However, there clearly is a linearly increasing trend in the data. To evaluate the significance of this trend, we utilise the above described significance testing framework. We randomly shuffle the observations in the observed time series, generating 1000 random samples such as the one in Figure 1.1b, calculate the slope of the trend line and compare the slope of the original observed time series to the distributions of slopes. This is shown in Figure 1.2a, yielding a (two-tailed) $p$-value significant on the 5% level, i.e., the observed slope represents a non-random structure. We now ask, is the trend explained by the first 25 observations, during which the average number of items produced appears to be lower? To test this hypothesis we generate samples by constrained randomisation such that the first 25 samples are permuted within themselves independently from the following 75 observations, thus retaining the overall temporal structure of the original time series. An example of an instance of a time series obtained by constrained randomisation is shown in Figure 1.1c. The histogram and $p$-value are shown in Figure 1.2b and we can observe that the slope is no longer significant. We hence conclude that the first 25 observations constitute a non-random structure and explain the temporal structure of the time series.




**(a)** Original time series

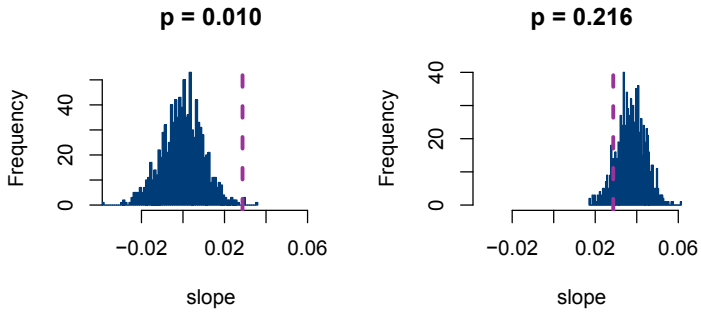



**(b)** Permutation of time series




**(c)** Constrained permutation of time series.

**Figure 1.1:** Randomised version of the observed time series. The linear trend of the data is shown in lilac.




**(a)** Random permutation **(b)** Constrained permutation

**Figure 1.2:** Histograms of the the distribution of linear slopes using different permutations. The vertical dashed line shows the value of the slope of the original data and the $p$-value describes the significance of the observed slope.

## 1.2 Publications and Contributions

In this section we present the publications that form the basis for this thesis and describe the author's contributions.

### Publication I

**Henelius, A., Korpela, J., and Puolamäki, K. Explaining Interval Sequences by Randomization. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2013)*, pages 337–352. Springer, 2013**

In Publication I we consider the problem of explaining the structure of interval sequences. To this end we present different representations of interval sequences; in addition to the typical time domain representation the sequences can also be represented and treated in the Fourier and autocorrelation spaces. We devise different randomisation schemes for interval sequences that preserve different aspects of the structure in the event series. The randomisation schemes represent different null models for the data and are used to calculate confidence intervals, with the help of which non-random structures in the data can be found. We perform experiments on synthetic and real datasets and show that we can capture different interesting phenomena in the interval sequences that allow us to interpret their structure.

All authors participated in the formulation of the initial problem and all authors contributed equally to the publication. The present author was responsible for the experiments, analysis, interpretation and reporting of data related to heart interbeat intervals. The present author was also responsible for the MCMC-based constrained randomisation in collaboration with Kai Puolamäki.

### Publication II

**Henelius, A., Sallinen, M., Huotilainen, M., Müller, K., Virkkala, J., and Puolamäki, K. Heart Rate Variability for Evaluating Vigilant Attention in Partial Chronic Sleep Restriction. *Sleep*, 37(7):1257–1267, 2014b**

In this publication we apply computational methods to investigate spectral heart rate variability (HRV) metrics in the measurement of vigilant

attention in chronic partial sleep restriction. We extend previous work by Chua et al. (2012) who reported a correlation between a particular HRV frequency band and reaction times on the psychomotor vigilance task (PVT). We perform an exhaustive search over different HRV frequency bands to determine the frequency band exhibiting the highest correlation between HRV power and PVT results. We also study the correlation between the HRV and PVT time series during the course of the day, over multiple days and subjects. Aggregated results are presented using Bootstrap confidence intervals. We further perform an analysis of the association between spectral HRV metrics and PVT reaction times by constructing a simple three-process model of alertness. By variance partitioning we investigate the cause of the correlation. We find that both HRV and PVT contain processes that are not described by the model and that HRV explains 33% of the variation in PVT. We conclude that HRV spectral power in the $[0.01, 0.08]$ Hz band shows a high correlation with PVT lapses in conditions involving sleep restriction.

The initial idea of Publication II was derived in collaboration with Mikael Sallinen. The present author performed all data analysis and interpretation, and wrote most of the manuscript. The statistical methods were chosen in collaboration with Kai Puolamäki. All authors participated in providing ideas and in discussing, commenting and preparing the manuscript.

#### Publication III

**Henelius, A., Puolamäki, K., Boström, H., Asker, L., and Papapetrou, P. A peek into the black box: Exploring classifiers by randomization. *Data Mining and Knowledge Discovery*, 28(5–6):1503–1529, 2014a**

Publication III introduces the `GoldenEye` algorithm, a novel iterative method for investigating how a given classifier exploits dependencies between attributes in making predictions. The `GoldenEye` algorithm is generic and can be used to gain insight into the workings of any classifier, even opaque classifiers such as random forests of support vector machines. This algorithm represents a novel direction of improving the interpretability of black-box classifiers since there previously was no method for investigating attribute interactions for any classifier. We perform com-

prehensive experiments using 15 different classifiers on 26 datasets from the UCI machine learning repository (Lichman, 2013).

The research question was jointly formulated and the GoldenEye algorithm derived by all authors. The Methods section was written jointly by Kai Puolamäki and Panagiotis Papapetrou with participation from the present author. The GoldenEye algorithm was implemented by the present author in collaboration with Kai Puolamäki. The present author performed, interpreted and reported all experiments. All authors participated in discussing, commenting and writing the manuscript.

#### Publication IV

**Henelius, A., Puolamäki, K., Karlsson, I., Zhao, J., Asker, L., Boström, H., and Papapetrou, P. GoldenEye++: A Closer Look into the Black Box. In *Proceedings of the Third International Symposium on Statistical Learning and Data Sciences (SLDS 2015)*, pages 96–105. Springer, 2015**

Publication IV continues the theme of Publication III. Here the GoldenEye algorithm is applied in the analysis of a real medical dataset in order to detect adverse drug events (ADEs), i.e., unwanted reactions in a patient caused by a combination of multiple drugs. The task is to find combinations of drugs that potentially cause ADEs. We introduce a new goodness measure in the GoldenEye algorithm instead of fidelity to make the grouping process more sensitive. The new algorithm, termed GoldenEye++, uses the correlation between class membership probabilities for the original data and for a randomised version of the data. Using the GoldenEye++ algorithm and a random forest classifier we find medically relevant drug interactions.

The idea for the publication was proposed by the co-authors affiliated with the Department of Computer and Systems Sciences (DSV) at Stockholm University and was jointly developed with them. The present author implemented the necessary changes to the original GoldenEye algorithm, wrote most of the theoretical parts of the Methods, and performed, interpreted and reported all experiments related to synthetic data. The co-authors from DSV performed the experiments, interpretation and reporting related to the real medical data.

### Publication V

**Henelius, A., Puolamäki, K., and Ukkonen, A. Finding Statistically Significant Attribute Interactions.** ***Submitted to ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2017), arXiv e-prints, arXiv:1612.07597*****, 2017**

Publication V continues and extends our previous work on the topic of investigating attribute interactions used by classifiers for making predictions. We approach the problem of investigating attribute interactions from a significance testing perspective and present the novel ASTRID method for automatically finding a factorisation of the class-conditional joint data distribution. The ASTRID method is based on a principled hypothesis testing framework using which it is possible to investigate the hypothesis that a given dataset has been sampled from a particular factorised class-conditional data distribution. This method hence provides insight into the attribute interactions in the dataset, i.e., what interactions in the data are needed to train a useful classifier. This can viewed as a complement to the GoldenEye algorithm in Publication III, which investigates attribute interactions exploited by the classifier for making predictions.

The idea for the publication was jointly developed by the present author and Kai Puolamäki. All authors participated in providing ideas and discussing the topic of the paper. The randomisation scheme in ASTRID was jointly developed by all authors and the grouping step was proposed by Kai Puolamäki. The present author wrote most of the introduction and related work. The methods section was written by the author together with Kai Puolamäki. The present author implemented all necessary methods, and prepared, performed, interpreted and reported all experiments. All authors participated in the preparation of the final manuscript.

## 1.3 Outline

The rest of this thesis is organised into four chapters.

Chapter 2 is an introductory chapter discussing some of the key concepts in this thesis. This chapter introduces, e.g., supervised learning, significance testing and confidence intervals, constrained randomisations, and Markov Chain Monte Carlo methods.

Chapter 3 focuses on the themes of Publications I and II and discusses

the topic of exploring the structure of time series using constrained randomisations. We also discuss the use of resampling-based methods for robust estimation. We present relevant examples and discuss practical considerations regarding the use of the methods.

In Chapter 4 we discuss the use of constrained randomisations in exploring attribute interactions utilised by classifiers. We present the `GoldenEye` and `GoldenEye++` algorithms for investigating attribute interactions exploited by classifiers and discuss the ASTRID method for finding a factorisation of the class-conditional joint data distribution. We provide examples of how the methods developed in this chapter can be used in data analysis.

The relation between the research themes, publications, and Chapter 3 and Chapter 4 are visualised using a Venn diagram in Figure 1.3.

Finally, the topics covered in the introductory part of the thesis are summarised, and the main conclusions are presented and discussed in Chapter 5.

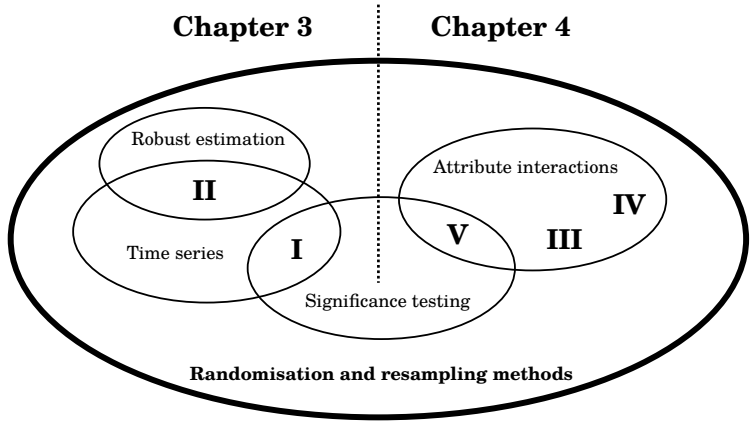



**Figure 1.3:** Venn diagram showing the relation between research themes, depicted as ovals, publications, denoted by Roman numerals, I–V, and the chapters of the thesis.

# 2. Background

This chapter is a review of the data mining, machine learning and statistical methods most central to this thesis.

## 2.1 Data types

The two main data types considered in this thesis are time series (Publications I and II) and data matrices (Publications III, IV and V).

**Time Series** A time series $S_t$ is an ordered sequence of $n$ data points defined as

$$S_t = (x_1, x_2, \ldots, x_n), \quad x_i \in \mathbb{R}^d, \tag{2.1}$$

where $d$ denotes the dimension of the time series. If $d = 1$ the time series is *univariate*, e.g., a single biomedical signal, and if $d > 1$ the time series is *multivariate*, in which case $S_t$ is an ensemble of time series. Each data point is associated with a particular occurrence time given by the vector

$$T = (t_1, t_2, \ldots, t_n), \ t_i \leq t_j \text{ for all } i, j.$$

**Interval Sequence** Given a sequence of $n$ events $a_1, \ldots, a_n$ occurring at the time instances $t_1, \ldots, t_{n+1}$, an interval sequence $S_i$ of length $n - 1$ is

$$S_i = (y_1, y_2, \ldots, y_{n-1}), \quad y_n = t_{n+1} - t_n, \tag{2.2}$$

i.e., each event $y_n$ in the interval sequence represents the duration (time interval) between two consecutive events.

**Data Matrix** A typical data matrix $X$ considered in classification problems is an $n \times m$ matrix with $n$ observations and $m$ attributes (features) in some

space $\mathcal{X}$. The values for the $m$ attributes can be, e.g., ordinal (quantitative variables) or categorical (qualitative variables).

## 2.2 Supervised Learning

Classification is one of of the most common machine learning tasks and belongs to the field of *supervised learning*. In supervised learning a set of labelled training data is used to construct a classifier able to predict the class of previously unseen data items, i.e., the learning process is guided (Hastie et al., 2009; Mohri et al., 2012). Methods for analysing *attribute interactions* used by classifiers are presented in Publications III, IV and V. We next define the classification task more formally.

Let $X$ be an $n \times m$ data matrix with $n$ observations of $m$ attributes from some space $\mathcal{X}$. One row from $X$ constitutes a data item and we denote the $i$th data item in $X$ as $x_i = X(i, \cdot)$.

Each data item $x_i$ is associated with a class label $c_i \in C$ from a finite set of labels $C$ and we let $C = (c_1, \ldots, c_n)$ be a vector of class attributes where the $i$th item corresponds to the class of the $i$th data item in $X$. The classifier function is now defined as follows.

**Classifier function** A *classifier* is a function $f$ that maps data items $x \in \mathcal{X}$ to class labels, i.e., $f : \mathcal{X} \mapsto C$.

When training a classifier the original dataset is typically divided into three parts: (1) a training set, (2) a validation set and (3) a test set (Hastie et al., 2009). The training set is used to train the classifier and the performance on the validation set is used for model selection, i.e., to tune the classifier. Finally, the test set is never used in the learning process and is reserved for analysing the generalisation error of the final model. The classifier is hence trained using a multiset of items from $\mathcal{X}$ (i.e., items may be repeated in the training set), and the classifier generalises to items from $\mathcal{X}$ not used for training.

### 2.2.1 Goodness of Classification

In order to evaluate the accuracy of the classification task we must employ a goodness measure. Classification accuracy is a commonly used measure.

**Accuracy** Let $C$ be the true class labels for a data matrix $X$ with $n$ items and let $C'$ be the class labels predicted by a classifier $f$. The accuracy $a$ of $f$ is then $a = \frac{1}{n}\sum_{i=1}^{n} I\left[C_i = C_i'\right]$, i.e., the accuracy is the fraction of correct predictions. Here $I$ denotes the indicator function, i.e., $I[\square] = 1$ if $\square$ is true, otherwise the value of $I$ is zero.

### 2.2.2 Classifiers

We here present briefly three classifiers used in Publications III, IV and V. The naïve Bayes classifier is an example of a widely used simple probabilistic classifier while the support vector machine (SVM) and random forest (RF) classifiers have been found to consistently be among the best-performing supervised learning method (see Fernández-Delgado et al., 2014).

**Naïve Bayes** The naïve Bayes is a probabilistic classifier which assumes that different attributes in the dataset are independent given the class (Hastie et al., 2009). Given the set of $k$ class labels $C = (C_1, C_2, \ldots, C_k)$ and a data vector $\mathrm{x} = (x_1, x_2, \ldots, x_m)$ the naïve Bayes model is

$$\mathrm{P}(C_k \mid \mathrm{x}) \propto \prod_{i=1}^{m} \mathrm{P}(x_i \mid C_k)\,. \tag{2.3}$$

Since the naïve Bayes classifier assumes that the attributes are independent it cannot utilise relationships such as, e.g., correlations, between attributes. However, it has been found that the naïve Bayes classifier works surprisingly well also in many situations where the assumption of independence of the attributes is violated (Domingos and Pazzani, 1997).

**Support Vector Machines** The support vector machine classifier (Cortes and Vapnik, 1995) belongs to the category of kernel methods. The main idea of support vector machines (SVMs) is to use hyperplanes as decision boundaries. The SVM uses a mapping function, the kernel, to transform the data to a higher-dimensional space where the data in different classes can be, at least approximately, linearly separated by a hyperplane. The boundaries of these hyperplanes are formed by support vectors, i.e., the points nearest to the decision surfaces are the support vectors. The hyperplanes in the SVM classifier are chosen so that the separation between classes is maximised (see, e.g., Boser et al., 1992 and Burges, 1998).

**Random forest** The random forest classifier (Breiman, 2001) is an ensemble classifier, in which a "forest" of decision trees are constructed and the final class is determined by a majority vote among all trees. Each tree in the forest is created using samples obtained from the training set by bootstrap aggregation (bagging, see Breiman, 1996). Features are chosen at random in the trees and the trees are not pruned.

## 2.3 Significance Testing

The goal of significance testing, or hypothesis testing, is to determine how likely it is that some observed test statistic is due to chance. The significance of a result is typically reported in the form of a $p$-value, $p \in (0, 1]$, at a particular *significance level* $\alpha \in (0, 1]$. A test is declared significant at the $\alpha$ level if $p \leq \alpha$. Significance testing forms an important part of the discussion in Publications I (significance of properties of time series) and V (significance of attribute interactions utilised by classifiers).

**Distribution of p-Values Under the Null Hypothesis** Consider a sample space $X$. The $p$-value is a random variable (Murdoch et al., 2008) that denotes the probability under the null hypothesis $H_0$ of obtaining a result at least as extreme as observed in the original data, i.e.,

$$p(x_{\text{obs}}) = \Pr(T \geq t_{\text{obs}} \mid H_0), \tag{2.4}$$

where $T(X)$ is the value of the test statistic under the null hypothesis $H_0$ and $t_{\text{obs}} = T(x_{\text{obs}})$ is the value of the test statistic computed from the observed data $x_{\text{obs}} \in X$. It also holds that $\Pr(p(X) \leq \alpha \mid H_0) \leq \alpha$, i.e., the distribution of $p$-values under the null hypothesis is stochastically not smaller than the standard uniform distribution (see, e.g., Casella and Berger, 2001).

**Hypothesis Testing** Hypothesis testing proceeds as follows:

1. A null hypothesis is defined (e.g., that the means of two groups are equal).

2. A test statistic is defined and its value is computed for the observed sample.

3. The value of the computed test statistic is compared to its distribution under the null hypothesis and the significance is determined in the form of a $p$-value as discussed above.

The distribution of the test statistic under the null hypothesis can be obtained *parametrically* if we know that the test statistic under the null hypothesis follows some particular distribution, e.g., the standard normal distribution. In the *nonparametric* case the distribution is unknown and different sampling and resampling schemes must be used in the sampling process (Good, 2002, 2005).

Permutation tests have been widely used in statistics, e.g., Fischer's exact test for the analysis of contingency tables. The samples in the permutation test do not necessarily have to be independent, but they must be exchangeable, i.e., the joint distribution of the observations is preserved under permutation of their order (see, e.g., Good, 2002 and Good, 2005). The permutation testing proceeds as follows.

1. Calculate the value of the test statistic $\theta_0$ from the observed dataset $D$.

2. Generate, using some suitable method, $R$ simulated datasets distributed according to the null hypothesis: $\{D_1^*, D_2^*, \ldots D_R^*\}$.

3. Calculate the value of the test statistic for each of the $R$ surrogate datasets, giving the set $T = \{\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots \hat{\theta}_R^*\}$.

The (one-tailed) empirical $p$-value is now obtained as (Davison and Hinkley, 1997):

$$p = \frac{1 + I\left[\hat{\theta}_i^* \geq \theta_0\right]}{1 + R}, \tag{2.5}$$

where $I$ is the indicator function. The $p$-value reflects the proportion of surrogate datasets in which the test statistic has a more extreme value than in the observed dataset, which is equivalent to our definition of the $p$-value in Equation (2.4).

The topic of how to generate the simulated data needed for the hypothesis testing is important and the used method must typically be tailored to the particular problem. Refer to Davison and Hinkley (1997) and Good (2005) for in-depth discussion of Monte Carlo tests and resampling-based

tests. Generation of simulated data for hypothesis testing using randomisation methods has been applied in Publications I and V and is discussed in more detail in Chapter 3 and Chapter 4.

**Number of Samples** The number of samples $R$ in Equation (2.5) affects the minimum obtainable $p$-value, which is $1/(1+R)$. Using, e.g., $R = 250$ samples a minimum value of $p = 0.004$ can be obtained.

**Multiple Correction Testing** An important topic in significance testing is the issue regarding correction of multiple testing. There are two main errors that can be made in statistical significance testing of a null hypothesis, usually referred to as *Type I* and *Type II* errors. The Type I error describes the probability of falsely rejecting a true null hypothesis (false positive) and this occurs at a rate corresponding to the significance level, i.e., $\alpha$. The *Type II* error describes the probability of failing to reject a false null hypothesis (false negative). The rate at which a Type II error is made, typically denoted by $\beta$, depends on the power of the used statistical test.

When testing a hypothesis, there is hence a probability of $\alpha$ to wrongly state that there is an effect when there is none (Type I error). Hence, when testing $m > 1$ hypotheses, there is a probability of $1 - (1-\alpha)^m$ that at least one result is erroneously declared significant due to chance (Westfall and Young, 1993). With *multiple correction* is meant the procedure of controlling the rate of Type I errors when performing multiple tests. There are different approaches for controlling the rate of Type I error in multiple testing, e.g., *familywise error rate* (FWER) which is the probability of at least one Type I error, and the *false discovery rate* (FDR) which is the proportion of falsely rejected hypotheses among all rejected hypotheses, i.e., the proportion of Type I errors among all rejected hypotheses.

Several methods have been developed for controlling the FWER and the FDR. E.g., to control the FWER the following methods are applicable: the Bonferroni correction or the Holm method (Holm, 1979). For FDR control, e.g. the following methods can be used: the Benjamini-Hochberg (Benjamini and Hochberg, 1995) or the permutation-based method of Westfall and Young (Westfall and Young, 1993).

**P-values as Heuristics** In this thesis $p$-values are also used as heuristics for model selection in Section 4.8.1. In such cases correction for multi-

ple testing is omitted and the $p$-values should not be interpreted as true significance tests.

## 2.4 The Bootstrap

The bootstrap (Efron, 1979) is a sampling-based method for estimating standard errors of statistics. The main idea of the bootstrap is to approximate the sampling distribution of a given statistic using a single sample with a limited number of observations, instead of sampling new datasets from the population.

Let $\theta$ be some statistic of interest concerning an unknown distribution $F$ and let $X = (x_1, x_2, \ldots, x_n)$ be a sample from $F$. Let $\hat{\theta}$ be an estimate of $\theta$ calculated from $X$. The bootstrap can be used as follows to determine the standard error of $\hat{\theta}$, i.e., how much $\hat{\theta}$ differs from $\theta$.

A bootstrap sample $X^* = \left(x_i^*, x_2^*, \ldots, x_n^*\right)$ is obtained by sampling with replacement from the empirical distribution $\hat{F}$, in which each sample in $X$ has equal weight. Some items in $X^*$ may hence be repeated. The size of the bootstrap sample equals the size of the original data sample $X$. A set of $B$ bootstrap samples are created and the value of $\hat{\theta}$ is calculated for each of these samples, yielding the bootstrap distribution $\hat{F}_{\mathrm{B}} = \left(\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^*\right)$, where each $\hat{\theta}_i^*$ is called a bootstrap replicate. Finally, the standard error of $\hat{\theta}$ is calculated as the standard deviation of the bootstrap replicates in $\hat{F}_{\mathrm{B}}$ (Efron and Tibshirani, 1994).

For an in-depth treatment of the bootstrap refer to Efron (1979); Efron and Tibshirani (1986) and Efron and Tibshirani (1994), and for a wide coverage of application areas for methods based on the bootstrap refer to Davison and Hinkley (1997). Several bootstrap techniques are also reviewed by Davison et al. (2003). The bootstrap is used in Publications I and II for estimating confidence intervals.

## 2.5 Confidence Intervals

Confidence intervals are used to describe the amount of uncertainty associated with a parameter $\theta$ and are of the form $\left[\hat{\theta}_{\mathrm{lower}}, \hat{\theta}_{\mathrm{upper}}\right]$. The intervals are constructed so that the interval covers the true, unknown value of a parameter $\theta$ with a probability of $1 - \alpha$, where $\alpha$ is the confidence level, typically, e.g., 0.05 corresponding to a 95% confidence interval.

Confidence intervals for parameters following a particular distribution, e.g., the exponential or standard normal distribution, can be constructed analytically. For parameters with an unknown sampling distribution nonparametric bootstrap confidence intervals can be constructed from the bootstrap distribution using several methods, such as the percentile method or the bias-corrected and accelerated ($BC_a$)-method by Efron (1987). Also see, e.g., Carpenter and Bithell (2000) for a discussion of different types of bootstrap confidence intervals.

In the percentile method, the bootstrap distribution is formed using $B$ replicates. The replicates are rank-ordered:

$$\left(\hat{\theta}^1, \hat{\theta}^2, \ldots, \hat{\theta}^B\right),$$

where the superscript denotes the rank of the replicate. The limits for the two-sided $100 \cdot (1-\alpha)\%$ confidence interval are then given by the replicates with indices $B(\alpha/2)$ and $B(1-\alpha)/2$ as the lower and upper limits, respectively (Carpenter and Bithell, 2000; Efron and Tibshirani, 1994).

The percentile bootstrap confidence intervals are wider than necessary (Good, 2005) and the use of, e.g., $BC_a$-confidence intervals (Efron, 1987) is hence advocated as these are more robust (Good, 2005).

### 2.5.1 Interpretation of Confidence Intervals

Confidence intervals (parametric and nonparametric) can also be used for hypothesis testing since the values outside the confidence intervals at a confidence level of $\alpha$ correspond to the rejection region of a significance test (see, e.g., Good, 2005). However, when using confidence intervals for testing hypothesis by comparing their overlap one should note that this procedure is more conservative than standard methods of significance testing. See, e.g., Payton et al. (2003); Schenker and Gentleman (2001) and Cumming and Finch (2005), for discussion of this issue and how confidence intervals may be adjusted to account for this bias.

The interesting problem of estimating confidence intervals for time series data has been studied by Korpela et al. (2014).

## 2.6 Constrained Randomisation

As discussed in Section 2.3, in empirical significance testing it is necessary to obtain samples of the test statistic under the null hypothesis. However,

direct sampling from this distribution is not always possible, since it may not be possible to express it analytically. In such cases it is necessary to generate surrogate datasets using, e.g., randomisation methods. The distribution of values of the test statistic calculated from the surrogate datasets follows the distribution of the test statistic under the null hypothesis.

Randomisation methods have been applied in several different contexts such as, e.g., in statistics (Good, 2005; Westfall and Young, 1993), to generate binary (Gionis et al., 2007) and real (Ojala et al., 2008) matrices with fixed margins, to analyse classifiers (Ojala and Garriga, 2010), in the study of graphs (Hanhijärvi et al., 2009a; Ying and Wu, 2009), in iterative data mining (De Bie, 2011; Hanhijärvi et al., 2009b), in pattern mining (Lijffijt et al., 2014) and to investigate time series (Henelius et al., 2013; Schreiber, 1998; Schreiber and Schmitz, 2000; Theiler and Prichard, 1996; Vuokko and Kaski, 2011).

The basic idea in constrained randomisation is to generate a new dataset such that it shares certain properties of interest with the original dataset, but is otherwise random. We next formalise this.

Let $D_0$ be an observed dataset from the space $\mathcal{D}$ of all datasets of this type and let $S(D_0)$ be some structural property of interest determined from $D_0$. We here define *constrained randomisation* as the process of generating datasets from $D_0$ using a randomisation function, such that the generated datasets share the structural property $S$ with $D_0$. The property $S$ hence acts as a *constraint* on the sampling process and defines a probability distribution over $\mathcal{D}$. The constraints can be either *crisp* or *soft*. With crisp constraints the property $S$ is exactly the same for the sampled datasets, whereas for soft constraints the property $S$ is only approximately the same. The property $S$ hence limits the sample space to subsets $\mathcal{D}_S \subseteq \mathcal{D}$ where the datasets meet the constraint.

In this thesis different constrained randomisation schemes are used in the investigation of time series (Chapter 3) and in investigating attribute interactions used by classifiers (Chapter 4).

## 2.7 Markov Chain Monte Carlo Sampling

As noted above, in some cases it is necessary to sample from complex distributions that cannot be analytically described. Markov Chain Monte Carlo methods can be used in this case.

Markov Chain Monte Carlo (MCMC) is a class of statistical sampling algorithms designed to allow sampling from a target distribution $\pi(\mathrm{x})$ of interest from which direct sampling is impossible. The only requirement is that the target distribution must be known up to a normalising constant (Liu, 2008), i.e., it suffices that we know a distribution $f$ to which the target distribution $\pi$ is proportional up to an unknown constant $c$ such that

$$f(\mathrm{x}) = c \cdot \pi(\mathrm{x}). \tag{2.6}$$

The basic principle in MCMC sampling is to construct a *Markov chain* having the target distribution $\pi(\mathrm{x})$ as its stationary distribution. A first-order Markov chain $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$ is a sequence of dependent random variables $\mathrm{x}_i \in \mathcal{X}$ in the state space $\mathcal{X}$ such that the transitions from the state $i$ to the state $j$ are described by a transition kernel $K(\mathrm{x}_i, \mathrm{x}_j)$ (Andrieu et al., 2003; Geyer, 2011).

The Metropolis-Hastings algorithm is used to construct a Markov Chain having the desired distribution $\pi$ as the stationary distribution. The algorithm has two steps (Geyer, 2011; Liu, 2008):

1. Propose a new state $\mathrm{x}'$ from a proposal distribution $q(\mathrm{x}' \mid \mathrm{x}_t)$ based on the current state $\mathrm{x}_t$.

2. Calculate the ratio $r = \frac{f(\mathrm{x}')q(\mathrm{x}'|\mathrm{x}_t)}{f(\mathrm{x})q(\mathrm{x}_t|\mathrm{x}')}$ and let

$$\mathrm{x}_{t+1} = \begin{cases} \mathrm{x}' & \text{with probability } \min(1, r) \\ \mathrm{x} & \text{otherwise.} \end{cases}$$

The ratio $r$ is referred to as the Metropolis-Hastings ratio. Note that if the proposal distribution $q$ is symmetric the calculation of $r$ only depends on $f$. The proposal states are generated by some small perturbation of the current state and hence the intuition behind the Metropolis-Hastings algorithm is that the sampling process will tend to move in the sample space towards regions where the proposed samples are more likely to come from the target distribution (Liu, 2008).

Several different MCMC sampling methods can be used. We next briefly consider Parallel Tempering Markov Chain Monte Carlo.

## 2.8 Parallel Tempering Markov Chain Monte Carlo

Parallel tempering MCMC sampling (or replica exchange Monte Carlo) (Geyer, 1991), is a versatile sampling method in which $M$ Markov chains are simulated in parallel, such that each replica is kept at a different *temperature* level. The tempered distribution from which sampling is performed hence has the form

$$\pi(S) \propto e^{-d(S)\beta}, \tag{2.7}$$

where $d$ is the non-negative distribution of interest and $\beta = (1/T) \in [\beta_{\min} = 0, \beta_{\max}]$ is the reciprocal temperature controlling the shape of the distribution. At high temperatures ($T \gg 1$ and $\beta \to 0$) the distribution is flat and the chain can efficiently move around in the state space. As the temperature is reduced ($T \to 0$ and $\beta \gg 1$) the distribution becomes more peaked and is centred around the minimum of the function $d(S)$ (Liu, 2008). The sample space is now given by the product space

$$\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_M, \tag{2.8}$$

where each factor is an identical copy of $\mathcal{X}$ (Liu, 2008) and samples are drawn from the joint distribution

$$\Pi(S_1, \ldots, S_M) \propto \prod_{i=1}^{M} \pi_i(S)\, e^{-d(S)\beta_i}. \tag{2.9}$$

Here, each factor in the distribution represents an MCMC chain kept at a different temperature: $\beta_1 = 0 \leq \beta_2 \leq \ldots \leq \beta_M = \beta$. The chain $S_M$ for which $\beta_M = \beta$ is referred to as the "cold chain" and is the chain of interest from which samples are drawn.

The main idea of parallel tempering is to allow exchanges between the different chains operating at different temperatures. The parallel tempering technique hence utilises two types of update steps (Liu, 2008): (i) within-chain updates (parallel step) and (ii) between-chain updates (swapping step). At each iteration either the parallel step, in which all chains are updated independently, or the swapping step, where two chains in the ensemble are swapped, is chosen with a given probability. The acceptance probability for each type of update is given by the Metropolis-Hastings ratio. The swapping process couples the chains in the ensemble such that the individual chains are no longer Markov chains, but the entire ensemble is still a Markov chain (Geyer, 1991).

# 3. Exploring the Structure of Time Series

In this chapter we consider exploring the temporal structure of time series. We first discuss the importance of investigating properties and patterns of time series, after which we present a hypothesis testing framework for investigating the structure of time series. The framework is based on sampling from an empirical data distribution representing the distribution of the data under the null hypothesis. We discuss different randomisation methods for generating samples corresponding to various null hypotheses and finally show some practical examples. We also consider the use of the bootstrap for estimating confidence intervals for data from a sleep study.

## 3.1 Introduction

A time series consists of a set of ordered events representing how some phenomena evolves over time. Time series are a fundamental type of data found in almost all domains: biosignals in medicine (Lehman et al., 2015), stock prices in economics (Kim, 2003) or measurements of oxygen and nutrients in oceanography (Michaels et al., 1994) to name a few. Time series can be of different types with respect to how the events are represented. In many cases the time interval between samples is regular: the temperature at noon every day, or a signal regularly sampled at a rate of 500 Hz. However, time series can also be irregularly sampled and this is the case for instance in astronomy (e.g., Scargle, 1997) and in the analysis of biomedical signals such as heart interbeat intervals (IBI) (e.g., Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology, 1996). IBI series are examples of interval sequences, the analysis of which is the main focus of Publication I.

Given the ubiquity of time series in different disciplines a wealth of domain-specific methods have been developed for their analysis and interpretation, such as biomedical signal processing techniques used to extract features describing the behaviour of the time series. Time series can be represented using different techniques allowing efficient comparison and characterisation using for instance piecewise representations (Faloutsos et al., 1997) or symbolic representations (Lin et al., 2003). Two common topics in the analysis of time series is investigating relationships between time series and investigating the internal temporal structure of a time series. Several techniques have been developed for these purposes. Relationships between time series can be examined using various techniques for matching and clustering (e.g., Chan and Fu, 1999; Gionis et al., 2004 and Rakthanmanon et al., 2012), while the structure of a single time series can be described using for instance autoregressive modelling (e.g., Kalliovirta et al., 2015).

The main focus in this chapter is on investigating the temporal structure of time series, i.e., on gaining insight into the underlying mechanisms that have generated the time series. The structure of a time series is due to the temporal order of the data points and this separates the exploration of time series from types of data without a temporal dimension. The temporal structure of a periodic time series is, for instance, reflected in structures such as the autocorrelation function of the time series and in the Fourier spectrum. Our goal is to find interesting structures in the data that are *non-random*. Non-random structures are here defined as structures that are significantly different from structures found in random instances of the time series and are hence unlikely to occur due to chance. Our goal is to make inference regarding one particular realisation of a time series, e.g., a single IBI series recorded from a subject, rather than studying phenomena that occur on the population level.

Before advanced analysis methods are used to model a time series it is meaningful to first examine whether it actually contains any complex structure, i.e., we should seek the simplest explanation to the observed structure of the time series (Schreiber and Schmitz, 2000). This exploratory analysis includes ruling out that the structure is random. Also, if the structure is explained already by simple patterns there might not be need for more complex analyses. When investigating the structure of a

time series we are hence interested in verifying a particular hypothesis regarding the structure. Investigating structural properties in time series is formally accomplished using a hypothesis testing framework, discussed in this chapter.

As an additional topic in this chapter we also briefly consider the use of the bootstrap (Efron, 1979) for assessing the uncertainty of metrics describing the association between time series. As an example we consider how the bootstrap was used in the sleep study of Publication II.

### 3.1.1 Contributions

Interval sequences are a special type of event sequences, found in many domains, where the events represent the time difference between consecutive events. Such sequences arise naturally for instance in medicine where they can be used to represent, e.g., time intervals between seizures (Andrzejak et al., 2003) or in the analysis of heart interbeat intervals (IBI) considered in Publication I (Henelius et al., 2013).

In Publication I we show how event sequences can be represented as interval sequences in the time, frequency and autocorrelation domains. We present different constrained randomisation schemes for investigating the structure of interval sequences. The constrained randomisation methods are used to identify interesting structural properties in the data and to explain what features in the data that give rise to the observed patterns, e.g., the autocorrelation structure.

In Publication II we apply computational methods in the analysis of data from a sleep study. In particular, we investigate the correlation between time series representing heart rate variability (HRV) and performance on the psychomotor vigilance task (PVT) (Dinges and Powell, 1985) in subjects undergoing sleep restriction and in control subjects. Our results regarding the association between a particular HRV frequency band and PVT agree with the results of Chua et al. (2012). Using our modelling approach we determine to what degree the association between HRV and PVT is explained by their shared variance and the components of a simplified three-process model of alertness.

### 3.1.2 Organisation of this Chapter

This chapter is organised as follows. We first present the basic framework for investigating the structure of time series. The framework is based on empirical significance testing and requires sampling from the distribution of the test statistic under the null hypothesis. We discuss how constrained randomisation schemes can be used efficiently to generate datasets under different null hypotheses. We briefly discuss the use of the bootstrap for estimating uncertainty in the form of confidence intervals. Finally, we conclude the chapter by providing practical data analysis examples.

## 3.2 Framework for Investigating Properties of Time Series

The investigation of the structure of time series follows the general format of empirical statistical hypothesis testing described in Section 2.3, i.e.,

1. Define the null hypothesis regarding the structure of the time series
2. Choose a suitable test statistic describing the property of interest in the time series
3. Compare the value of the test statistic of the original time series to the distribution of the test statistic under the null hypothesis.

The value of the test statistic is calculated from a time series and represents some structural property, e.g., the position of some outliers in the data. The null hypothesis describes the aspect of the data that we want to examine. We might, for instance, investigate if some outliers are the cause for the observed autocorrelation function. The test statistic calculated from the observed time series is compared to the distribution of the test statistic under the null hypothesis, allowing the significance of the observed property to be determined in the form of a $p$-value. The distribution of the test statistic under the null hypothesis can be derived analytically only in some cases, permitting direct sampling (Theiler et al., 1992). The main question in the empirical significance testing framework is therefore *how to sample from the distribution of the test statistic under the null hypothesis*.

When direct sampling from the null distribution is not possible, it is necessary to use simulation methods. In this case an ensemble of synthetic time series are constructed such that the value of the test statistic calculated from these follows the distribution under the null hypothesis.

There are two main approaches for generating synthetic data for empirical hypothesis testing (Schreiber and Schmitz, 2000; Vuokko and Kaski, 2011):

1. The typical realisations approach (model-based)
2. The constrained realisations approach (model-free)

**Model-based sampling** In the model-based approach the observed data is described by a parametric model where the model parameters are estimated from the data. These models are then used to create synthetic time series. However, it is very difficult or even impossible to model complex relationships using this method and hence the model-based approach is limited to the investigation of simple hypotheses (Vuokko and Kaski, 2011).

**Model-free sampling** In the model-free approach synthetic time series are generated using simulation methods such that the structural properties described by the null hypothesis are imposed on the generated time series using different *constraints* (Schreiber and Schmitz, 2000; Theiler and Prichard, 1996). The synthetic time series hence resemble the observed time series in terms of these properties but are otherwise random (Theiler et al., 1992). This widely used framework is referred to as the method of *surrogate data* in the field of nonlinear time series analysis (Schreiber and Schmitz, 2000; Theiler et al., 1992). The model-free method is hence a flexible technique for generating surrogate data. This approach is, naturally, not totally free of assumptions regarding the data. The assumptions are encoded in the form of *constraints* used in the generation of surrogate data and these can be interpreted as a form of data model.

The focus in the rest of this chapter is on model-free methods for generating surrogate data for hypothesis testing.

### 3.2.1 Generating Synthetic Time Series

In this section we provide an overview of generating synthetic time series using the model-free approach before going into more detail. Here surrogate data is generated from an observed time series by randomisation or resampling schemes, thus avoiding the fitting of models to the data.

Permutation and resampling methods have been used extensively in statistics for hypothesis testing (e.g., Good, 2005, 2006; Westfall and Young,

1993) and for calculating confidence intervals (see Efron and Tibshirani, 1994). Refer to, e.g., Schreiber (1998); Theiler and Prichard (1996) and Schreiber and Schmitz (2000) for how constrained randomisations are used to generate surrogate data for time series hypothesis testing.

Different null hypotheses are described in terms of different constraints on the randomisation process. These constraints represent our assumptions and knowledge regarding the time series. As an example of a constraint, consider generating surrogate data such that a subsequence from $i = 10$ to $i = 30$ of a time series $S_t = (x_1, \ldots, x_n)$ is kept fixed. As a second example, consider generating surrogate data where the first 10 elements of the autocorrelation sequence have approximately the same values as in an observed time series. In the first case the constraint is *crisp* and is met exactly or not at all. In the second case the constraint is *soft*. Soft constraints are used when the properties they describe can be met only approximately using the randomisation scheme.

The constrained methods for generating surrogate data can hence be further divided into two categories based on how exactly the constraints are met:

1. methods using soft constraints

2. methods using crisp constraints

In some cases it might be difficult to devise suitable constraints for generating synthetic data where some property is similar to the observed data. Constraints can then also be modelled by a distance function of the form $d\left(\hat{S}, S\right)$ describing the distance between the original sequence $S$ and a randomised sequence $\hat{S}$. We here call the type of constraints described by a distance function *arbitrary constraints*, since distance functions can be used to constrain any aspect of the data being investigated and these hence form the most general method for generation of surrogate data. A distribution over datasets close to the original dataset can then be defined as $\pi\left(\hat{S}\right) \propto e^{-d(S,\hat{S})}$, i.e., the samples from $\pi\left(\hat{S}\right)$ are close to having the desired property, but usually do not exactly meet the constraints. Generating surrogate data with arbitrary constraints is realised using optimisation methods such as simulated annealing (e.g., Schreiber, 1998) or by Markov Chain Monte Carlo (MCMC) methods. Randomising with

soft constraints are discussed in the next section. In this chapter we use MCMC methods for handling arbitrary constraints.

However, although the technique of using distance functions allows any constraint to be used in the sampling process, it is computationally complex and therefore slow in practice. For this reason it is important to consider methods which can be used to efficiently generate surrogate data, although these methods are not generic and are limited to investigating particular null hypotheses.

The randomisation of time series can also be performed in a different domain than the time domain, depending on the null hypothesis and the features it involves. For instance, permuting the phases of the Fourier representation of a time series in the frequency domain can be used to generate time series surrogates having identical Fourier spectra. The non-MCMC methods for generating synthetic time series can hence be divided into two main categories depending on in which domain they operate: (i) methods operating in the time domain and (ii) methods operating in the frequency (Fourier and Wavelet) domain.

Finally, the generation of surrogate data can be divided into methods for (i) univariate data and (ii) multivariate data. Before we proceed to discuss methods for generation of synthetic time series using randomisation methods we will provide an example summarising the above concepts of hypothesis testing of properties of time series.

**Example** Assume that we have observed a time series with an interesting autocorrelation function. We now want to investigate if the autocorrelation is due to some actual temporal process and not due to chance. We hence formulate a null hypothesis stating that the observed autocorrelation is independent of the ordering of the events in the time series. If we are able to reject this null hypothesis we can conclude that the observed autocorrelation is likely non-random. The test statistic describing the structural property of interest is the autocorrelation function. We draw samples from the null distribution by randomly permuting the events in the time series. The value of the test statistic from the original time series is compared to the distribution of the test statistic obtained from the samples from the null distribution, allowing us to calculate an empirical $p$-value. If $p \leq \alpha$ we conclude at a significance level $\alpha$ that the observed

autocorrelation structure is unlikely to have arisen at random.

Furthermore, we suspect that a few outlying data points in the time series give rise to some other interesting structural property. The surrogates can now be generated again by randomly permuting the events in the time series, but this time keeping the outliers fixed, i.e., the randomisation is constrained such that the temporal effect of the outliers is taken into account by keeping their positions fixed. If we in this case fail to reject the null hypothesis we can conclude that we have located the source of the observed pattern and hence explained the structure of the time series.

Different randomisation schemes can hence be used to generate simulated data for hypothesis testing and thereby gain insight into the structure of the time series.

We next discuss different methods for generating surrogate time series with an emphasis on univariate methods using constrained randomisations. We begin by discussing how to generate surrogate data with arbitrary constraints, after which we proceed to discuss non-MCMC methods for surrogate data generation in the time and frequency domain.

### 3.3 Randomising with Soft Constraints

When arbitrary constraints are used the data distribution is difficult, or even impossible, to derive analytically (Theiler et al., 1992) and hence methods such as Markov Chain Monte Carlo (MCMC) sampling (see Section 2.7) are used.

Arbitrary constraints are modelled using a non-negative distance function (cost function) of the form

$$d\left(\hat{S}, S\right), \tag{3.1}$$

describing the distance between $S$ (the original sequence) and $\hat{S}$ (a randomised version of the original sequence $S$). For the original sequence we have $d(S, S) = 0$, i.e., the global minimum of the distance function is reached when the constraints are exactly met.

As an example of a distance function we can consider generating surrogate time series that share the same first ten elements of the autocorrelations sequence with the original sequence. In this case we can use the

function

$$d_r\left(\hat{S}, S\right) = \sum_{i=1}^{10} \mid r_i' - r_i \mid, \tag{3.2}$$

where $r'$ and $r$ denote the values of the autocorrelation function at lag $i$ for the surrogate and original time series, respectively. The distribution of datasets from which we want to sample has the form

$$\pi\left(\hat{S}\right) \propto e^{-d(S,\hat{S})\beta}, \tag{3.3}$$

i.e., samples $\hat{S}$ from this distribution are likely to be close to the original sequence $S$ in terms of the distance function $d$. The parameter $\beta$ determines how strictly the constraints must be satisfied in the sampling process. Sampling can be accomplished, e.g., using MCMC methods such as the parallel tempering technique described in Section 2.8. The drawback of MCMC-based techniques is that they are slow in practice, depending on the required sampling scheme. The methods discussed next are more efficient but only allow the testing of specific null hypotheses.

## 3.4 Randomising in the Time Domain

In this section we consider the generation of surrogate data in the time domain using methods based on random permutation. We also briefly consider bootstrapping of time series.

### 3.4.1 Random Permutation Techniques

The null hypothesis that the samples in a time series are independent can be tested by randomly permuting all elements in the sequence at random (Schreiber and Schmitz, 2000; Theiler et al., 1992). This method is used in Publication I (Henelius et al., 2013) in the analysis of interval sequences and an example is presented below in Section 3.8.

Random permutation of the samples breaks all temporal structure in the time series. However, the amplitude distribution and all statistics that do not depend on the order of the samples (e.g., mean and variance) are unchanged. As an example, consider the original sequence shown in Figure 3.1a that appears to contain six segments, delimited by vertical lines. An instance obtained by randomly random shuffling from the original time series is shown in Figure 3.1b.

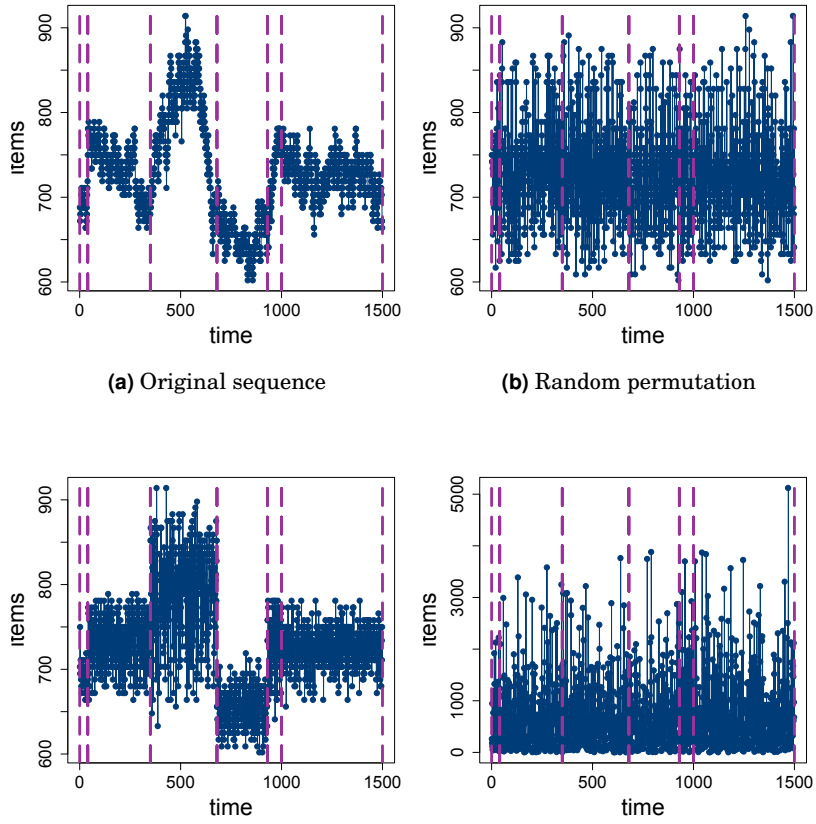
The random permutation scheme can also be expressed as a distance function (Equation (3.1)), to exemplify the relation between different ran-

domisation schemes and the above discussed sampling using arbitrary constraints. In the case of random permutation, every permutation of the samples in the original time series is equally likely and consequently the probability distribution over the datasets (Equation (3.3)) is flat and the distance function is constant, i.e.,

$$d\left(\hat{S}, S\right) = 0. \tag{3.4}$$

A variation of the full random permutation of events is obtained by fixing a subsequence of samples in the time series. This method has been used in Publication I (Henelius et al., 2013) for randomising interval sequences: a set of intervals in the original time series is kept in a fixed position while all other intervals are randomly permuted. Statistics such as the mean and variance remain unaffected here as well. This randomisation scheme allows hypothesis testing concerning certain intervals, for instance whether the observed structure of the time series is explained by some outliers. This is exemplified in Figure 1.1. Another example is given in Figure 3.1c where the original sequence has been permuted using a constrained permutation scheme in which each of the six segments are permuted individually. This scheme preserves the overall temporal structure of the time series and also, e.g., the mean of each segment, and the surrogates can be used to investigate hypotheses related to the time-dependent structure of the sequence.

A further example of a randomisation method applied in Publication I to interval sequences is a uniform sampling of interval. Here samples are drawn from a distribution of interval sequences, such that the number of events and the total duration of the synthetic time series agree with the original time series. For an interval sequence the total duration of the time series equals the sum of the intervals. This particular randomisation technique is only applicable to interval sequences. An example is shown in Figure 3.1d. Note that the randomised samples in this scheme are not obtained by permuting the original time series, instead the properties of the original time series (number of intervals and their sum) are used as constraints when creating surrogate data. Using this distribution it is possible to examine the null hypothesis that the intervals in the event series are uniformly distributed.




**(a)** Original sequence

**(b)** Random permutation

**(c)** Constrained permutation

**(d)** Sample from uniform distribution

**Figure 3.1:** Different randomisation schemes applied to an interval sequence. The boundaries of the segments in the sequence are marked with vertical dashed lines. In the constrained permutation the data within the segments are randomised individually, thus preserving the temporal structure of the sequence.

### 3.4.2 Bootstrapping Techniques

Bootstrap techniques can be applied to time series in order to estimate the sampling distribution of some statistics (Carlstein, 1990), for instance to estimate the standard error of the mean of a time series (e.g., Politis, 2003).

For reviews of various bootstrap methods for time series refer to e.g., Bühlmann (2002); Carlstein (1990); Kreiss and Lahiri (2012) and Politis (2003). Bootstrap schemes for time series differ from the traditional i.i.d. bootstrap due to the dependencies between samples in time series. Examples of non-parametric time series bootstrap methods are, e.g., the

blockwise jackknife and bootstrap methods.

Assume that a time series with $n$ samples has been observed. In the blockwise jackknife method (Kunsch, 1989) a surrogate time series is obtained by deleting a randomly chosen block of $l$ samples from the original data. In the blockwise bootstrap (Kunsch, 1989), the time series is divided into $n - l + 1$ blocks of length $l$ after which $n/l$ blocks are sampled with replacement to yield the surrogate time series. The circular block bootstrap (Politis and Romano, 1991) extends the block bootstrap by a circular extension of the time series, which overcomes the bias caused by samples in the beginning and end of the time series having less weight in the (non-circular) block bootstrap.

## 3.5 Randomising in the Frequency Domain

In some cases it is convenient to perform the randomisation of the time series in the frequency domain, where certain properties can be easily modelled. An example is the generation of surrogate data having the same Fourier spectrum as the original data.

### 3.5.1 Fourier domain

**Phase Randomisation** Phase-randomised surrogates allow the testing of the null hypothesis that the time series is autocorrelated Gaussian noise where the structure of the time series is contained in its Fourier coefficients in the frequency domain (Theiler et al., 1992). The effect of the phase randomisation scheme is to remove nonlinear relationships in the data while preserving linear relationships (Breakspear et al., 2003). We first briefly describe the basic phase randomisation technique (see Theiler et al. (1992)) after which we present some extensions to this method.

Let $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively. A time series $S = (s_0, s_1, \ldots, s_{n-1})$ can be represented using the discrete Fourier transform as

$$\mathcal{F}(S) = Z = (z_0, z_1, \ldots, z_{n-1}) \text{ where } z_j = \sum_{k=0}^{n-1} s_k e^{-2\pi ijk/n}. \tag{3.5}$$

Here $z_j = c_j e^{i\varphi_j}$, where the coefficients $c_j$ are the Fourier amplitudes and $\varphi_j$ are the Fourier phases. The inverse Fourier transform of $Z$ is given by

$$\mathcal{F}^{-1}(Z) = S = (s_0, s_1, \ldots s_{n-1}) \text{ where } s_j = \frac{1}{n} \sum_{k=0}^{n-1} z_k e^{2\pi ijk/n}. \tag{3.6}$$

A surrogate time series is generated by adding random phase shifts $\varphi_j^* \in [0, 2\pi)$, $j = 0, \ldots n-1$ to the Fourier phases, which corresponds to multiplying each $z_j$ in Equation (3.5) by $e^{i\varphi_j^*}$. The samples of the surrogate time series $S^*$ are then given by

$$s_j^* = \frac{1}{n} \sum_{k=0}^{n-1} z_k e^{i\left(2\pi jk/n + \varphi_j^*\right)}. \tag{3.7}$$

Since the Fourier amplitudes are not affected in this randomisation scheme it follows that the surrogates have the same power spectrum and, from the Wiener–Khinchin theorem, the surrogates also have the same periodic (circular) auto-covariance function (Prichard and Theiler, 1994; Schreiber and Schmitz, 2000; Theiler et al., 1992).

An example of phase randomisation is given in Figure 3.2a. The surrogate time series has here been obtained by randomly permuting the Fourier phases, after which the time series has been reconstructed.

Some shortcomings associated with the above described scheme of phase randomisation have been noted. Since it is the periodic autocorrelation function that is preserved and not the standard autocorrelation function (Prichard and Theiler, 1994), it is possible that spurious high-frequency noise is introduced in the surrogate time series if the values at the beginning and end of the time series have markedly different values preventing a smooth circular extension of the time series (Theiler et al., 1992). The problem with spurious high-frequency components present in the surrogate data can be solved, e.g., by choosing a subsequence from indices $A$ to $B$ from the original time series $S$ such that the values of $s_A \approx s_B$, i.e., the values at the end points are approximately similar (Theiler et al., 1992). Alternatively a windowing function that sets the time series to zero at the endpoints can be applied before the time series is transformed to the Fourier domain (Theiler et al., 1992).

**Amplitude Adjusted Fourier Transform** The amplitude adjusted Fourier transform (AAFT) method (Theiler et al., 1992) can be used when the observed data $S$ is not Gaussian, but is assumed to be a nonlinear transformation $h$ of a linear Gaussian time series $Y$, i.e., $S = h(Y)$ (Theiler et al., 1992). The surrogate time series has the same amplitude distribution as the original time series $S$ and the linear correlations in the underlying time series $Y$ are also preserved (Theiler et al., 1992). The AAFT method

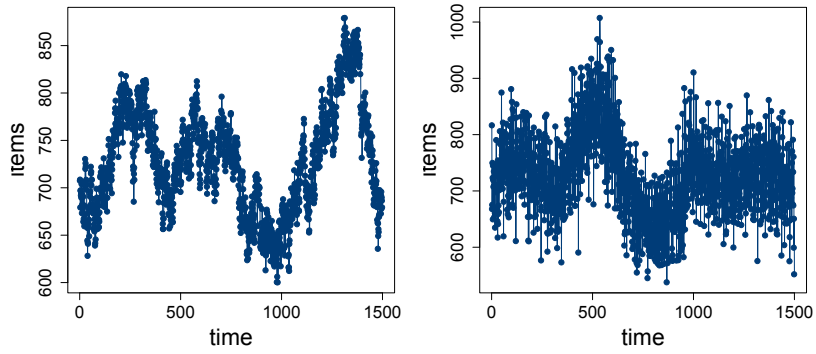
proceeds as follows. Let $S$ be the original time series. A time series $S'$ is created by sampling from a Gaussian distribution and the samples in $S'$ are ordered to have the same rank-order as the samples in $S$. Phase randomisation is applied to $S'$ in the Fourier space and the sequence is transformed back into the time domain. Finally, $S$ is rank-ordered according to the values in $S'$ to yield the surrogate time series $S^*$.

The iterated AAFT (IAAFT) algorithm (Schreiber and Schmitz, 1996) has been proposed to solve the problem that the surrogates from the AAFT method have a flatter spectrum than the original time series. In the IAAFT method, the surrogates are constructed in a two-step iterative process where in each step both the spectrum (in the frequency domain) and the amplitudes (in the time domain) are modified as to approach their counterparts in the original time series $S$.

**Constrained Fourier Randomisation** A variation of a Fourier-based resampling technique is given in Henelius et al. (2013), in which it is applied in the analysis of interval sequences. This constrained randomisation scheme proceeds as follows. Randomisations are applied in the Fourier space such that a set of Fourier amplitudes and phases are kept fixed in the original sequence while all other amplitudes and phases are replaced with samples from a Fourier-transformed realisation obtained from a copy of the original sequence randomly permuted in the time domain. Finally, the randomised sequence is transformed back into the time domain using the inverse Fourier transform. An example of surrogate data generated using this constrained randomisation scheme is shown in Figure 3.2b where the first five Fourier amplitudes have been kept fixed. Since the low-order Fourier coefficients correspond to low frequencies in the data, the randomisation in this case is comparable to a form of low-pass filtering of the sequence, i.e., the coarse structure of the time series is preserved while the fine details are changed.

### 3.5.2 Wavelet domain

A signal is represented in the wavelet domain (e.g., Mallat, 2009) by a multilevel decomposition, such that the different levels successively provide more detail regarding the signal. At each level, the signal is represented both by coarse *approximation* coefficients (comparable to low-




**(a)** Phase randomisation

**(b)** Constrained randomisation of Fourier amplitudes

**Figure 3.2:** Different randomisation schemes applied in the frequency domain to the original interval sequence in Figure 3.1a. In phase randomisation the phases are randomly permuted. In the constrained randomisation of amplitudes the first five amplitudes are kept fixed.

frequency phenomena in the data) and *detail* coefficients (comparable to high-frequency phenomena).

Generation of surrogate data is here in essence accomplished by random permutation of the detail coefficients of the wavelet decomposition of a time series, after which the time series is again reconstructed in the time domain (e.g., Breakspear et al., 2003; Bullmore et al., 2001 and Vuokko and Kaski, 2011). The IAAFT-method has also been used for constrained randomisation of wavelet coefficients (Keylock, 2006, 2007).

## 3.6 Considerations Regarding Surrogate Data

No general rule for how to choose a suitable null hypothesis for a given time series can be given. One should try to find the simplest explanation for the data being analysed (Schreiber and Schmitz, 2000).

The general approach is to start with simple explanations and to proceed to more extensive explanations, i.e., one should apply Occam's razor as noted by Schreiber and Schmitz (2000). In terms of constrained randomisations, this means that constraints are iteratively added until the structural properties of the data have been explained, thus gradually building up an understanding of the data, see, e.g., Hanhijärvi et al. (2009a).

Vuokko and Kaski (2011) note that a typical problem with surrogate

data is that the null distribution does not accurately reflect the structure of the observed time series which leads to Type I errors. E.g., the problems in the AAFT method regarding the slight deviation in the surrogates from the true spectrum may lead to problems in significance testing (Kugiumtzis, 1999; Schreiber and Schmitz, 1996).

**Multivariate Surrogate Data** Although not discussed here, the use of multivariate data has applications in several domains, e.g., studying the relationship between biomedical time series. Some of the above presented methods have readily been extended also to the multivariate case. For instance, Prichard and Theiler (1994) presented a method for constructing multivariate time series surrogates such that both linear correlations in each time series is preserved as well as linear correlations between the different time series. They show that this can be achieved by adding the same set of randomised phases to the Fourier transform of each individual time series in the multivariate ensemble. Also note the multivariate methods presented by, e.g., Breakspear et al. (2003); Jentsch and Kreiss (2010), and Vuokko and Kaski (2011).

## 3.7 Estimating Uncertainty using the Bootstrap

Biomedical data is often characterised by high inter- and intrasubject variability and often limited sample size. The interactions in the data can also be complex and must be taken into account when investigating cause-effect relationships. Due to these factors it is important to quantify the degree of variability in the data. We here consider the use of the bootstrap for assessing the uncertainty of the association between time series.

In medical research it can be of interest to investigate differences over time, e.g., between a study group receiving treatment and a control group without treatment. Such relationships can be captured for instance by considering the correlation between treatment and response in the two groups. The correlation provides a single measure of the relationship between the time series. The nonparametric bootstrap can then be used to determine confidence intervals for the correlation. In the next section we consider how the bootstrap was used to estimate confidence intervals in the sleep study of Publication II.

## 3.8 Examples of Time Series Randomisation

Different types of randomisation schemes have been applied in various settings. For instance, in the field of medicine the dynamics of heart rate variability have been investigated by Fourier-based randomisation (Garde et al., 2001; Li et al., 2010; Porta et al., 2007) and using time-varying surrogates (Faes et al., 2009). Constrained randomisation has also been applied in the study of epileptic seizures by considering time intervals between seizures (Andrzejak et al., 2003). Fourier and wavelet-based randomisation schemes applied in the analysis of MRI data have also been compared (Laird et al., 2004).

We will next provide some examples on how the randomisation methods and the bootstrap can be used for analysis of time series. The examples are based on Publications I and II.

### 3.8.1 Interval Sequences

**Dataset** In this section we consider the `chf210` record in the *Congestive Heart Failure RR Interval Database* (Goldsmith, 2003) from the PhysioBank signal archive (Goldberger et al., 2000). The `chf210` record contains heart interbeat intervals (IBIs) from a subject with congestive heart failure.

**Interval Sequences** We consider the structural properties of interval sequences. Given an interval sequence $S = (s_1, s_2, \ldots s_n)$ it is convenient to use its log-transformed version $S_z = (z_1, z_2, \ldots, z_n)$, where $z_i = \log s_i$. The logarithm transformation means that multiplicative scaling of the intervals is transformed into an additive scaling, i.e., the effect of scaling by a factor $K$ or $1/K$ corresponds to a change of $\log K$ in the intervals. Furthermore, although the intervals in the time series $S$ are positive, the intervals in $S_z$ can also take negative values which is important for, e.g., the randomisation schemes operating in the Fourier domain.

**Significance Testing** We here consider the structure of the interval series in terms of the Fourier coefficients, as represented in Equation (3.5).

The analysis regarding the structural properties of time series in this section is exploratory and based on the use of confidence intervals. This analysis should therefore be considered as a means of identifying structural

properties instead of a formal significance test. The following method is used in the analysis.

We estimate 95% quantile-based confidence intervals for the Fourier coefficients based on an ensemble of $R = 1000$ surrogate interval time series generated according to the different null hypotheses we are investigating. Using 1000 samples ensures that the standard deviation of the upper and lower bounds for the confidence intervals, corresponding to the 0.025 and 0.975 quantiles, is approximately 0.5%. To reduce variance we average the values of the features in bins of width 10.

We estimate the validity of a null hypothesis in terms of how well the Fourier coefficients calculated from the observed time series are contained within the confidence intervals, i.e., the agreement is assessed in terms of the number of Fourier coefficients inside the confidence intervals. Fourier coefficients outside the confidence interval are termed significant. In the exploratory analysis performed here we can choose to reject the null hypothesis if, for instance, more than 25% of the features are significant.

It should be noted that the comparison based on the confidence intervals would require multiplicity correction using, e.g., the conservative Bonferroni method, which would make the confidence bands wider. In the following analysis we, however, omit multiplicity correction. For principled methods of estimating confidence intervals for time series refer to Korpela et al. (2014).

**Null Distributions** We here consider the first 1500 samples of the `chf210` sequence, shown in Figure 3.3a. Outliers in the data were detected using an automatic method for artifact detection of IBI signals (Xu et al., 2001) and are marked with rectangles. The appearance of the data is investigated under different null distributions. Application of the uniform sampling scheme applied in Publication I and discussed above in Section 3.4.1 is shown in Figure 3.3b. Random shuffling without constraints is shown in Figure 3.3c which breaks the temporal structure of the data. In Figure 3.3d the data is randomly shuffled keeping the outliers fixed. In Figure 3.3e the data is randomly shuffled such that the values of the autocorrelation sequence at lags 1 to 10 is approximately preserved. Note that the figures show the original data and not the logarithm-transformed data.

**Significance of Fourier Coefficients** We consider the significance of the Fourier coefficients in terms of the above described four null distributions. The results are shown in Figure 3.4.

(A) Uniform Distribution We first investigate the null hypothesis that the data follows a uniform distribution.

From Figure 3.4a it is clear that all Fourier coefficients for randomisations using the uniform distribution are outside the confidence intervals and we reject the null hypothesis that the data is uniformly distributed.

(B) Temporal Dependence Next we investigate the null hypothesis that there is no temporal dependence among the elements in the interval sequence. We use random shuffling of the intervals to break the temporal structure in the interval series. The test statistic is the number of significant Fourier coefficients.

Figure 3.4b shows that about half of the Fourier coefficients are significant, meaning that the data is not completely explained by this null distribution. We reject the null hypothesis and conclude that there is temporal dependence among the intervals in the sequence.

(C) Outliers We now form a null hypothesis that observed structure of the Fourier coefficients is explained by the outliers in the data. We use the same test statistic as above and as a null distribution we use a random shuffling where the outliers in the data are kept fixed.

The confidence intervals calculated using random shuffling with fixed outliers are shown in Figure 3.4c. We observe that there are fewer significant Fourier coefficients than in the previous models and in this exploratory investigation we therefore do not reject the null hypothesis and conclude that the outliers explain some of the structure in the data reflected in the Fourier coefficients.

(D) Autocorrelation We continue the investigation of the data by testing the null hypothesis that the structure of the Fourier coefficients is explained by the values at lags 1 to 10 of the autocorrelation function of the sequence. We use the same test statistic as above. The null distribution consists of datasets sampled using the parallel tempering MCMC method from a distribution where the values of the autocorrelation function at lags 1 to 10 are fixed with respect to the original sequence, i.e., we sample using a distance function of the form given by Equation (3.2).

Based on Figure 3.4d we conclude that only a few Fourier coefficients

are significant and we do not reject the null hypothesis, concluding that the autocorrelations at lags 1 to 10 explains some of the structure in the Fourier coefficients.

**Summary** In this section we showed how different null distributions can be used to investigate different aspects of the data. The null distribution must be properly chosen, e.g., here the uniform distribution is not a suitable representation of the data. Without constraints on the randomisation, i.e., using random shuffling, we determined that there is some temporal structure in the data. By constraining the randomisation to take the outliers into account more of the structure of the data is explained. Similarly, the values of autocorrelation function at lags 1 to 10 also explain some of the structure of the data. We conclude that the structure of the time series can be explained both by the outliers and by the autocorrelation structure.

### 3.8.2 Sleep Study

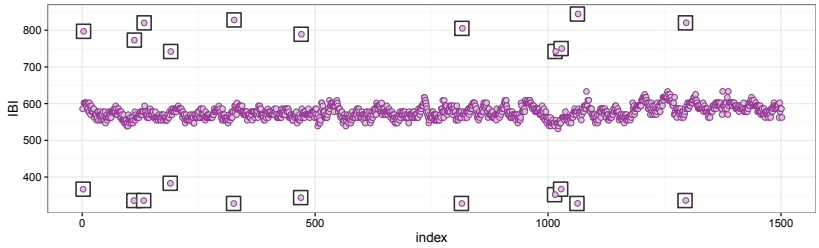
**Dataset** In this section we consider the sleep study of Publication II (Henelius et al., 2014b). The goal in Publication II was to computationally examine the association between spectral HRV metrics and neurobehavioural metrics of vigilant attention represented by the psychomotor vigilance task (PVT) (Dinges and Powell, 1985). The subjects in the study were assigned into either a sleep restriction group, allowed only four hours of nightly sleep for five consecutive nights, or into a control group in which subjects were allowed eight hours of nightly sleep. The PVT was recorded, together with an electrocardiogram used for HRV analyses, four times daily in the sleep restriction group and three times daily in the control group. In Publication II it was of interest to investigate Pearson's correlation between HRV and PVT for subjects in the sleep restriction group and control group, in order to investigate the correlation between HRV power in the 0.02–0.08 Hz frequency band and lapses on the PVT, see Chua et al. (2012). In Publication II the correlation between the HRV and PVT time series was determined separately for each frequency band from 0.01–0.40 Hz in steps of 0.01 Hz. 95% confidence intervals (using the $BC_a$ method, see Section 2.5) for the correlation were estimated by bootstrapping the correlation coefficients, using 5000 samples, on the subject level, separately for each group. Using 5000 samples the standard deviation of the 0.025 and 0.975 quantiles (raw, not $BC_a$) is less than 0.25%.

We use these confidence bands to examine whether sleep restriction affects the correlation between HRV and PVT. The results for control group are shown in Figure 3.5a and in Figure 3.5b for the sleep restriction group. The figures correspond to Figure 3 in Publication II and show the average correlation between HRV and PVT together with bootstrap confidence intervals. Zero is contained in the confidence interval for the control group for all frequency bands. This means, that we cannot rule out the possibility, at the 95% confidence level, that the correlation between HRV and PVT is zero in the control group. In the sleep restriction group, on the other hand, it is clear that the correlation in the region from about 0.01 Hz to approximately 0.09 Hz is larger than zero, indicating that this region is interesting, which agrees with the results of Chua et al. (2012).

**Summary** In this section we examined the association between a physiological signal and a time series representing neurobehavioural attention. The use of bootstrap confidence intervals make it possible to estimate the variance associated with different subjects, i.e., the effect of sampling a different set of subjects. The use of confidence intervals also allows for visual hypothesis testing, making the interesting patterns, i.e., where zero is not included in the interval, easy to detect.

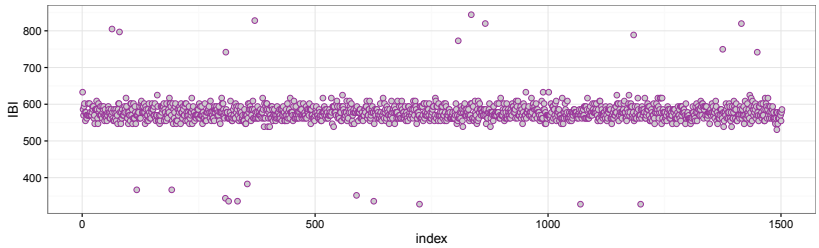
## 3.9 Chapter Summary

In this chapter we discussed the topic of investigating the structure of time series using a hypothesis testing framework based on constrained randomisation and resampling and we also briefly considered bootstrapping for robust estimation. Different null hypotheses correspond to different constrained distributions from which surrogate time series are sampled. Arbitrary constraints require sampling methods such as MCMC. In some specific cases the distributions can also be modelled by direct application of efficient randomisation techniques in, for instance, the time and frequency domains, in which case possibly slow MCMC sampling is not required. The randomisation-based methods discussed in this chapter form a viable framework for exploring the structure of time series. The goal is to gain insight into the time series and determine if any non-random structures are present, after which more advanced and specialised analysis techniques can be employed.

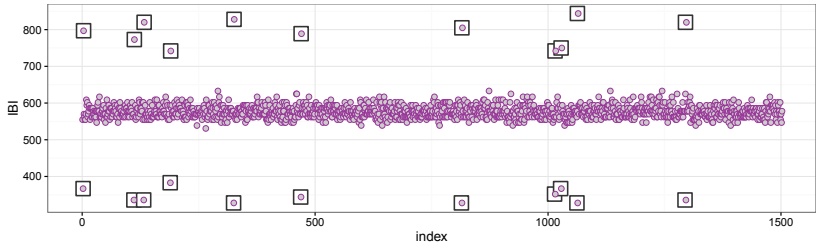



**(a)** Original sequence

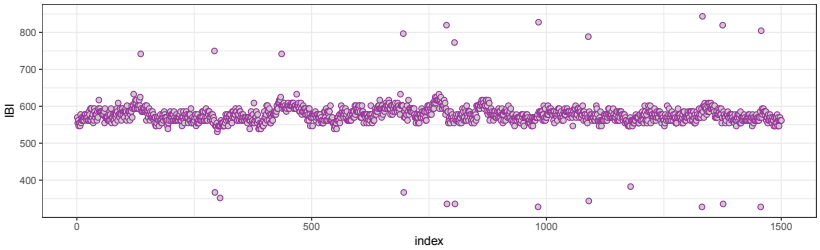

**(b)** Uniform distribution




**(c)** Random shuffling

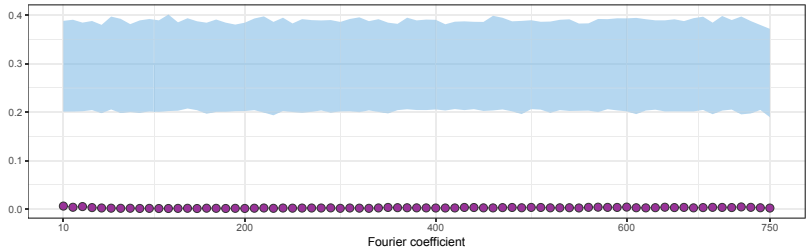



**(d)** Random shuffling with fixed outliers

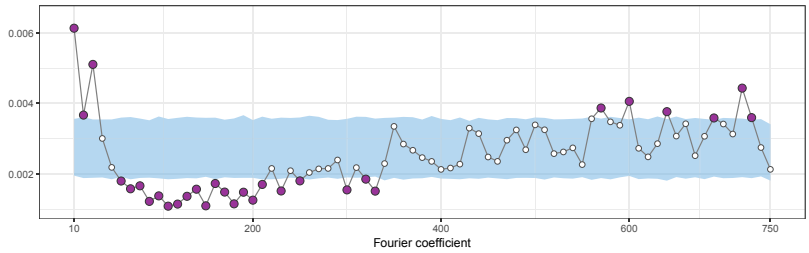



**(e)** Random shuffling preserving the autocorrelation function at lags 1 to 10

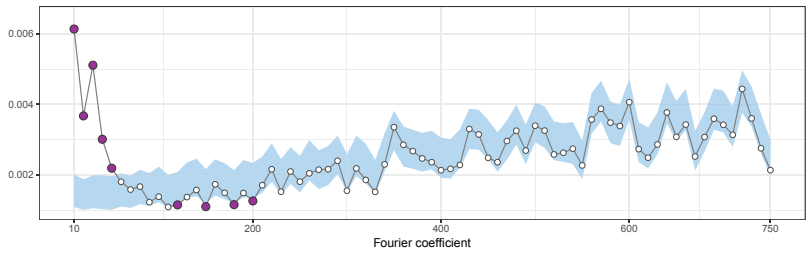
**Figure 3.3:** Randomisations of the `chf210` sequence.



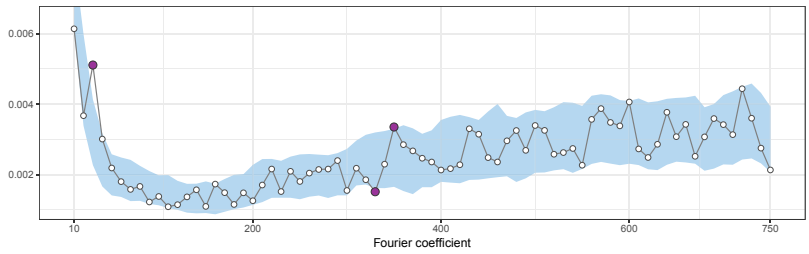

**(a)** Uniform distribution




**(b)** Random shuffling

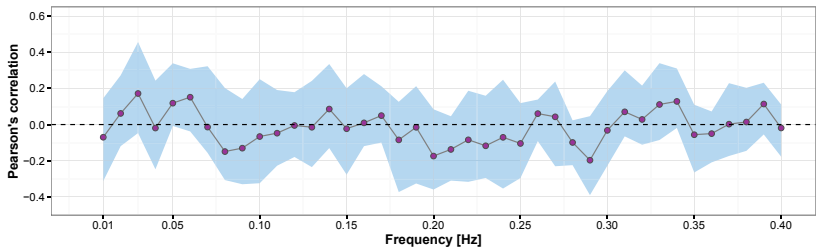



**(c)** Random shuffling with fixed outliers

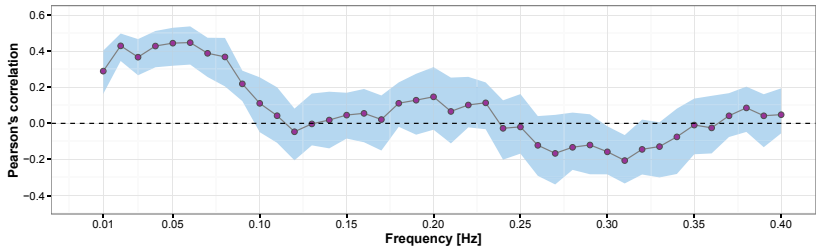



**(d)** Fixed autocorrelation coefficients

**Figure 3.4:** 95% confidence intervals, shown in blue, for the Fourier coefficients of the `chf210` sequence under different null distributions.




**(a)** Control group




**(b)** Sleep restriction group

**Figure 3.5:** Average correlation with 95% bootstrap confidence intervals, shown in blue, for the correlation between HRV and PVT for the sleep restriction and control groups.

# 4. Exploring Attribute Interactions Utilised by Classifiers

In this chapter we explore two topics concerning attribute interactions utilised by classifiers.

Firstly, we study the problem of understanding how classifiers exploit attribute interactions in making predictions. This is an important problem, since many classifiers are essentially opaque and it is difficult to understand how the classifier uses the structure of the data. We develop a novel randomisation-based technique to solve this problem using which we gain insight into how the classification algorithm utilises the structure of the data.

Secondly, we consider the problem of investigating the structure of a dataset in terms of finding groups of attributes that interact with respect to some specific variable in the dataset, such as a class label. We show that this problem corresponds to finding a factorisation of the class-conditional joint distribution which is internally modelled by a classifier. Knowledge of attribute interactions in datasets is useful in several data analysis tasks, but finding these interactions is generally not easy. We develop a novel method, based on a statistical hypothesis testing methodology, for automatically finding a factorisation of the class-conditional joint data distribution.

In essence, in both of the topics considered in this chapter, we want to find a disjoint partition of the attributes of a dataset in terms of some goodness metric. In this chapter we develop a framework that allows us to study the above mentioned attribute interactions and we present methods that efficiently implement the framework.

## 4.1 Introduction

We first consider the topic of exploring interactions utilised by classifiers, after which we discuss how to examine the interaction structure in a dataset.

### 4.1.1 Exploring Classifiers

Supervised learning, such as classification, is a fundamental topic in machine learning. Consequently, a wide variety of algorithms have been developed to efficiently solve classification problems. A lot of focus has been on algorithms that achieve high predictive power, e.g., state-of-the-art high-performing classifiers such as support vector machines (SVM) (Cortes and Vapnik, 1995), random forests (Breiman, 2001), C5.0 decision trees (Quinlan, 1993) or extreme learning machines (Huang et al., 2012), which have been found to be among the best-performing algorithms (Fernández-Delgado et al., 2014). However, although these algorithms reach high predictive performance a drawback is that they in practice are opaque, i.e., they are more or less *black boxes*. With black boxes we here mean that it is difficult to understand what the basis for the predictions made by the classification algorithms are.

As an example, consider a medical researcher analysing a dataset containing a large number of biomedical markers with the goal of developing a model for predicting whether or not a patient runs the risk of contracting a certain disease. Such a model must clearly have high predictive performance, but to be useful the model must also be *interpretable*, i.e., we must on some level be able to understand how the results are reached by the classifier. As another example of the importance of being able to interpret machine learning models, the new General Data Protection Regulation by the European Union requires that individuals are entitled to an explanation for algorithmic decisions (Goodman and Flaxman, 2016). Furthermore, Raccuglia et al. (2016) used classifiers to predict the outcome of chemical reactions and by determining the features used by the classifier they could explain which factors are important for the chemical reactions.

The problem we are faced with in predictive modelling using classifiers is therefore twofold: (i) we want to use models with high predictive performance and (ii) we want to be able to understand the basis for the predictions. Here we have several possibilities. One approach is to use

classifiers that are readily interpretable such as linear models (e.g., logistic regression), decision trees or classification rules. However, based on the results of Fernández-Delgado et al. (2014) such more interpretable models are outperformed by more complex, opaque classifiers as, for instance, SVM or random forest.

A further complication arises from the fact that typical datasets today are high-dimensional, in which case even simple models such as logistic regression or decision trees may end up including a too large number of attributes for the models to be interpretable. This also implies that an interpretable model should preferably be sparse to allow for easy interpretation and consequently some methods favouring sparsity have been created, e.g., novel methods based on scoring systems trying to balance accuracy and interpretability (e.g., Ustun et al., 2014).

The concept of model interpretability is complex and consists of several factors, see, e.g., Lipton (2016) for a good discussion of this topic. An interpretable model can be thought of as being characterised by transparency and post-hoc interpretability (Lipton, 2016). Here the former means that it is possible to understand how the algorithm works while the latter means that the decisions of the classifier can be explained, e.g., using text or by examples or visualisations (Lipton, 2016). The method developed in this chapter for investigating how classifiers exploit attribute interactions for making predictions belongs to the former category. Our goal is to make the algorithm more transparent, i.e., we want to *peek into the black box*.

### 4.1.2 Attribute Interactions and Data Structure

Interpretability in terms of understanding relationships in a dataset, as defined above, can also be viewed in terms of gaining insight into the *structure of the dataset*. The concept of structure, however, is ambiguous and depends on the context. In classification problems it is natural to consider the structure of a dataset in terms of *attribute interactions*.

With attribute interactions we here mean that two or more attributes carry complementary information such that they must be jointly considered by the classifier in the prediction task. Consider a data matrix $X$ with $m$ attributes and let $C$ be a vector of class labels, one for each data item in $X$. Our main goal when investigating attribute interactions is to find a disjoint partition of the $m$ attributes of $X$ into $k$ groups, i.e., we want

to find a set of groups $\mathcal{S} = \{S_1, \ldots, S_k\}$ such that attributes in the same group $S_i$ are interacting (dependent) and attributes in different groups are independent. Each attribute only belongs to one group. We call this disjoint partition $\mathcal{S}$ a *grouping* of the attributes of $X$ and this partition describes the attribute interaction structure.

In this chapter we consider two highly related problems concerning attribute interactions:

P1 Finding groups of interacting attributes that are exploited by a classifier in making predictions. This is the approach in Publication III where we present the `GoldenEye` algorithm and which is applied in Publication IV in the form of the `GoldenEye++` algorithm.

P2 Finding groups of attributes in a dataset that interact with respect to the class label. This provides insight into the structure of the dataset in terms of class-dependent interactions and also corresponds to a factorisation of the class-conditional joint distribution of the data. This is the approach in Publication V where we present the ASTRID method.

Below we show that these two problems are both instances of a *generic clustering problem*, where the goal is to find an optimal partition of the attributes of a dataset with respect to a value function, that depends on whether we consider P1 or P2. This problem of grouping the attributes constitutes one part of the framework presented here. The other part is formed by a method for evaluating the significance that a dataset originates from a specific factorised class-conditional joint distribution. This can be used to evaluate the statistical significance of different partitions of the attributes of a dataset in terms of P2.

It should be noted that although the attribute interactions considered in these two problems P1 and P2 are very similar, they are not identical. Problem P1 is concerned with exploring which attribute interactions the classifier is using, whereas problem P2 is about exploring the attribute interactions needed to train a useful classifier.

Our goal when studying attribute interactions are to (i) better interpret the predictions of the classifier (`GoldenEye` and ASTRID) and (ii) to gain insight into the structure of the data (ASTRID).

Knowledge of these two types of attribute interactions is important in several data analysis applications such as, e.g., in feature selection and data anonymisation. We now consider a brief example.

**Example** Consider the analysis of a dataset with a large number of attributes. The dimensionality of the dataset is typically reduced by variable and feature selection. Now, if we know the attributes that are interacting in terms of a grouping of the attributes, it is meaningful to perform variable subset selection based on the interacting groups as this ensures that the relationships in the data remain intact.

In problem P2, considered in Publication V, our goal is to find a suitable factorisation of the joint distribution of the data. We briefly consider this next.

Estimating the full joint distribution of the data is not straightforward and would typically require complex modelling. However, when studying classifiers we can explore the structure of the dataset in terms of the *class-conditional joint distribution* that captures the interactions between the attributes with respect to the class label.

The goal of a classifier is to find a model describing class probabilities given the data, i.e., the classifier considers a model of the form

$$\Pr(C \mid X) = \frac{\Pr(X \mid C)\Pr(C)}{\Pr(X)}, \tag{4.1}$$

where the factor $\Pr(X)$ is constant given the data and hence

$$\Pr(C \mid X) \propto \Pr(X \mid C)\Pr(C). \tag{4.2}$$

Here $\Pr(X \mid C)$ is the class-conditional joint distribution of the attributes. We consider a factorisation of $\Pr(X \mid C)$ into independent factors given by a partitioning $\mathcal{S}$ of the attributes of $X$:

$$\Pr(X \mid C; \mathcal{S}) = \prod_{S \in \mathcal{S}} \Pr(X_S \mid C), \tag{4.3}$$

where $X_{S_i}$ only contains the part of $X$ corresponding to the attributes in the group $S_i$. This factorised class-conditional distribution captures the dependencies between the attributes and thus provides information on the structure of the dataset in terms of class-conditional attribute interactions.

We approach the problem of finding such a factorisation of the class-conditional joint data distribution as a statistical hypothesis testing problem based on the following intuition. Assume that we have two classifiers:

$f_1$ trained using data from an unfactorised distribution, and $f_2$ trained using data from a factorised distribution. If these two classifiers cannot be distinguished from each other in terms of performance, it means that the factorisation correctly captures the essential class-dependent interactions in the data and all relevant relationships between attributes that are needed to train a good classifier are intact. However, if $f_2$ performs worse than $f_1$ it means that some essential interactions in the data needed by the classifier were broken and we conclude that the factorisation in this case was incorrect. In the method we develop a randomisation test is used to determine the significance of the factorisation in the sense described here.

In other words, the method developed here leverages classifiers for examining the structure of attribute interactions in the dataset, which is based on the following observation: in order for classifiers to achieve high predictive performance we can assume that they at least implicitly model and exploit the interactions between attributes in the dataset. Hence, if we are able to observe the attribute interactions needed to train a good classifier it means we can find the interacting attributes.

We continue the above example of the analysis of a biomedical dataset. Knowledge of which attribute interactions are exploited by the classifier allows the researcher to understand how the predictions are made and can also reveal that some irrelevant spurious attribute interactions are being utilised, e.g., due to overlearning or errors in the data. On the other hand, understanding the structure of attribute interactions in the data can be motivated using economic and ethical considerations. If, for instance, it turns out that only a few of the biomedical markers are needed to construct a reliable classifier, it means that the diagnosis can be made quicker and cheaper by leaving out unnecessary markers. This is also an example of how the partition of attributes into interacting groups can be used for feature subset selection. It should also be noted that a grouping of attributes as discussed above only reveal that there are relations between attributes but the underlying causal relation must be determined separately by the practitioner.

Hence, by solving the two problems we focus on in this chapter we (i) understand better how the used classification algorithm works, making the predictions more transparent, and (ii) gain insight into the relations

in the data thus giving important information on the structure of the data. We provide several examples of the utility of these topics in real-world data analysis applications below in Section 4.9.2 related to, e.g., pharmacovigilance and data anonymisation.

### 4.1.3 Contributions

The problem of exploring which attribute interactions are exploited by a classifier is very important and has wide applicability. As noted, it is not straightforward to verify how attribute interactions are used by generic, opaque classifiers in making predictions. In Publications III and IV we consider methods for determining attribute interactions utilised by classifiers. In order to identify how classifiers exploit interactions between attributes we utilise constrained randomisations for breaking dependencies between attributes and then observe how this impacts classification performance. In Publication V we present a method for investigating attribute interactions in a dataset in terms of a factorisation of the class-conditional joint distribution of the dataset, with statistical guarantees.

The methods developed in Publications III, IV and V form a framework with two parts. One part is a randomisation test for evaluating whether a dataset originates from a given factorised class-conditional joint distribution (Publication V). The other part of the framework corresponds to partitioning the attributes of a dataset into interacting groups based on optimisation of a value function. We show how this corresponds to an interesting generic clustering problem. Solving this problem allows us to automatically find an optimal grouping of attributes in terms of problems P1 and P2.

We present algorithms that allow problems P1 and P2 to be efficiently solved under certain assumptions on the form of the value function. The `GoldenEye` algorithm (Publication III) is the first method for exploring attribute interactions exploited by generic classifiers. The `GoldenEye++` algorithm (Publication IV) is a modification of the `GoldenEye` algorithm implementing a different value function which is more sensitive in some cases compared to the value function used in the `GoldenEye` algorithm. Both the `GoldenEye` and `GoldenEye++` algorithms address the problem of determining the interactions that are used by a classifier for making predictions (P1). The ASTRID method allows investigating attribute interaction struc-

tures in the data (P2) and is based on a hypothesis testing methodology, using which the significance of the found interactions can be determined.

#### 4.1.4 Organisation of This Chapter

In this chapter we present a framework for investigating attribute interactions. We first review different methods for investigating classifiers and summarise methods related to investigating the structure of data in terms of attribute interactions. We then formalise the discussion of attribute interactions, showing the connection between attribute interactions and the class-conditional joint data distribution, after which we develop the framework. After the framework has been discussed we present algorithms that implement the framework. Finally, we conclude the chapter by providing practical examples related to exploring attribute interactions.

### 4.2 Background

In this section we review methods related to the topics discussed in this chapter.

#### 4.2.1 Methods for Analysing Classifiers

The area of investigating attribute interactions utilised by classifiers is a relatively young field in machine learning and hence there are not many methods in this area, especially methods applicable to the analysis of arbitrary classifiers. Methods for analysing different aspects of classifiers have been presented and we here discuss the methods summarised in Table 4.1, where they are sorted by publication year. Only the methods by Henelius et al. (2014a, 2015) and Henelius et al. (2017) address the problem of determining attribute interactions for generic classifiers.

A randomisation-based method has been presented by Breiman (2001) for analysing the importance of individual attributes. The method was introduced for analysing random forests but is applicable to any classifier. In this method, each individual attribute is permuted at random, one attribute at a time, and the effect on the classification performance is noted. The importance of an attribute correlates with the decrease in accuracy: permuting an important attribute decreases accuracy more than permuting a less important attribute. Based on this, the importance of the attributes can be ranked.

The work by Ojala and Garriga (2010) studies how classifiers utilise the structure in two cases: (i) whether there is a connection between the structure of the data and the class labels predicted by the classifier, and (ii) whether the classifier exploits dependencies between features. Both problems are solved using permutation tests. To solve the first problem the labels in the dataset are permuted at random, producing a dataset where the connection between the features and the data is broken. The empirical $p$-value obtained from the test reflects whether the classifier finds and utilises any connections between the features and the class labels. The second problem is solved by permuting the values for each attribute at random within the classes. This breaks any interdependence between attributes, but keeps the connection between individual attributes and class labels. This randomisation scheme corresponds to the assumptions of the naïve Bayes classifier where the distribution of each attribute is assumed to be independently associated with the class. The $p$-value here reflects whether the classifier exploits interactions in the data. It should be noted that the method by Ojala and Garriga (2010) investigates the question whether or not attribute interactions are used, but it does not allow the identification of the interactions.

Duivesteijn and Thaele (2014) propose the SCaPE model class for exceptional data mining. The goal of this method is to find easily interpretable subgroups of items, in which the performance of a soft classifier is either exceptionally low or high. The identified subgroups can be used by the practitioner to understand the classifier model used in the exceptional model mining.

In the work by Adler et al. (2016) a method is developed for examining black-box models with the goal of rank-ordering individual features based on how they influence the output of the classifier. The importance of an individual attribute is determined by removing ("obscuring") that feature from the rest of the data, such that it cannot be predicted from the other attributes. The decrease in accuracy of a model trained using data where an attribute has been obscured is then due to the particular attribute being considered. This is repeated for each attribute, after which a ranking of attributes is obtained describing the usefulness of each attribute with respect to the original model.

Datta et al. (2016) present a method for investigating how individual

features influence on the outcome of an algorithm. The method is aimed at explaining decisions made by an automated system and is based on breaking correlations between input features and observing how this impacts some quantity of interest.

Ribeiro et al. (2016) present a method termed LIME for explaining individual predictions, i.e., one item, from any black box classifier using easily interpretable representation such as, e.g., linear models. The method reveals the features in the data item that contributed to the decision and thus provides a local interpretation of the classifier for that data item. A global explanation of the model is obtained by combining local explanations for individual items, such that the local explanations explain as many different instances as possible.

Turner (2016) presents the *Model Explanation System* for explaining individual predictions of binary classifiers in terms of logical statements concerning the features.

### 4.2.2 Methods for Determining Attribute Interactions

Several methods for investigating attribute interactions in general have been proposed. For a review on the topic of attribute interactions in data mining see, e.g, Freitas (2001). An information-theoretical approach for quantifying the degree of interaction has been proposed (limited to three-way interactions) by Jakulin et al. (2003) and Jakulin and Bratko (2003), Jakulin and Bratko (2004) considered factorising the joint distribution of the data and presented a method for testing the significance of attribute interactions (limited to two and three-way interactions). The role of attribute interactions in feature selection has been studied by, e.g., Zhao and Liu (2007, 2009). Tatti (2011) considered ordering the attributes of a dataset such that successive attributes are maximally dependent while Mampaey and Vreeken (2013) presented a method for clustering correlated attributes into groups. Bayesian network learning is also concerned with finding the structure of a dataset in terms of factorising the joint data distribution, see, e.g., Koski and Noble (2012). Furthermore, variable and feature selection is a related problem, see, e.g., Guyon and Elisseeff (2003).

**Table 4.1:** Summary of methods for analysing classifiers.

| Method | Goal | Scope |
|---|---|---|
| Breiman (2001) | Rank individual attributes by importance | Generic, i.e., can be used with any classifier |
| Ojala and Garriga (2010) | Assess (i) if there is any structure in the data and (ii) if the classifier uses attribute interactions | Generic |
| Henelius et al. (2014a), Henelius et al. (2015), Henelius et al. (2017) | Discover groups of interacting attributes | Generic |
| Duivesteijn and Thaele (2014) | Finding subgroups where the classifier performs especially good / bad | Soft classifiers in exceptional model mining with binary classes |
| Adler et al. (2016) | Rank individual attributes by importance | Generic |
| Datta et al. (2016) | Determine the influence input features have on the outcome | Generic |
| Ribeiro et al. (2016) | Explain the predictions of a classifier in terms of the attributes that contribute to the outcome | Generic |
| Turner (2016) | Explain individual predictions | Generic, for binary classifiers |

## 4.3 Attribute Interactions

In this section we formalise the concept of attribute interactions and class-conditional joint data distribution since these concepts are central to the ideas developed in this chapter.

With *attribute interactions* is meant that two or more attributes contain complementary information needed in the prediction task. As an example, in the study by Jakulin et al. (2003) it was found that neurological disease was a high risk factor only in the presence of other complications during surgery, i.e., there is an *interaction* between these two attributes

and they must be taken into account jointly when making predictions. In other words, the attributes are useless by themselves whereas together they are useful for prediction. This general definition of attribute interactions is valid both when considering attribute interactions exploited by a classifier and when considering attribute interactions in the data. Next we present the notation used in this chapter and the relation between attribute interactions and a factorisation of the joint distribution of the data.

### 4.3.1 Notation

Let $X$ be an $n \times m$ data matrix (see Section 2.1). We denote the $i$th row (data item) in $X$ by $X(i,\cdot)$ and the $j$th column (attribute) by $X(\cdot,j)$. We denote a subset of the columns of $X$ by $X(\cdot,S)$, where $S \subseteq [m]$, using the shorthand notation $[m[ = \{1,\ldots,m\}$. Each data item in $S$ is associated with a class label from a finite set of class labels $\mathcal{C}$. Let $C$ be a vector of class labels where $C(i)$ is the class label for $X(i,\cdot)$. The tuple $D = (X,C)$ containing the data matrix $X$ and the vector of class labels $C$ forms a dataset $D \in \mathcal{D}$, where $\mathcal{D}$ is the set of all possible datasets.

A *grouping* $\mathcal{S}$ is a disjoint partition of the set of attributes of $X$ into $k$ groups $S_i$ such that $\mathcal{S} = \{S_1,\ldots,S_k\}$ and $S_i \cap S_j = \emptyset$ for all $i \neq j$. All attributes are included in some group in the grouping, i.e., $\bigcup_{S\in\mathcal{S}} = \{1,\ldots,m\}$. The size of a grouping $\mathcal{S}$ is $k$ if $|\mathcal{S}| = k$. We denote the set of all possible partitions of the attributes of $X$ by $\mathcal{P}$.

The dataset $D$ now follows a joint probability distribution of the form

$$\Pr(D) = \overbrace{\prod_{i\in[n]} \Pr(X(i,\cdot) \mid C(i))}^{\Pr(X|C)} \Pr(C(i)), \tag{4.4}$$

where $\Pr(X \mid C)$ is the *class-conditional distribution*, which is of interest to us. We here consider a factorisation of the joint distribution into class-conditional factors given by a grouping $\mathcal{S}$ and write

$$\Pr(D) = \overbrace{\prod_{i\in[n]} \prod_{S\in\mathcal{S}} \Pr(X(i,S) \mid C(i))}^{\prod_{S\in\mathcal{S}} \Pr(X(\cdot,S)|C)} \Pr(C(i)). \tag{4.5}$$

We define that the factorisation in factors given by $\mathcal{S}$ describes the structure of the dataset in terms of the attribute interactions, such that attributes in the same group are *interacting* and attributes in different

groups are *independent* of the attributes in the other groups. Finally we illustrate the concept of attribute interactions by an example.

**Running Example** We use this as a running example of attribute interactions in a dataset. Assume that in a dataset with six attributes $1, 2, \ldots, 6$ we have found that attributes 1, 2 and 3 are important together for classification, and similarly 4 and 5 are important whilst the information provided by attribute 6 is, in fact, unimportant for the classification. The interaction between the attributes in the dataset is hence represented by a grouping of attributes with three groups:

$$\mathcal{S} = \{S_1, S_2, S_3\} = \{\{1, 2, 3\}, \{4, 5\}, \{6\}\},$$

where attributes in the same group interact with each other but are unrelated to the attributes in the other groups.

### 4.4 Framework for Investigating Attribute Interactions

Our goal is to develop a framework for investigating attribute interactions as outlined in Section 4.1.2. The framework consists of two parts:

1. *Identification:* automatically finding a suitable partition of the attributes of the dataset, such that the partition reflects the attribute interactions, either in terms of P1 or P2 stated above.

2. *Testing:* assessing the hypothesis that a given dataset has been sampled from a factorised joint distribution where the factors correspond to groups in a particular grouping $\mathcal{S}$, describing the attribute interaction structure of the dataset.

The two parts in the framework are independent. Note that the hypothesis testing part of the framework applies only to cases where we explore interactions in the dataset (P2), not when we consider interactions exploited by the classifier (P1).

We next develop the two parts of the framework, starting with the *testing* part. Both parts of the framework rely on the relationship between classifier performance and attribute interactions and we hence consider this next.

## 4.5 Classifier Performance and Attribute Interactions

We assume that high-performing classifiers learn to exploit attribute interactions in order to make accurate predictions. We also assume that the dataset contains a meaningful structure.

Consider a dataset where the attributes are interacting. Assume that the classifier correctly learns to exploit the interactions and therefore reaches high classification performance. If we now, using some method, break an interaction in the dataset important for the classification task (e.g., the interactions in the group $S_1$ in the running example), the classifier can no longer utilise these interactions and classification performance is negatively impacted. However, if an interaction in the data not utilised by the classifier is broken, e.g., by perturbing the values of attribute 6 in the running example, classification performance is not impacted.

This observation concerning the connection between classification performance and attribute interactions is important and forms the basis for our framework. This suggests that classifier performance combined with breaking of attribute interactions can be used as a proxy for studying attribute interactions used by the classifier: if we break important relationships the performance is negatively affected but if we break unused relationships the performance is unchanged or is only marginally affected.

Based on the above discussed connection between the attribute interactions and the joint distribution of the data (Equation (4.5)) we conclude that if we in a controlled manner *generate versions of a dataset where some known interactions are intact, while all other interactions are broken* we can investigate how the classifier performs on these different datasets and gain insight into which attribute interactions the classifier exploits. Generation of such datasets is discussed in the next section.

### 4.5.1 Constrained Permutation of Datasets

In this section we consider the permutation of a dataset such that attribute interactions are broken in a controlled fashion. The permutation scheme is parametrised by a grouping of attributes $\mathcal{S} \in \mathcal{P}$. We also consider the relation between the constrained permutation schemes and the joint distribution of Equation (4.5).

There are two important aspects of a dataset that can be easily manipulated using different permutation schemes: (i) the distribution of

attribute values, either without respect to class or within class, and (ii) the interactions between different attributes.

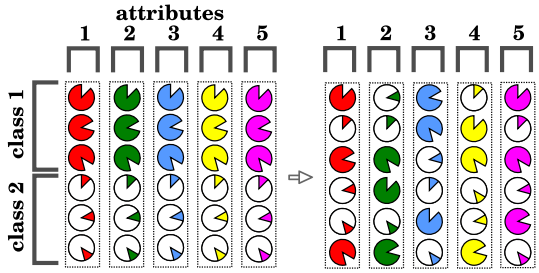
We next consider three randomisation schemes based on manipulating these aspects that can be used to break different attribute interactions in a dataset $D$:

**R1** Randomly permute the values of one / each attribute at random

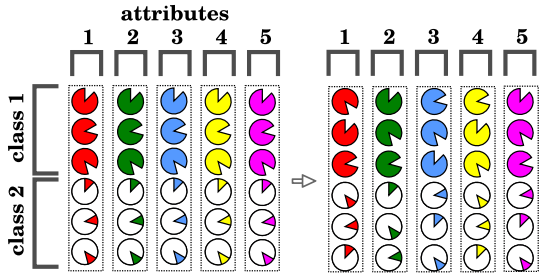
**R2** Randomly permute the values of one / each attribute within class

**R3** Randomly permute the values of the attributes in each group together within-class, separately within each group in the grouping $\mathcal{S}$.

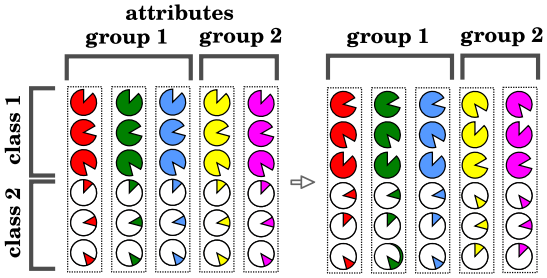
These three randomisation schemes employ (i) random permutations of attribute values and (ii) within-class permutations in which the values of an attribute are permuted separately within each class, thus not affecting the marginal distribution of values for that attribute within each class. The effect of these three randomisation schemes are visualised in Figure 4.1.




**(a)** Random permutation disregarding classes.




**(b)** Random permutation within class.




**(c)** Random permutation within class and within group.

**Figure 4.1:** Different randomisation schemes applied to a binary dataset with five attributes and six items. Original data on the left and after permutation on the right. Classes are denoted by filled glyphs (class 1) and unfilled glyphs (class 2). Different attributes (vertical columns) are marked by different colours. The angle of the sector in the glyph marks the row in the data matrix, the angle increases clockwise towards the bottom of the matrix. In (a) each attribute is permuted individually at random without regards to the class. In (b) each attribute is permuted individually within-class. In (c) the attributes are divided into two groups; each group is then permuted within group and within class.

**Full Random permutation (R1)** When randomisation scheme R1 is applied to one particular attribute it breaks the connection between the distribution of values for the attribute and the class labels. This scheme can hence be used to investigate if the value distribution of a particular attribute is independent of the class label, i.e., whether the attribute carries any class-related information. This randomisation scheme is used to determine attribute importance in the random forest algorithm (Breiman, 2001) discussed above.

When randomisation scheme R1 is applied to all attributes as in Figure 4.1a, such that each column is permuted independently, it clearly breaks all connections between the information in the attributes and the class label and renders it impossible for a classifier to learn the relationship between the distribution of values for the attributes and the class labels. This full randomisation scheme can be used to test the hypothesis whether there is any structure at all in the data. This can also be compared to the scheme of randomly permuting the class labels used by Ojala and Garriga (2010) to investigate if the classifier has found any real structure in the data.

**Individual Within-Class Permutation (R2)** Randomisation scheme R2 applied to all attributes imposes the constraint on the randomisation that the values for each attribute are permuted individually within each class, but each attribute is treated separately from the other attributes. In other words, each column is permuted independently within-class as in Figure 4.1b. This randomisation scheme matches the independence assumptions of the naïve Bayes classifier (see Section 2.2.2). This is also the randomisation scheme used by Ojala and Garriga (2010) to investigate whether a classifier is exploiting dependencies between attributes.

**Groupwise Within-Class Permutation (R3)** Randomisation scheme R3 is parametrised by the grouping $\mathcal{S}$. Here the attributes in the same group in $S_i$ are permuted together as in Figure 4.1c. This keeps interactions between attributes in the same group intact, while breaking interactions between attributes in different groups, since the groups are permuted independently. This randomisation scheme hence allows us to break interactions in a dataset in a controlled manner parametrised by a grouping of attributes $\mathcal{S}$.

In terms of the factorised joint distribution of Equation (4.5), the constraints imposed by randomisation scheme R3 ensures that the class-conditional joint distribution of the attributes remains intact since attributes in the same group are permuted together and different attribute groups are permuted individually.

We can formally define permutation scheme R3 as follows. Let $\pi_i : [n] \rightarrow [n]$ be one of $m$ bijective permutation functions from the set of constrained permutation functions $\Pi_{\mathcal{S}}$. We denote the permutation of the dataset $D = (X, C)$ parametrised by $\mathcal{S} \in \mathcal{P}$ by $D^{\mathcal{S}} = \left(X^{\mathcal{S}}, C\right)$, where $X^{\mathcal{S}}(i,j) = X\left(\pi_j(i), j\right)$. The constrained permutation functions in $\Pi_{\mathcal{S}}$ satisfy the following properties:

- $C\left(\pi_j(i)\right) = C(i)$ for all $i \in [n]$, i.e., the class labels do not change since the permutations are within-class.
- For any two attributes $j, j' \in S$, where $S \subset \mathcal{S}$, it holds that $\pi_j = \pi_{j'}$, i.e., the same permutation function is applied to all attributes that belong to the same group and all attributes in the group are hence permuted together.

To put these randomisation schemes into context, consider the running example with the dataset with six attributes and the known structure $\mathcal{S} = \{S_1, S_2, S_3\} = \{\{1,2,3\}, \{4,5\}, \{6\}\}$, where attribute 6 does not contain information on the class. Assume that a classifier $f$ utilises these interactions in making predictions. Recall, that we assume that the attributes in each group are interacting and that they are independent of the attributes in the other groups. Here randomisation scheme R1 would prevent the classifier from making correct predictions as the interactions between attributes is broken. Randomisation scheme R2 would similarly render the classifier useless while randomisation scheme R3 parametrised by a grouping of the form $\mathcal{S} = \{\{1,2,3\}, \{4,5\}, \{6\}\}$ would still allow the classifier to exploit attribute dependencies since the interactions between the attributes in each group are unaffected by the permutation. In this example we could also permute attribute groups $\{1,2,3\}$ and $\{4,5\}$ within-class (scheme R3) and the singleton attribute 6 at random (scheme R1) since attribute 6 carries no class information.

Permutation scheme R3 parametrised by different attribute groupings hence allows us to break interactions in a dataset in a systematic manner

and investigate how this impacts classification performance.

In the following we will, for brevity, refer to scheme R3 when we say that we permute a dataset using a grouping $\mathcal{S}$. We also use the shorthand $D^{\mathcal{S}} = \left(X^{\mathcal{S}}, C\right)$ to denote that $D$ has been permuted using scheme R3 parametrised by $\mathcal{S}$. Note here that the class vector $C$ remains the same since the permutations are within-class, i.e., class labels for items do not change.

## 4.6 Hypothesis Testing of Data Distribution

We now consider the *testing* part of the framework. Here we are interested in examining the structure of the data in terms of a factorisation of the class-conditional joint distribution of the data in Equation (4.5), where the structure is described in terms of the grouping $\mathcal{S}$. Given an observed dataset $D_0$, we formulate a null hypothesis:

**Hypothesis 4.1.** *The observed dataset $D_0$ has been sampled from the distribution given by Equation* (4.5) *with the groups given by $\mathcal{S} \in \mathcal{P}$.*

In order to test this hypothesis we employ the hypothesis testing framework described in Section 2.3 for which we need (i) a test statistic and (ii) the distribution of the test statistic under the null hypothesis. We consider these next.

### 4.6.1 Test Statistic

We want to test the hypothesis that a particular grouping $\mathcal{S}$ represents the factorisation of Equation (4.5). As noted, our test is based on the following rationale: if a classifier $f_2$ trained using data from a factorised distribution is indistinguishable from a classifier $f_1$ trained using data from an unfactorised distribution, we conclude that the factorisation is valid. The test statistic must hence reflect how well the grouping $\mathcal{S}$ represents the structure of the data. As discussed above, we can assume that a classifier must learn the structure of the data and hence the quality of a grouping is reflected in classification performance. We here choose to use classification accuracy (See Section 2.2.1) as our test statistic. Assume now that we have an observed dataset $D_0$ and that we also have a separate test dataset $D_{\text{test}} = (X_{\text{test}}, C_{\text{test}})$ with $n_{\text{test}}$ items sampled from the same distribution as $D_0$. We here use the notation $f_D$ to denote a classifier $f$ that has been

trained using the dataset $D = (X, C)$ and we denote the predictions of $f$ on a data matrix $X$ as $f(X)$. The test statistic $T$ can now be defined as

$$T(D) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} I\left[f_{D^S}(X_{\text{test}}(i,\cdot)) = C_{\text{test}}\right], \tag{4.6}$$

where $I$ is the indicator function. It should be noted that we could also use other test statistics based on classifier performance metrics that are affected by breaking interactions, e.g., the $F_1$ measure.

For the hypothesis testing we need samples of the test statistic under the null hypothesis, to which the value of the test statistic for the original data can be compared. We next consider how to obtain these samples.

### 4.6.2 Sampling Datasets Under the Null Hypothesis

The distribution of the test statistic under the null hypothesis cannot in general be determined analytically and instead we use the following method to sample the test statistic.

We apply permutation scheme R3 parametrised by the grouping $\mathcal{S}$ to the observed dataset $D_0$, which generates a new dataset $D^{\mathcal{S}} = \left({X_0}^{\mathcal{S}}, C\right)$. A particular grouping $\mathcal{S}$ defines a subset $\mathcal{D}_{\mathcal{S}} \subseteq \mathcal{D}$ of the space of all datasets $\mathcal{D}$. Here $\mathcal{D}_{\mathcal{S}}$ is the set of all datasets that can be generated from $D_0$ using permutation scheme R3 parametrised by $\mathcal{S}$. We here show that permutation scheme R3 yields i.i.d. datasets uniformly from $\mathcal{D}_{\mathcal{S}}$.

**Lemma 4.1.** *Given a dataset $D_0 \in \mathcal{D}_{\mathcal{S}}$, permutation scheme R3 generates all datasets in $\mathcal{D}_{\mathcal{S}}$ with equal probability.*

*Proof.* Let $D, D' \in \mathcal{D}_{\mathcal{S}}$. Denote by $\Pr(D \to D')$ the probability that $D$ is transformed into $D$ using permutation functions from $\Pi_{\mathcal{S}}$. For any two datasets $D, D' \in \mathcal{D}_{\mathcal{S}}$ it holds that $D$ can be transformed into $D'$ in one step. It follows that $\Pr(D \to D') = \Pr(D' \to D)$ since the reverse transformation is equally likely.. □

We next show that the datasets in $\mathcal{D}_{\mathcal{S}}$ are uniformly distributed.

**Lemma 4.2.** *The datasets in $\mathcal{D}_{\mathcal{S}}$ are uniformly distributed given that an observed dataset $D_0 \in \mathcal{D}_{\mathcal{S}}$ follows the factorised distribution of Equation* (4.5)*.*

*Proof.* Permutation of a dataset $D_0 \in \mathcal{D}_{\mathcal{S}}$ using scheme R3 parametrised by $\mathcal{S}$ only changes the order of rows within-class for each group in $\mathcal{S}$. This

means that $\Pr(X(\cdot, S) \mid C)$ remains the same for all $S \in \mathcal{S}$. It follows that $\Pr(D) = \Pr\left(D^{\mathcal{S}}\right)$, i.e., the datasets in $\mathcal{D}_{\mathcal{S}}$ are uniformly distributed. □

We now state

**Theorem 4.1.** *Applying permutation scheme R3 parametrised by a grouping $\mathcal{S}$ to a dataset $D_0 \in \mathcal{D}_{\mathcal{S}}$ generates i.i.d. samples uniformly from $\mathcal{D}_{\mathcal{S}}$.*

*Proof.* The proof follows directly from Lemma 4.1 and Lemma 4.2. □

The distribution of the test statistic under the null hypothesis is now obtained by sampling $R$ datasets ${D_1}^{\mathcal{S}}, \ldots {D_R}^{\mathcal{S}}$ from $\mathcal{D}_{\mathcal{S}}$ and computing the value of the test statistic from these. We now compare the value of the test statistic from $D_0$ to the values obtained from the datasets sampled from $\mathcal{D}_{\mathcal{S}}$, and compute the empirical $p$-value (Section 2.3):

$$p_{\mathcal{S}} = \frac{1 + \sum_{i=1}^{R} I\left[T\left({D_i}^{\mathcal{S}}\right) \geq T(D_0)\right]}{1 + R}, \tag{4.7}$$

where $I$ is the indicator function. Finally, we formulate the following problem:

**Problem 4.1.** *Given an observed dataset $D_0 \in \mathcal{D}$, a grouping $\mathcal{S} \in \mathcal{P}$ and a classifier $f$, determine at a significance level $\alpha \in [0, 1]$ if $D_0$ has been sampled from the factorised distribution of Equation* (4.5) *with the groups given by $\mathcal{S}$.*

Summarising, in order to solve Problem 4.1 we

1. Compute the value of the test statistic $T$ using the original dataset
2. Determine the distribution of the test statistic under the null hypothesis based on datasets sampled using the above described permutation scheme parametrised by the grouping $\mathcal{S}$ which we are investigating
3. Compute $p_{\mathcal{S}}$ using Equation (4.7).

If $p_{\mathcal{S}} > \alpha$ we do not reject the null hypothesis and hence conclude that the dataset $D_0$ can have been sampled from the factorised distribution of Equation (4.5) with the groups given by $\mathcal{S}$.

The implication of this test is also that a classifier trained using data from the factorised distribution is indistinguishable from a classifier trained using data from the unfactorised distribution.

**A Note on the Statistical Significance Testing** Our goal is to find such a grouping of the attributes that the performance of the classifier trained using the grouping is as close as possible to the performance using the original data where the attributes have not been partitioned. We hence define the null hypothesis as above in Hypothesis 4.1 where the comparison is made to the original data.

When we from Equation (4.7) obtain $p \leq \alpha$ at a significance level $\alpha$, we conclude that some interactions in the data used by the classifier were broken, which leads to decreased performance. However, when we obtain $p > \alpha$, we can only conclude that we fail to reject the null hypothesis that the data has been sampled from a factorised distribution with the groups given by the grouping $\mathcal{S}$ currently being investigated. It should be noted that this does not mean that the null hypothesis is true, only that there is no evidence to reject it.

## 4.7 Identifying Attribute Interactions

In this section we consider the *identification* part of the framework, which focuses on how to automatically find a grouping of attributes in the two cases outlined in Section 4.1.2:

1. attribute interactions in a dataset
2. attribute interactions exploited by a classifier in making predictions

As noted above, these two types of interactions are highly related but not necessarily identical. Next we show how these two cases are instances of a general clustering problem, where attributes are grouped in terms of a value function describing the quality of a particular grouping. We define the general clustering problem and in Section 4.8 we present algorithms for solving these problems. The two cases of attribute interactions require different value functions and we define these below. Assume, for now, that there is a value function $\widehat{T} : \mathcal{S} \in \mathcal{P} \rightarrow \mathbb{R}$ describing the quality of a particular grouping of attributes.

The *testing* part of the framework allows us to verify whether *a particular* grouping $\mathcal{S}$ represents a valid factorisation of the class-conditional joint data distribution. However, this method cannot be used to *find* a suitable grouping $\mathcal{S}$. The number of possible partitions of attributes of a dataset $D$ with $m$ attributes is given by the $m$th Bell number and hence grows

exponentially with the number of attributes. This means that in general it is not possible to exhaustively attempt to find a grouping $\mathcal{S}$ representing the attribute interactions of a dataset by testing the significance of each of these.

We now consider the problem of investigating attribute interactions exploited by classifiers. In this case we consider a classifier $f$ and a dataset $D$ and are interested in determining how $f$ exploits the attribute interaction structure in $D$. In brief, we do this by computing the value of the grouping $\mathcal{S}$ using $\widehat{T}$, which allows us to compare different groupings. However, it is again not in general possible to do this exhaustively.

Both these problems are instances of the following general clustering problem:

**Problem 4.2.** *Given a dataset $D_0$ with $m$ attributes, a constant $k \in [m]$, a classifier $f$ and a value function $\widehat{T}$, find a partition $\mathcal{S}$ of the attributes of $D$ into $k$ groups such that the value of $\widehat{T}$ is maximised.*

This problem of partitioning the attributes into interacting groups hence represents an interesting but atypical clustering problem. In usual clustering problems a distance function between different data items can be defined, which is not possible here. The grouping of attributes must hence be accomplished in terms of a value function, computed separately for each candidate grouping $\mathcal{S}$. The value function depends on which of the above discussed attribute interaction problems we are solving.

Recall that the value function is of the form $\widehat{T} : \mathcal{P} \mapsto \mathbb{R}$ and assigns a value to the grouping $\mathcal{S}$. Also recall that we use the above described permutation schemes to generate datasets where attribute interactions are modified according to the grouping $\mathcal{S}$. We now discuss the choice of value function.

**Attribute Interactions in a Dataset** Here we assume that if there are class-dependent attribute interactions in a dataset, then a classifier $f$ can learn these and achieve high performance. However, if we break the interactions in the data prior to training the classifier, the interactions in the data become unavailable and the classifier cannot utilise these, which impacts performance. We here hence investigate the question of *what attribute interactions in the data are needed to train a high-performing classifier*.

To investigate attribute interactions in a dataset we use the following scheme. Assume that we have an observed dataset $D = (X, C)$ that has been sampled from a distribution given by Equation (4.5) with the groups given by $\mathcal{S}$. Also assume that we have a test dataset $D = (X_{\text{test}}, C_{\text{test}})$ sampled from the same distribution as $D$. We here define the value function in terms of accuracy as follows, using the test statistic $T$ in Equation (4.6):

$$\widehat{T}_{\text{acc}}(\mathcal{S}) = \frac{1}{N}\sum_{i=1}^{N} T\left({D_i}^{\mathcal{S}}\right), \tag{4.8}$$

where $N$ is the number of samples used to calculate the expected value of the value function. In this value function we compute the expected accuracy of a classifier trained using randomised data, when predicting the class labels of an unpermuted test dataset. This value function can hence be used to explore which attribute interactions are needed to train a useful classifier.

**Interactions Exploited by Classifiers** We now consider the problem of finding the attribute interactions in a dataset $D$ exploited by a classifier.

Assuming that a classifier $f$ uses some attribute interactions in making predictions, then classification performance is reduced if we break those interactions. We here answer the question of *what interactions does a trained classifier $f$ exploit*.

As the value function we here use *fidelity*, which describes the degree to which classifications change due to randomisation. We now define the value function as follows.

$$\widehat{T}_{\text{fid}}(\mathcal{S}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n}\sum_{j=1}^{n} I\left[f_D(X(j,\cdot)) = f_D\left(X^{\mathcal{S}}(j,\cdot)\right)\right], \tag{4.9}$$

where $I$ is the indicator function, $N$ is the number of samples used to calculate the expected value and $n$ is the number of items in $X$ and $X^{\mathcal{S}}$. This value function hence measures the average fraction of matching predictions by $f$ on the original dataset and on a randomised dataset. If the classifier $f$ trained using the original data $D$ is exploiting some interactions in $D$ for making predictions, then classification performance decreases if permuting $D$ by $\mathcal{S}$ breaks any of these interactions. This is consequently reflected in fidelity.

### 4.7.1 Monotonicity of Classification Performance

In order to provide polynomial-time algorithms for solving Problem 4.2 in the cases discussed above we must make an assumption regarding the used value function. We here assume that the value functions are monotonic with respect to breaking of groups in an unknown optimal partition of attributes $\mathcal{S}_O$. As an example, let $\mathcal{S}_O = \{S_1, S_2, \ldots, S_k\}$ and let $\mathcal{S}' = \{S_{1a}, S_{1b}, S_2, \ldots, S_k\}$ where $S_{1a} \cap S_{1b} = \emptyset$ and $S_{1a} \cup S_{1b} = S_1$. Monotonicity implies that $\widehat{T}(\mathcal{S}') < \widehat{T}(\mathcal{S}_O)$ since the group $S_1$ in the optimal partition has been broken.

Formally we define monotonicity as follows.

**Definition 4.1.** *Let $\widehat{T}$ be a value function and let $\mathcal{S}_O$ be a grouping of size $k \leq m$. $\widehat{T}$ is monotonic with respect to $\mathcal{S}_O$ if for every $\mathcal{S}, \mathcal{S}' \in \mathcal{P}$ for which $F(\mathcal{S}) \subset F(\mathcal{S}')$ it holds that $\widehat{T}(\mathcal{S}) < \widehat{T}(\mathcal{S}')$. Here we use the notation $F(\mathcal{S}) = \{X \in \mathcal{S}_O \mid \exists Y \in \mathcal{S} : X \subseteq Y\}$.*

Furthermore we state that

**Theorem 4.2.** *If the value function $\widehat{T}$ is monotonic with respect to a partition $\mathcal{S}_O$ of size $k$, it follows that $\mathcal{S}_O$ is the solution to Problem 4.2.*

*Proof.* $\mathcal{S}_O$ is the solution to Problem 4.2 since for any other $\mathcal{S} \neq \mathcal{S}_O$ of size $k$ it holds that $\widehat{T}(\mathcal{S}) \leq \widehat{T}(\mathcal{S}_O)$. □

Hence, an algorithm that finds the optimal grouping $\mathcal{S}_O$ in Definition 4.1 also solves Problem 4.2. Below when presenting the ASTRID method we provide a proof of how the algorithm finds the optimal solution to Problem 4.2, assuming monotonicity.

It should be noted that monotonicity is an intuitive heuristic assumption required to devise efficient top-down algorithms for investigating attribute interactions used by classifiers, discussed below in Section 4.8. Although it cannot be shown that the above discussed value functions in Equation (4.8) and Equation (4.9) are monotonic, the assumption of monotonicity appears to work well in practice for these; an example using accuracy is shown below. The assumption of monotonicity can be compared to, e.g., the assumption of normality of data; even though the assumption in most cases does not hold perfectly it is still useful. If the monotonicity property does not hold the solution to Problem 4.2 is approximate.

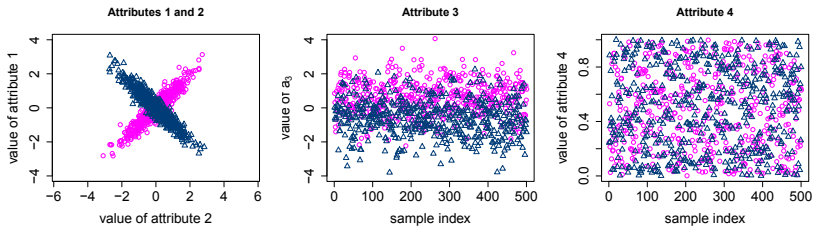



**Figure 4.2:** A synthetic four-attribute dataset with the known structure of $\mathcal{S}_O = \{\{1,2\},\{3\},\{4\}\}$. The dataset has two classes shown as pink circles (class 1) and blue triangles (class 2).

**Example of Monotonicity** We illustrate monotonicity of classification performance using the value function in Equation (4.8) and the synthetic data from Publication V shown in Figure 4.2. The synthetic dataset has four attributes and two classes, each with 500 items. The items in class 1 are shown as pink circles and items in class 2 as blue triangles. The dataset is constructed as follows: attributes 1 and 2 carry meaningful information regarding the class only when considered jointly, whereas attribute 3 carries some weak information on its own and attribute 4 is just random noise. The known attribute interaction structure of this dataset is hence given by $\mathcal{S}_O = \{\{1,2\},\{3\},\{4\}\}$.

Figure 4.3 shows the possible partitions into groups of size $k = 1$ (level 1) to $k = 4$ (level 4). For each partition, the average accuracy (Equation (4.8)) and the $p$-value (Equation (4.7)) is shown. Both were computed from 250 random samples, obtained by shuffling the data according to the given grouping and using an SVM classifier with a radial basis function kernel. The figure shows that when the group $\{1,2\}$ in $\mathcal{S}_O$ is broken the average accuracy decreases as the joint information in attributes 1 and 2 cannot be utilised by the classifier.

The groupings in which the group $\{1,2\}$ in $\mathcal{S}_O$ is intact are highlighted in blue. For these groupings the $p$-value is also above 0.05, indicating that we cannot reject the null hypothesis that the data originates from a class-conditional joint distributed parametrised by the given grouping. It should be noted that the minimum accuracy for the partitions where the group $\{1,2\}$ is intact, marked with blue, is higher than the maximum accuracy for the partitions where the group is broken. Classification accuracy is in this example hence monotonic with respect to breaking of groups.

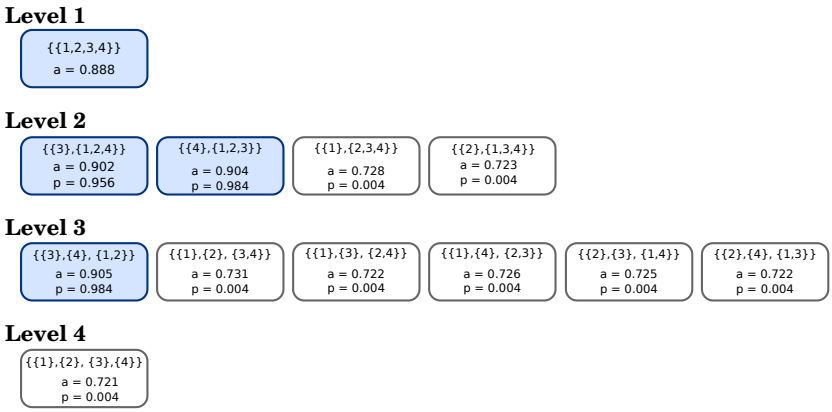



**Figure 4.3:** An example illustrating monotonicity of average classification accuracy using the synthetic dataset shown in Figure 4.2 with the known structure of $\mathcal{S}_O = \{\{1,2\},\{3\},\{4\}\}$. The figure shows the partitions of attributes into groups of size $k = 1$ (level 1) to $k = 4$ (level 4). The average accuracy in Equation (4.8) is denoted by $a$ while $p$ is the $p$-value ($p_{\mathcal{S}}$) in Equation (4.7). Partitions in which the groups in $\mathcal{S}_O$ are intact are highlighted in blue. The values are obtained using an SVM classifier with a radial basis function kernel, using 250 random samples.

### 4.7.2 Summary of the Framework

We have here presented a framework for investigating different types of attribute interactions and for testing the hypothesis that a dataset originates from a factorised class-conditional joint distribution. In the next section we present polynomial-time algorithms for solving Problem 4.2.

We first consider the ASTRID method, presented in Publication V. This method incorporates both parts of the framework and utilises the value function in Equation (4.8). The ASTRID is used to automatically find the grouping of attributes that describes the factorised form of the class-conditional joint distribution from which the dataset can have been sampled.

After this we consider the GoldenEye algorithm, presented in Publication III. Here we use the value function of Equation (4.9). We then consider the GoldenEye++ algorithm, which is a variant of the GoldenEye algorithm using a different value function.

It should be noted that the automatic partitioning algorithm in the ASTRID method can be used with both value functions discussed here and can hence solve both of the attribute interaction problems we consider.

## 4.8 Algorithms for Analysing Classifiers

In this section we present three algorithms corresponding to the *identification* part of the framework for solving Problem 4.2. The main principle behind the algorithms can briefly be summarised as follows. We employ the concepts in the framework described in Section 4.4 in the following way. We systematically break interactions between attributes in the dataset and observe how this impacts classification performance, allowing us to identify interactions between attributes. The monotonicity assumption is important and allows us to employ a top-down greedy approach, i.e., we start with all attribute interactions present in the dataset corresponding to a grouping with all attributes in one group, i.e., $|\mathcal{S}| = 1$. At each iteration we break down the structure of the data one step at a time by greedily separating the attribute that impacts accuracy the least into a singleton group in the grouping.

### 4.8.1 The ASTRID Method

The ASTRID method implements both parts of the framework of Section 4.4 and consists of two main steps: (1) sorting and (2) grouping. These steps result in $m - 1$ groupings, where the attributes have been optimally partitioned into $k = 2, \ldots, m$ groups. In Problem 4.2 we need to specify the cardinality of the grouping, i.e., $k$. Instead of directly specifying the cardinality $k$, we use the method developed in the testing part of the framework as a heuristic for model selection by selecting the grouping with the highest value for $k$ for which $p_{\mathcal{S}} > \alpha$ in Equation (4.7), at the significance level $\alpha$. No correction for multiple comparisons is made. The $p$-value is hence used to select the grouping and this provides the solution to Problem 4.2.

The ASTRID method uses a classifier, a training dataset and a testing dataset. Consider a dataset $D$ with $m$ attributes. We assume that there is an optimal grouping of attributes $\mathcal{S}_{\text{opt}} = \{S_1, \ldots, S_k\}$ describing the structure of the dataset in terms of attribute interactions. We further assume that the used classifier $f$ has learned to predict the classes with a sufficiently high accuracy. As an example we consider the above discussed six-attribute dataset with the known attribute interactions given by the grouping $\{\{1, 2, 3\}, \{4, 5\}, \{6\}\}$.

**Sorting Step** In the sorting step (Figure 4.4) the attributes are ordered groupwise. The starting condition in the sorting step is a full grouping, i.e., $\mathcal{S} = \{\{1, 2, 3, 4, 5, 6\}\}$ and $|\mathcal{S}| = 1$. At each step, the algorithm detaches one attribute from $\mathcal{S}$ into a singleton group, forming a new grouping $\mathcal{S}'$. The attribute to detach is found by greedily choosing the attribute which impacts performance the least, i.e., the attribute for which the value function $\widehat{T}_{\text{acc}}$ remains highest is detached. This greedy procedure continues until the grouping $\mathcal{S}'$ is all-singleton.

It follows from monotonicity that attributes are detached groupwise. Let $S$ be one of the groups in the optimal solution $\mathcal{S}_O$ and let $s \in S$ be one of the attributes in $S$. Now, once $s$ is detached it follows that the joint distribution of the attributes in $S$ is broken and is no longer useful for predicting the class. Hence, assuming monotonicity, it follows that detaching the rest of the attributes in $S$ one by one decreases performance less than detaching an attribute $s' \notin S$ from another, yet unbroken, group in the optimal solution.

**Grouping Step** The starting point for the grouping step (Figure 4.5) is the ordered grouping from the sorting step in which it is known that the attributes are ordered groupwise, but the group borders are unknown. In the grouping step the borders between the groups are determined.

The ordered set of attributes can be split into two halves in $m - 1$ ways without reordering the attributes, giving $m - 1$ attribute groupings $\mathcal{S}_i$ for which $|\mathcal{S}_i| = 2$. The value of $\widehat{T}_{\text{acc}}$ is determined for each of these groupings.

From the monotonicity assumption it follows that the accuracy is highest when the attributes are split so that the split occurs at a group border, i.e., when both of the groups contain only full groups in the optimal solution $\mathcal{S}_O$. An example of this is given by groupings (a) and (c) in Figure 4.5. A grouping of size $k$ can then be obtained from the group borders of the top $k - 1$ groupings with the highest accuracy. In the example, grouping (a) is split after attribute 6 and grouping (c) is split after attribute 5. The optimal 3-grouping is hence obtained when the original grouping $\{\{6, 4, 5, 2, 1, 3\}\}$ is split after attribute 6 and attribute 5, giving $\{\{6\}, \{4, 5\}, \{2, 1, 3\}\}$. After the grouping step the optimal $k$-grouping can hence be calculated directly for any $k \in [1, m]$.

Finally, we determine the value of $p_{\mathcal{S}}$ in Equation (4.7) for each of the

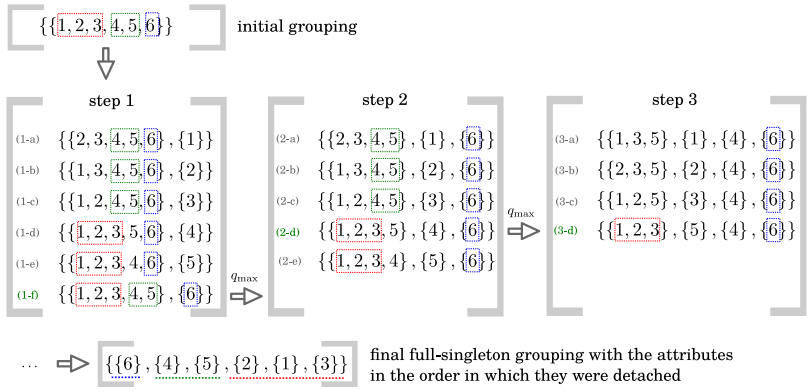



**Figure 4.4:** The sorting step of the ASTRID method

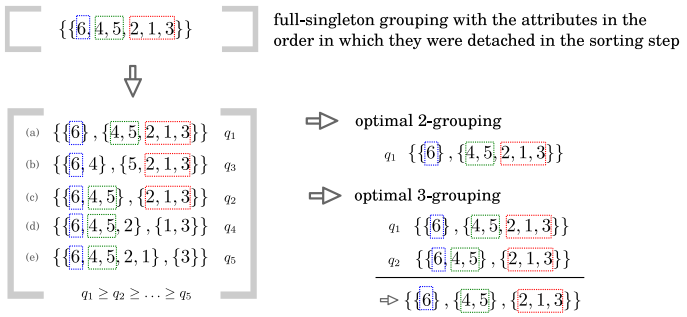



**Figure 4.5:** The grouping step of the ASTRID method

$m - 1$ candidate groupings and choose the grouping of highest cardinality for which $p_{\mathcal{S}} > \alpha$. We choose to find the grouping of maximum size such that the distribution in Equation (4.5) is fully factorised.

The complexity of the ASTRID method is quadratic in the sorting step and linear in the grouping step, with respect to the number of attributes.

**Theorem 4.3.** *The* ASTRID *method finds the exact solution to Problem 4.2 if the value function in Equation* (4.7) *is monotonic.*

*Proof.* The proof follows from the discussion of the sorting and grouping steps above. □

The proof is also presented in detail in Publication V.

### 4.8.2 The GoldenEye Algorithm

The GoldenEye algorithm implements the *identification* part of the framework. The goal is to find a grouping of attributes which corresponds to the

attribute interactions exploited by the classifier and we use the value function of Equation (4.9). The GoldenEye algorithm does not incorporate the cardinality $k$. A sensitivity parameter $\delta$ is instead used to adjust the granularity of the grouping. The GoldenEye algorithm finds an approximate solution to Problem 4.2.

Consider the example shown in Figure 4.6 demonstrating how the GoldenEye algorithm iteratively finds the optimal grouping of the dataset with the known structure $\{\{1,2,3\},\{4,5\},\{6\}\}$. We assume that the classifier has learned the correct structure of the dataset, i.e., that the classification accuracy is sufficiently high.

In the GoldenEye algorithm the grouping is based on comparing the current grouping against a baseline fidelity $\Delta$. The baseline fidelity is the fidelity for the all-singleton grouping, corresponding to the independence assumption of the naïve Bayes classifier, to which is added a small sensitivity parameter $\delta$ controlling the granularity of the grouping. In other words, $\Delta = \text{fid}\,(\textit{all-singleton grouping}) + \delta$. The grouping is visualised in Figure 4.6 and proceeds as follows.

The GoldenEye algorithm iteratively identifies the groups in the optimal solution $\mathcal{S}_O$ one at a time. The starting point in the GoldenEye algorithm is a full grouping $\mathcal{S}$ and during each iteration, one attribute at a time is greedily detached giving a new grouping $\mathcal{S}'$, maximising fidelity. This proceeds until the current fidelity drops below $\Delta$, indicating that some interactions used by the classifier were broken and a group has been found. The group consists of the last attribute that was detached and the attributes not yet detached. An example is given in Figure 4.6, where during the first iteration the group $\{1,2,3\}$ is found, by noticing in step 4 that fidelity drops below $\Delta$ when attribute 1 is detached, indicating that attribute 1 and attributes $\{2,3\}$ are jointly important and hence form a group.

The starting point for the next iteration is formed by placing the attributes identified so far as belonging to some group as singletons, while the attributes not yet considered are all in one group. This is exemplified in Figure 4.6, where the starting point for the second iteration is $\{\{4,5,6\},\{1\},\{2\},\{3\}\}$, i.e., the attributes in the group $\{1,2,3\}$ found during the first iteration are singletons.

After all groups have been found, an optional pruning of singletons can

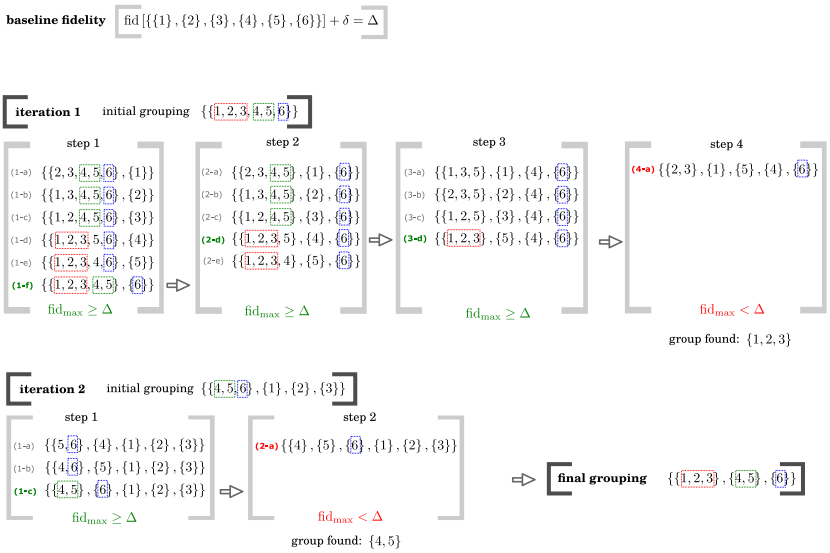



**Figure 4.6:** The grouping process of the GoldenEye algorithm. Attributes belonging to the same group are enclosed in like-coloured rectangles.

be performed by greedily permuting singletons at random (using scheme R1) until the fidelity again drops below $\Delta$. The computational complexity of the GoldenEye algorithm is quadratic with respect to the number of attributes in the dataset.

#### 4.8.2.1 The GoldenEye++ Algorithm

The GoldenEye++ algorithm extends the GoldenEye algorithm by introducing a new value function suitable for use in binary classification tasks with classifiers that output class probabilities. The fidelity goodness measure is susceptible to, e.g., class imbalance. In the GoldenEye++ value function interacting attributes are identified by considering the correlation between the class probabilities in the original dataset and a in a randomised version of the dataset. Given a classifier $f$, a grouping $\mathcal{S}$ and the original dataset $D = (X, C)$, let $\rho$ be a vector of class probabilities given by $f$ on $X$, and let $\rho^\star$ be a vector of class probabilities given by $f$ on $X^{\mathcal{S}}$. We now define the value function as

$$\widehat{T}_{\text{cor}}(\mathcal{S}) = \text{cor}\left(\rho, \rho^\star\right). \tag{4.10}$$

This value function detects more subtle changes in classifier performance than the fidelity-based value function used in GoldenEye and hence more fine-grained groupings can be found. The solution to Problem 4.2 is ap-

proximate.

### 4.8.3 Notes on Algorithms

The identification of interacting attributes can be performed either using the GoldenEye algorithm or using the ASTRID method.

Different value functions can be used to describe the quality of a given grouping $S$. However, the value function must be carefully chosen if it is used as the discriminating statistic in a significance testing framework. Fidelity, for instance, is not suitable for use in significance testing since it always decreases when the data is perturbed, as the comparison is made with respect to the predictions on unperturbed data.

The GoldenEye algorithm can also be used with an already trained classifier. As noted by Adler et al. (2016), this is useful for analysing a classifier that already exists and which can be accessed, but not retrained.

**Uniqueness and Stability of the Results** It should be noted that both the GoldenEye and ASTRID algorithms are exploratory tools for classifier analysis.

The results are not unique and are influenced by several factors such as the size of the data, the random variation in the data and the strength of the interactions in the dataset. It is also possible that multiple structures in the data can yield similar results, e.g., due to collinearity of attributes, in which case removal of one attribute from a group might not impact the value function. Different groupings can hence be found on different runs. The analysis of a dataset can be hence re-run several times using different random seeds in order to compare the results. This is analogous to clustering data using, e.g., the k-means algorithm where multiple runs can yield different cluster solutions.

As an example, the algorithms work well for the synthetic dataset from Publication V described above in Section 4.7.1 and for the dataset in Publication III, since the interactions are strong in these datasets. However, in real-world datasets the nature of the relations in the dataset are unknown. The methods presented above make it possible to explore how the interactions in the data are being exploited by classifiers.

Furthermore, it is essential that the used classifier is capable of learning the structure and that it has correctly learned the structure of the data such that a reasonable accuracy is reached. As an example, if already

the baseline performance of the classifier is very bad, then randomising the data might not impact classification performance. The naïve Bayes classifier provides an example of this: the classifier explicitly assumes that attributes are independent and is hence incapable of learning interactions between attributes. Practical examples using the naïve Bayes classifier are shown below in Section 4.9. Classification performance and the assumptions of the classifier should hence be considered when interpreting discovered attribute groupings. Due to this, the found groupings are not identical for different classifiers.

## 4.9 Examples

We conclude this chapter by presenting examples of how the `GoldenEye` and ASTRID algorithms are used in data analysis.

### 4.9.1 Attribute Interactions

**Datasets and Classifiers** We here consider two datasets from the UCI machine learning repository (Lichman, 2013). We here employ three classifiers: support vector machine (SVM) with an RBF kernel, random forest and naïve Bayes, see Section 2.2.2. The classifiers were used at their default settings. In the ASTRID method we used $N = 250$ samples to compute the value of $p_S$ in Equation (4.7) and $R = 250$ samples to evaluate the expectation in Equation (4.8). We used a significance level of $\alpha = 0.05$.

The number of samples used to calculate the expectation and the $p$-values in the ASTRID algorithm should be high enough to adequately capture the size of the investigated effect, and ensure the stability of the results with respect to random variation in the data. This can be investigated, e.g, by running the analysis several times using different random seeds. In practice it is also necessary to make a trade-off between the number of samples and computation time. Also, as mentioned in Section 2.3, the number of samples limits the minimum obtainable $p$-value, which using 250 samples is $\approx 0.004$.

Resampling is used in the `GoldenEye` algorithm to ensure that the fidelity can be calculated to a high enough precision. A minimum of 1000 samples is used and the standard deviation accuracy of fidelity is hence $\leq 1.58\%$.

The *Deterding Vowel Recognition* dataset (`vowel`) contains ten attributes extracted from speech signals of vowels uttered by different speakers. The task is to identify the correct vowel (11 classes).

The *Credit Approval* dataset (`credit-a`) contains 15 anonymised attributes with information relevant for credit card applications. The prediction task is binary, i.e., the task is to determine whether the credit application decision is positive or negative.

The results for the `credit-a` dataset using the `GoldenEye` algorithm are shown in Table 4.2 and using the ASTRID method in Table 4.4. The groupings found using `GoldenEye` and ASTRID indicate that there are few interactions in this dataset and that the attributes can be permuted without significantly affecting classification performance, for all classifiers. The results from the `GoldenEye` algorithm show that all classifiers use an all-singleton solution. Using the ASTRID method we find that the highest-cardinality grouping for the SVM classifier at a 5% significance level is the grouping with $k = 12$, where the classifier utilises a structure with one main attribute group, marked with $L$ in the table. For the SVM we also notice a drop in average prediction accuracy between the grouping with $k = 1$ and the grouping with $k = 15$. The random forest classifier appears to use no attribute interactions, and based on the $p$-value the highest-cardinality grouping is the all-singleton grouping with $k = 15$. Also, the average accuracy for different values of $k$ are also quite similar. For the naïve Bayes classifier it is clear that all groupings are equally good, which is consistent with the independence assumption of the classifier.

The results for the `vowel` dataset are shown in Table 4.3 and Table 4.5 for the `GoldenEye` and ASTRID algorithms, respectively. The `GoldenEye` groupings show that all classifiers utilise large attribute groups. This is further evidenced by the groupings for this dataset using the ASTRID algorithm, showing that the SVM and random forest classifiers both exploit attribute interactions and permuting the dataset leads to decreased performance. The highest-cardinality grouping for the SVM classifier is given by $k = 1$ and for random forest classifier by $k = 2$. For the naïve Bayes classifier all groupings are equally good.

**Table 4.2:** Grouping of the `credit-a` dataset using the `GoldenEye` algorithm. Column headers are as follows. Classification accuracy on unshuffled data ($acc_o$), classification accuracy on data permuted using the final grouping ($acc_f$), fidelity using an all-singleton grouping ($fid_b$) and the fidelity using the final grouping ($fid_f$). The other columns represent the attributes. Attributes marked with the same letter belong to the same group in the grouping. Attributes marked with small letters are pruned.

| | $acc_o$ | $acc_f$ | $fid_b$ | $fid_f$ | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| svm | 0.86 | 0.85 | 1.00 | 0.98 | a | b | c | d | e | f | g | h | I | j | k | l | m | n | o |
| randomForest | 0.88 | 0.84 | 0.96 | 0.95 | a | b | c | d | e | f | g | h | I | J | K | l | m | N | o |
| naiveBayes | 0.75 | 0.71 | 0.96 | 0.92 | a | b | c | d | e | f | g | H | I | j | K | l | m | n | O |

**Table 4.3:** Grouping of the `vowel` dataset using the `GoldenEye` algorithm. Column headers as in Table 4.2.

| | $acc_o$ | $acc_f$ | $fid_b$ | $fid_f$ | F0 | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| svm | 0.85 | 0.60 | 0.53 | 0.69 | A | A | A | B | B | B | C | C | C | C |
| randomForest | 0.91 | 0.68 | 0.64 | 0.73 | A | A | A | B | B | B | B | B | c | d |
| naiveBayes | 0.65 | 0.54 | 0.76 | 0.79 | A | A | A | B | C | D | E | F | G | h |

### 4.9.2 Applications of Groupings

The found groupings have several important practical applications, discussed in Publications III, IV and V.

**Interactions in Pharmacovigilance** In Publication IV the `GoldenEye++` algorithm is used in the field of pharmacovigilance to investigate interactions between different drugs. Detecting drug-drug-interactions is a difficult problem as the data is very sparse, i.e., adverse drug events are rare and not all patients potentially exhibit a similar reaction to the same combination of drugs. The `GoldenEye++` algorithm was here used to determine the interaction between the attributes (drugs) and the outcome (adverse drug event or not). Several interesting sets of interacting drugs were found.

**Anonymisation** The goal of data anonymisation (see, e.g., Agrawal and Srikant, 2000; Sweeney, 2002, and Bayardo and Srikant, 2003) is to create a dataset having the same properties as the original dataset, but where individual items in the original dataset cannot be identified. The two main methods of data anonymisation are data generalisation and randomisation (Article 29 Data Protection Working Party, 2014). The *k*-anonymity algorithm (Sweeney, 2002), for instance, uses generalisation. If randomisation is used to anonymise a dataset it is essential that the randomisation is performed without breaking important interactions in the data.

**(a)** SVM

| k | acc | $acc_{min}$ | $acc_{max}$ | sd | p | A14 | A7 | A13 | A1 | A6 | A12 | A15 | A9 | A8 | A2 | A3 | A5 | A4 | A10 | A11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.871 | | | | | (A | A | A | A | A | A | A | A | A | A | A | A | A | A | A) |
| 2 | 0.868 | 0.847 | 0.883 | 0.006 | 0.570 | (A | A | A) | (B | B | B | B | B | B | B | B | B | B | B | B) |
| 3 | 0.869 | 0.853 | 0.883 | 0.007 | 0.594 | (A | A | A) | (B) | (C | C | C | C | C | C | C | C | C | C | C) |
| 4 | 0.867 | 0.847 | 0.883 | 0.007 | 0.418 | (A | A | A) | (B) | (C) | (D | D | D | D | D | D | D | D | D | D) |
| 5 | 0.866 | 0.847 | 0.883 | 0.007 | 0.402 | (A | A) | (B) | (C) | (D) | (E | E | E | E | E | E | E | E | E | E) |
| 6 | 0.867 | 0.847 | 0.883 | 0.007 | 0.446 | (A | A) | (B) | (C) | (D) | (E) | (F | F | F | F | F | F | F | F | F) |
| 7 | 0.867 | 0.847 | 0.883 | 0.006 | 0.474 | (A) | (B) | (C) | (D) | (E) | (F) | (G | G | G | G | G | G | G | G | G) |
| 8 | 0.866 | 0.847 | 0.877 | 0.006 | 0.394 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H | H | H | H | H | H | H | H) |
| 9 | 0.861 | 0.834 | 0.877 | 0.009 | 0.251 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I | I | I | I | I | I | I) |
| 10 | 0.854 | 0.822 | 0.877 | 0.010 | 0.116 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I | I | I) | (J | J | J | J) |
| 11 | 0.850 | 0.822 | 0.877 | 0.011 | 0.064 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J | J) | (K | K | K | K) |
| 12 | 0.848 | 0.816 | 0.890 | 0.011 | 0.056 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L | L | L | L) |
| 13 | 0.846 | 0.816 | 0.871 | 0.009 | 0.008 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L | L) | (M | M) |
| 14 | 0.845 | 0.822 | 0.871 | 0.009 | 0.012 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M) | (N | N) |
| 15 | 0.847 | 0.816 | 0.877 | 0.011 | 0.020 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M) | (N) | (O) |

**(b)** Random forest

| k | acc | $acc_{min}$ | $acc_{max}$ | sd | p | A9 | A5 | A1 | A4 | A6 | A13 | A7 | A12 | A2 | A15 | A14 | A3 | A8 | A10 | A11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.865 | | | | | (A | A | A | A | A | A | A | A | A | A | A | A | A | A | A) |
| 2 | 0.862 | 0.828 | 0.902 | 0.011 | 0.510 | (A) | (B | B | B | B | B | B | B | B | B | B | B | B | B | B) |
| 3 | 0.860 | 0.828 | 0.890 | 0.010 | 0.442 | (A) | (B) | (C | C | C | C | C | C | C | C | C | C | C | C | C) |
| 4 | 0.859 | 0.828 | 0.896 | 0.011 | 0.414 | (A) | (B) | (C) | (D | D | D | D | D | D | D | D | D | D | D | D) |
| 5 | 0.857 | 0.828 | 0.883 | 0.011 | 0.347 | (A) | (B) | (C) | (D) | (E | E | E | E | E | E | E | E | E | E | E) |
| 6 | 0.860 | 0.828 | 0.902 | 0.012 | 0.438 | (A) | (B) | (C) | (D) | (E | E | E) | (F | F | F | F | F | F | F | F) |
| 7 | 0.858 | 0.828 | 0.896 | 0.012 | 0.390 | (A) | (B) | (C) | (D) | (E | E) | (F) | (G | G | G | G | G | G | G | G) |
| 8 | 0.859 | 0.828 | 0.890 | 0.012 | 0.390 | (A) | (B) | (C) | (D) | (E | E) | (F) | (G | G) | (H | H | H | H | H | H) |
| 9 | 0.858 | 0.822 | 0.896 | 0.012 | 0.378 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H | H) | (I | I | I | I | I | I) |
| 10 | 0.858 | 0.822 | 0.890 | 0.012 | 0.367 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J | J | J | J | J | J) |
| 11 | 0.863 | 0.834 | 0.890 | 0.011 | 0.514 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J | J) | (K | K | K | K) |
| 12 | 0.863 | 0.822 | 0.890 | 0.011 | 0.518 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L | L | L | L) |
| 13 | 0.865 | 0.828 | 0.896 | 0.011 | 0.641 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L | L | L) | (M) |
| 14 | 0.866 | 0.834 | 0.896 | 0.012 | 0.618 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M | M) | (N) |
| 15 | 0.863 | 0.834 | 0.902 | 0.011 | 0.534 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M) | (N) | (O) |

**(c)** naïve Bayes

| k | acc | $acc_{min}$ | $acc_{max}$ | sd | p | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.785 | | | | | (A | A | A | A | A | A | A | A | A | A | A | A | A | A | A) |
| 2 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B | B | B | B | B | B | B | B | B | B | B | B | B | B) |
| 3 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C | C | C | C | C | C | C | C | C | C | C | C | C) |
| 4 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D | D | D | D | D | D | D | D | D | D | D | D) |
| 5 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E | E | E | E | E | E | E | E | E | E | E) |
| 6 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F | F | F | F | F | F | F | F | F | F) |
| 7 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G | G | G | G | G | G | G | G | G) |
| 8 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H | H | H | H | H | H | H | H) |
| 9 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I | I | I | I | I | I | I) |
| 10 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J | J | J | J | J | J) |
| 11 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K | K | K | K | K) |
| 12 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L | L | L | L) |
| 13 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M | M | M) |
| 14 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M) | (N | N) |
| 15 | 0.785 | 0.785 | 0.785 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M) | (N) | (O) |

**Table 4.4:** Grouping of the `credit-a` using the ASTRID algorithm. The columns show the size of the grouping (k), the average, minimum and maximum accuracy for the grouping (acc, $acc_{min}$ and $acc_{max}$), the standard deviation of the accuracy and the $p$-value for the grouping calculated using Equation (4.7).

**(a)** SVM

| k | acc | $acc_{min}$ | $acc_{max}$ | sd | p | F1 | F4 | F3 | F9 | F6 | F7 | F0 | F8 | F2 | F5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.897 | | | | | (A | A | A | A | A | A | A | A | A | A) |
| 2 | 0.872 | 0.851 | 0.901 | 0.010 | 0.020 | (A) | (B | B | B | B | B | B | B | B | B) |
| 3 | 0.805 | 0.764 | 0.864 | 0.018 | 0.004 | (A) | (B | B | B | B | B | B | B | B) | (C) |
| 4 | 0.750 | 0.690 | 0.802 | 0.020 | 0.004 | (A) | (B) | (C | C | C | C | C | C | C) | (D) |
| 5 | 0.703 | 0.653 | 0.764 | 0.023 | 0.004 | (A) | (B) | (C | C | C | C | C | C) | (D) | (E) |
| 6 | 0.689 | 0.632 | 0.756 | 0.025 | 0.004 | (A) | (B) | (C) | (D | D | D | D | D) | (E) | (F) |
| 7 | 0.643 | 0.579 | 0.707 | 0.025 | 0.004 | (A) | (B) | (C) | (D | D | D | D) | (E) | (F) | (G) |
| 8 | 0.618 | 0.545 | 0.682 | 0.026 | 0.004 | (A) | (B) | (C) | (D) | (E | E | E) | (F) | (G) | (H) |
| 9 | 0.595 | 0.517 | 0.682 | 0.024 | 0.004 | (A) | (B) | (C) | (D) | (E | E) | (F) | (G) | (H) | (I) |
| 10 | 0.585 | 0.504 | 0.645 | 0.024 | 0.004 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) |

**(b)** Random forest

| k | acc | $acc_{min}$ | $acc_{max}$ | sd | p | F9 | F2 | F7 | F8 | F6 | F5 | F4 | F3 | F0 | F1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.884 | | | | | (A | A | A | A | A | A | A | A | A | A) |
| 2 | 0.882 | 0.860 | 0.901 | 0.007 | 0.458 | (A) | (B | B | B | B | B | B | B | B | B) |
| 3 | 0.862 | 0.839 | 0.884 | 0.010 | 0.008 | (A) | (B) | (C | C | C | C | C | C | C | C) |
| 4 | 0.837 | 0.806 | 0.872 | 0.012 | 0.004 | (A) | (B) | (C) | (D | D | D | D | D | D | D) |
| 5 | 0.812 | 0.773 | 0.851 | 0.013 | 0.004 | (A) | (B) | (C) | (D) | (E | E | E | E | E | E) |
| 6 | 0.720 | 0.665 | 0.781 | 0.019 | 0.004 | (A) | (B) | (C) | (D) | (E | E | E | E | E) | (F) |
| 7 | 0.687 | 0.632 | 0.744 | 0.018 | 0.004 | (A) | (B) | (C) | (D) | (E) | (F | F | F | F) | (G) |
| 8 | 0.656 | 0.599 | 0.711 | 0.019 | 0.004 | (A) | (B) | (C) | (D) | (E) | (F | F | F) | (G) | (H) |
| 9 | 0.637 | 0.599 | 0.686 | 0.017 | 0.004 | (A) | (B) | (C) | (D) | (E) | (F) | (G | G) | (H) | (I) |
| 10 | 0.626 | 0.574 | 0.678 | 0.018 | 0.004 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) |

**(c)** naïve Bayes

| k | acc | $acc_{min}$ | $acc_{max}$ | sd | p | F0 | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.640 | | | | | (A | A | A | A | A | A | A | A | A | A) |
| 2 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B | B | B | B | B | B | B | B | B) |
| 3 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C | C | C | C | C | C | C | C) |
| 4 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D | D | D | D | D | D | D) |
| 5 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E | E | E | E | E | E) |
| 6 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F | F | F | F | F) |
| 7 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G | G | G | G) |
| 8 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H | H | H) |
| 9 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I | I) |
| 10 | 0.640 | 0.640 | 0.640 | 0.000 | 1.000 | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) |

**Table 4.5:** Grouping of the `vowel` using the ASTRID algorithm. The columns show the size of the grouping (k), the average, minimum and maximum accuracy for the grouping (acc, $acc_{min}$ and $acc_{max}$), the standard deviation of the accuracy and the $p$-value for the grouping calculated using Equation (4.7).

The structure of the attribute interactions found using he ASTRID algorithm represent a factorisation of the class-conditional joint distribution of a dataset in terms of the grouping $\mathcal{S}$. Groupings $S$ for which $p_{\mathcal{S}} > 0.05$ in Equation (4.7) can be considered principled parametrisations for shuffling the data, since we at the 5% level do not reject the null hypothesis that the data originates from a factorised class-conditional data distribution as in Equation (4.5) where the factors are given by the grouping. This means that the structure of a dataset identified by the ASTRID algorithm can be used to anonymise a dataset such that attribute interactions utilised by a classifier are preserved. An example of this is provided in Publication V.

We next consider how attribute interactions can be used in data fusion, e.g., to enhance questionnaire data collection.

**Data Fusion** Knowledge of attribute interactions in a dataset has interesting practical implications for data fusion and efficient data collection.

Assume that the structure of a survey dataset with a relatively small number of items (survey respondents) and with a large number of attributes (survey questions) has been found using ASTRID. We denote the grouping of attributes in the dataset by $\mathcal{S} = \{S_1, \ldots, S_k\}$, i.e., the survey questions can be partitioned into smaller groups. The class label of each data item has been assigned from an external source, e.g., a clinical test.

If we now want to collect more training data for a new classifier, we can split up the original survey into smaller parts based on the groups in $\mathcal{S}$. Each respondent now only needs to answer a subset of questions, corresponding to a group $S \in \mathcal{S}$, and this set of answers is assigned a class label from the external source.

A training dataset can now be formed as follows. A new complete data item is obtained by fusing sets of answers to each group $S_1, \ldots, S_k$ from different respondents such that they all have the same externally assigned class label. The dataset obtained from data fusion can be used to train a new classifier, since the class-conditional joint data distribution of the fused dataset has the same factorised form as the initial dataset. Splitting up a survey dataset based on an initial grouping of attributes can make the data collection process faster and less expensive.

## 4.10 Chapter Summary

In this chapter we presented a framework for investigating attribute interactions. The framework consists of two parts. The *testing* part of the framework corresponds to a randomisation test for evaluating the hypothesis that a dataset originates from a particular factorised class-conditional joint distribution. The *identification* part of the framework corresponds to a generic clustering problem with the goal of partitioning the attributes of a dataset in terms of optimising a value function.

We showed how to automatically find an optimal grouping of attributes corresponding to a factorisation of the class-conditional joint data distribution and we also presented a generic method for finding the attribute

interactions exploited by a classifier in making predictions. We presented three algorithms implementing the concepts in the framework.

We also showed how the methods presented here can be used in data analysis and discussed interesting potential applications of the found interactions, e.g., for data anonymisation. The methods for exploring attribute interactions utilised by classifiers provide us with insight into the structure of the data and make the used machine learning model more interpretable in terms of revealing attribute interactions in the data and interactions exploited by the classifier.

# 5. Discussion

## 5.1 Conclusions

The central goal of data mining is to discover relations in data that are useful, e.g., in decision-making. One way of gaining insight into the data is to explore different structures. In this thesis we considered two topics: (i) investigating non-random structures in time series and finding explanations for these, and (ii) exploring attribute interactions utilised by classifiers. In the latter case we discussed the two related problems of exploring which attributes a classifier is jointly exploiting when making predictions, and finding a suitable factorisation for the class-conditional joint distribution of the attributes in a dataset.

In Chapter 3 we considered the problem of investigating the structure of time series. The principle of parsimony is important in modelling, which means that one should attempt to find the simplest explanations for the observed structures in the time series. Investigating structures in time series is formally accomplished using a statistical significance testing framework and we considered various methods for generating surrogate data for this purpose, e.g., constrained randomisation of the time series in the time and frequency domains. We illustrated different constrained randomisation schemes by analysing interval sequences and we also briefly considered the use of the bootstrap for estimating confidence intervals with applications to a sleep study.

In Chapter 4 we focused on the problem of exploring attribute interactions utilised by classifiers. Much of the work regarding classifiers has been on creating models with high predictive accuracy. Consequently, state-of-the-art classifiers are often complex models and interpretability is sacri-

ficed for performance. We presented the GoldenEye and GoldenEye++ algorithms for exploring which attributes a classifier is jointly using for making predictions. We also presented the ASTRID method which approaches the problem of investigating attribute interactions in datasets as a significance testing problem, using classifiers as a tool. Knowledge of attribute interactions has important practical applications. With the methods presented here we gain insight both into the data, and into the interaction between the classifier and the data when making predictions. We showed examples of how the methods presented here can be applied in data analysis.

## 5.2 Applicability in Time Series Analysis

Constrained randomisation methods are relatively easy to apply in the analysis of time series but there is, however, no general rule or method for formulating the null hypothesis. The formulation typically also requires domain knowledge, depending on the data. A clear benefit of the constrained randomisation methods in time series analysis is that the understanding of data can be built up gradually by adding constraints (see, e.g., Hanhijärvi et al., 2009b). The constrained randomisation methods can therefore be used both when exploring the data to find interesting patterns in a step-by-step fashion, and to test hypotheses regarding some particular features in the data. Already known or discovered patterns can be used as constraints in the generation of surrogate data. The constraints encode our current best knowledge of the structure and the effect of these structures are factored out from the surrogate datasets. This allows interesting patterns in the time series not explained by the current model to be discovered. This makes it relatively easy to verify different hypotheses regarding the data. In addition, constrained randomisations can also be used to investigate more complex null hypotheses, such as time varying structures that are difficult to capture using traditional parametric models. As an example, constrained randomisation has been used in the medical domain to investigate patterns of brain activity during epileptic seizures (see Kunhimangalam et al., 2008). Constrained randomisations hence constitute a powerful and flexible technique for exploratory data analysis.

A trend in many disciplines is the growth of the data volume available for analysis, due to an increased number of sensors being used and higher sampling rates, e.g., in the medical domain. New techniques and tools are

hence required for the analysis of, for instance, collections of time series. The randomisation-based methods discussed in this thesis are well suited for the exploratory analysis of collections of signals as complex relationships can be modelled in terms of different constraints. One interesting future aspect could be the development of randomisation-based data analysis schemes, e.g., for analysing multivariate biomedical time series in the medical domain where different relations between the time series could be modelled by constraints. As an example, Prichard and Theiler (1994) generated surrogate data using constrained randomisations preserving auto and cross correlations in an ensemble of time series.

### 5.3 Impact on Analysis of Classifiers

The main impact of this thesis with respect to the development of new methods is in the field of classifier analysis. The framework and algorithms presented in this thesis opened a new avenue for gaining insight into generic, black-box classifiers. The `GoldenEye` method presented in Publication III is the first method for analysing generic classifiers in terms of discovering groups of interacting attributes. For instance, in the previous work on investigating attribute interactions utilised by classifiers in the general case by Ojala and Garriga (2010), only the problem of whether or not a classifier utilises attribute interactions was addressed, but not the question of *what* the attribute interactions are.

The field of studying classifiers in terms of how they exploit attribute interactions hence represents a relatively young and emerging topic in data mining. However, this field is important from the perspective of enhancing the interpretability of predictive models and gaining insight into the structure of the data. In other words, by studying attribute interactions used by classifiers we learn something about both the algorithm and the data.

The ASTRID method is a continuation of the work on `GoldenEye`. Here we focus on interactions in the data and utilise classifiers to investigate the structure of datasets. We approach this as a significance testing problem, where we examine the hypothesis that a dataset has been sampled from a particular factorised class-conditional joint distribution. With this method we gain insight into the data in terms of the distribution generating the data and we thus learn about interactions in the data.

As an example of attribute interactions, assume that a domain expert in some field is working on a classification problem. Knowledge of the attribute interactions allows the expert to better understand what structure the classifier is using, which in turn makes it possible for the domain expert to check if the interaction structure used by the classifier is meaningful in terms of the expert's knowledge. Knowledge of the relationships between attributes in the data might reveal new interaction patterns to the expert, thus facilitating learning from the data.

However, users of classifiers are not always experts and issues such as parameter tuning is difficult (Fernández-Delgado et al., 2014). In these circumstances, it is very helpful to be able to peek into the black box to better understand how the algorithm is working. Methods allowing attribute interactions to be explored are hence important tools for machine learning practitioners.

Understanding the relation between attribute interactions is, however, only one aspect of being able to interpret the data. The causal background to the interactions, i.e., the reason for why attributes are interacting, must still be separately investigated. Furthermore, the exact nature of the interactions may not be easily interpretable.

An interesting and potential area of future research is in applying the methods discussed here in the important field of variable and feature selection. Since interpretable models should preferably be sparse, the GoldenEye and ASTRID algorithms could be valuable in the creation of sparse models, corresponding to feature selection and dataset sparsification, which is important in the analysis of datasets with a large number of attributes. Eliminating unnecessary features is important both for improving classification performance and also for potentially speeding up the classifier training process. The attribute interaction structures found using these methods could also be used as a basis for feature engineering, which could then be based on both domain knowledge and insight into the class-conditional joint data distribution.

The size of datasets has increased both in terms of volume and number of attributes (Guyon and Elisseeff, 2003). We can expect this trend to continue. Data mining methods are also now common in almost every domain. This means that it is important to develop efficient methods for investigating the structure of different types of data. Exploring attribute

interactions used by generic classifiers is hence a very important topic with many important real-world applications such as, e.g., data anonymisation and analysis of drug interactions in pharmacovigilance, as discussed here. The methods for analysing classifiers discussed in this thesis have a high generic applicability in the important field of supervised machine learning in problems where there is a need to understand, interpret and utilise attribute interactions. Better understanding of the models means that we do not need to make the trade-off between interpretability and high performance.

It can be expected that the research in this field will grow steadily in the next few years and the methods presented here might even become standard tools for practitioners in the analysis of classifiers due to their powerfulness, ease of applicability and interpretability.

# Bibliography


Adler, P., Falk, C., Friedler, S. A., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S. Auditing Black-box Models for Indirect Influence. In *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM)*, pages 1–10. IEEE, 2016.

Agrawal, R. and Srikant, R. Privacy-Preserving Data Mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD'00)*, pages 439–450. ACM, 2000.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1-2):5–43, 2003.

Andrzejak, R. G., Mormann, F., Kreuz, T., Rieke, C., Kraskov, A., Elger, C. E., and Lehnertz, K. Testing the null hypothesis of the nonexistence of a preseizure state. *Physical Review E*, 67(1):010901, 2003.

Article 29 Data Protection Working Party. Opinion 05/2014 on Anonymisation Techniques. Technical report, European Commission, 2014.

Bayardo, R. J. and Srikant, R. Technological Solutions for Protecting Privacy. *Computer*, 36(9):115–118, 2003.

Benjamini, Y. and Hochberg, Y. Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 57(1): 289–300, 1995.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

Breakspear, M., Brammer, M., and Robinson, P. A. Construction of multivariate surrogate sets from nonlinear data using the wavelet transform. *Physica D: Nonlinear Phenomena*, 182(1):1–22, 2003.

Breiman, L. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

Breiman, L. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

Bühlmann, P. Bootstraps for Time Series. *Statistical Science*, 17(1):52–72, 2002.

Bullmore, E., Long, C., Suckling, J., Fadili, J., Calvert, G., Zelaya, F., Carpenter, T. A., and Brammer, M. Colored Noise and Computational Inference in Neurophysiological (fMRI) Time Series Analysis: Resampling Methods in Time and Wavelet Domains. *Human Brain Mapping*, 12(2):61–78, 2001.

Burges, C. J. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Carlstein, E. Resampling Techniques for Stationary Time-Series: Some Recent Developments. 1990. University of North Carolina at Chapel Hill, Department of Statistics.

Carpenter, J. and Bithell, J. Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Statistics in Medicine*, 19(9):1141–1164, 2000.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. Intelligible Models for Healthcare: Predicting Pneumonia Risk and Hospital 30-day Readmission. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

Casella, G. and Berger, R. L. *Statistical Inference*. Duxbury, Pacific Grove, CA, USA, 2nd edition, 2001.

Chan, K.-P. and Fu, A. W.-C. Efficient Time Series Matching by Wavelets. In *Proceedings of the 15th IEEE International Conference on Data Engineering (ICDE)*, pages 126–133. IEEE, 1999.

Chua, E. C.-P., Tan, W.-Q., Yeo, S.-C., Lau, P., Lee, I., Mien, I. H., Puvanendran, K., and Gooley, J. J. Heart Rate Variability Can Be Used to Estimate Sleepiness-related Decrements in Psychomotor Vigilance during Total Sleep Deprivation. *Sleep*, 35(3):325–334F, 2012.

Cortes, C. and Vapnik, V. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

Cumming, G. and Finch, S. Inference by Eye: Confidence Intervals and How to Read Pictures of Data. *American Psychologist*, 60(2):170, 2005.

Datta, A., Sen, S., and Zick, Y. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *Proceedings of the 37th IEEE Symposium on Security and Privacy (SP)*, pages 598–617. IEEE, 2016.

Davison, A. C. and Hinkley, D. V. *Bootstrap Methods and Their Applications*. Cambridge University Press, 1st edition, 1997.

Davison, A. C., Hinkley, D. V., and Young, G. A. Recent Developments in Bootstrap Methodology. *Statistical Science*, 18(2):141–157, 2003.

De Bie, T. An Information Theoretic Framework for Data Mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 564–572. ACM, 2011.

Dinges, D. F. and Powell, J. W. Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations. *Behavior Research Methods, Instruments, & Computers*, 17(6):652–655, 1985.

Domingos, P. and Pazzani, M. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, 29(2):103–130, 1997.

Duivesteijn, W. and Thaele, J. Understanding Where Your Classifier Does (Not) Work–The SCaPE Model Class for EMM. In *Proceedings of the 14th IEEE International Conference on Data Mining (ICDM)*, pages 809–814. IEEE, 2014.

Efron, B. Bootstrap Methods: Another Look at the Jackknife. *The annals of Statistics*, 7(1):1–26, 1979.

Efron, B. Better Bootstrap Confidence Intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987.

Efron, B. and Tibshirani, R. Bootstrap Methods for Standard Errors, Confidence Intervals, and Other measures of Statistical Accuracy. *Statistical Science*, 1(1):54–75, 1986.

Efron, B. and Tibshirani, R. J. *An Introduction to the Bootstrap*. CRC Press, 1st edition, 1994.

Faes, L., Zhao, H., Chon, K. H., and Nollo, G. Time-Varying Surrogate Data to Assess Nonlinearity in Nonstationary Time Series: Application to Heart Rate Variability. *IEEE Transactions on Biomedical Engineering*, 56(3):685–695, 2009.

Faloutsos, C., Jagadish, H., Mendelzon, A. O., and Milo, T. A Signature Technique for Similarity-Based Queries. In *Proceedings of the Compression and Complexity of Sequences*, pages 2–20. IEEE, 1997.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3):37, 1996.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

Freitas, A. A. Understanding the Crucial Role of Attribute Interaction in Data Mining. *Artificial Intelligence Review*, 16(3):177–199, 2001.

Garde, S., Regalado, M. G., Schechtman, V. L., and Khoo, M. C. Nonlinear dynamics of heart rate variability in cocaine-exposed neonates during sleep. *American Journal of Physiology-Heart and Circulatory Physiology*, 280(6):H2920–H2928, 2001.

Geyer, C. Markov Chain Monte Carlo Maximum Likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*. American Statistical Association, 1991.

Geyer, C. Introduction to Markov Chain Monte Carlo. In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

Gionis, A., Mannila, H., and Terzi, E. Clustered segmentations. In *3rd SIGKDD Workshop on Mining Temporal and Sequential Data (KDD/TDM)*, 2004.

Gionis, A., Mannila, H., Mielikäinen, T., and Tsaparas, P. Assessing Data Mining Results via Swap Randomization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(3):14, 2007.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101 (23):e215–e220, 2000.

Goldsmith, R. Congestive Heart Failure RR Interval Database, 2003. URL `http://dx.doi.org/10.13026/C2F598`.

Good, P. I. Extensions Of The Concept Of Exchangeability And Their Applications. *Journal of Modern Applied Statistical Methods*, 1(2): 243–247, 2002.

Good, P. I. *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Springer, 3rd edition, 2005.

Good, P. I. *Resampling Methods: A Practical Guide to Data Analysis*. Birkhäuser, 3rd edition, 2006.

Goodman, B. and Flaxman, S. EU regulations on algorithmic decision-making and a "right to explanation". In *ICML Workshop on Human Interpretability in Machine Learning*, 2016. URL `http://arxiv.org/abs/1606.08813`.

Guyon, I. and Elisseeff, A. An Introduction to Variable and Feature Selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

Hanhijärvi, S., Garriga, G. C., and Puolamäki, K. Randomization Techniques for Graphs. In *Proceedings of the 9th SIAM International Conference on Data Mining (SDM)*, volume 928, pages 780–791. SIAM, 2009a.

Hanhijärvi, S., Ojala, M., Vuokko, N., Puolamäki, K., Tatti, N., and Mannila, H. Tell Me Something I Don't Know: Randomization Strategies for Iterative Data Mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 379–388. ACM, 2009b.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2nd edition, 2009.

Henelius, A., Korpela, J., and Puolamäki, K. Explaining Interval Sequences by Randomization. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2013)*, pages 337–352. Springer, 2013.

Henelius, A., Puolamäki, K., Boström, H., Asker, L., and Papapetrou, P. A peek into the black box: Exploring classifiers by randomization. *Data Mining and Knowledge Discovery*, 28(5–6):1503–1529, 2014a.

Henelius, A., Sallinen, M., Huotilainen, M., Müller, K., Virkkala, J., and Puolamäki, K. Heart Rate Variability for Evaluating Vigilant Attention in Partial Chronic Sleep Restriction. *Sleep*, 37(7):1257–1267, 2014b.

Henelius, A., Puolamäki, K., Karlsson, I., Zhao, J., Asker, L., Boström, H., and Papapetrou, P. GoldenEye++: A Closer Look into the Black Box. In *Proceedings of the Third International Symposium on Statistical Learning and Data Sciences (SLDS 2015)*, pages 96–105. Springer, 2015.

Henelius, A., Puolamäki, K., and Ukkonen, A. Finding Statistically Significant Attribute Interactions. *Submitted to ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2017), arXiv e-prints, arXiv:1612.07597*, 2017.

Holm, S. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Huang, G.-B., Zhou, H., Ding, X., and Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.

Jakulin, A. and Bratko, I. Analyzing Attribute Dependencies. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 229–240. Springer, 2003.

Jakulin, A. and Bratko, I. Testing the Significance of Attribute Interactions. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 52–59. ACM, 2004.

Jakulin, A., Bratko, I., Smrke, D., Demsar, J., and Zupan, B. Attribute Interactions in Medical Data Analysis. In *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe (AIME)*, pages 229–238. Springer, 2003.

Jalali, A. and Pfeifer, N. Interpretable per case weighted ensemble method for cancer associations. *BMC Genomics*, 17(1):501, 2016.

Jentsch, C. and Kreiss, J.-P. The multiple hybrid bootstrap – Resampling multivariate linear processes. *Journal of Multivariate Analysis*, 101(10): 2320–2345, 2010.

Kalliovirta, L., Meitz, M., and Saikkonen, P. A gaussian mixture autoregressive model for univariate time series. *Journal of Time Series Analysis*, 36(2):247–266, 2015.

Keylock, C. Constrained surrogate time series with preservation of the mean and variance structure. *Physical Review E*, 73(3):036707, 2006.

Keylock, C. J. A wavelet-based method for surrogate data generation. *Physica D: Nonlinear Phenomena*, 225(2):219–228, 2007.

Kim, K.-j. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1–2):307–319, 2003.

Kleinberg, J., Papadimitriou, C., and Raghavan, P. A Microeconomic View of Data Mining. *Data Mining and Knowledge Discovery*, 2(4):311–324, 1998.

Korpela, J., Puolamäki, K., and Gionis, A. Confidence bands for time series data. *Data Mining and Knowledge Discovery*, 28(5-6):1530–1553, 2014.

Koski, T. J. and Noble, J. M. A Review of Bayesian Networks and Structure Learning. *Annales Societatis Mathematicae Polonae. Series 3: Mathematica Applicanda*, 40(1):53–103, 2012.

Krause, J., Perer, A., and Ng, K. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *Proceedings of the 2016 ACM CHI Conference on Human Factors in Computing Systems - CHI'16*, pages 5686–5697. ACM, 2016.

Kreiss, J.-P. and Lahiri, S. Bootstrap Methods for Time Series. In *Handbook of Statistics Volume 30: Time Series Analysis: Methods and Applications*, pages 3–26. Elsevier, 2012.

Kugiumtzis, D. Test your surrogate data before you test for nonlinearity. *Physical Review E*, 60(3):2808–2816, 1999.

Kunhimangalam, R., Joseph, P. K., and Sujith, O. Nonlinear analysis of EEG signals: Surrogate data analysis. *IRBM*, 29(4):239–244, 2008.

Kunsch, H. R. The Jackknife and the Bootstrap for General Stationary Observations. *The Annals of Statistics*, 17(3):1217–1241, 1989.

Laird, A. R., Rogers, B. P., and Meyerand, M. E. Comparison of Fourier and Wavelet Resampling Methods. *Magnetic Resonance in Medicine*, 51 (2):418–422, 2004.

Lehman, L.-w. H., Adams, R. P., Mayaud, L., Moody, G. B., Malhotra, A., Mark, R. G., and Nemati, S. A Physiological Time Series Dynamics-Based Approach to Patient Monitoring and Outcome Prediction. *IEEE Journal of Biomedical and Health Informatics*, 19(3):1068–1076, 2015.

Li, C., Ding, G.-H., Wu, G.-Q., and Poon, C.-S. Band-Phase-Randomized Surrogate Data Reveal High-Frequency Chaos in Heart Rate Variability. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2806–2809. IEEE, 2010.

Lichman, M. UCI Machine Learning Repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Lijffijt, J., Papapetrou, P., and Puolamäki, K. A statistical significance testing approach to mining the most informative set of patterns. *Data Mining and Knowledge Discovery*, 28(1):238–263, 2014.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.

Lipton, Z. C. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability in Machine Learning*, 2016. URL `https://arxiv.org/abs/1606.03490`.

Liu, J. S. *Monte Carlo Strategies in Scientific Computing*. Springer, 1st edition, 2008.

Mallat, S. *A Wavelet Tour of Signal Processing*. Academic Press, 3rd edition, 2009.

Mampaey, M. and Vreeken, J. Summarizing Categorical Data by Clustering Attributes. *Data Mining and Knowledge Discovery*, 26(1):130–173, 2013.

Michaels, A. F., Knap, A. H., Dow, R. L., Gundersen, K., Johnson, R. J., Sorensen, J., Close, A., Knauer, G. A., Lohrenz, S. E., Asper, V. A., et al. Seasonal patterns of ocean biogeochemistry at the US JGOFS Bermuda Atlantic time-series study site. *Deep Sea Research Part I: Oceanographic Research Papers*, 41(7):1013–1038, 1994.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT press, 2012.

Murdoch, D. J., Tsai, Y.-L., and Adcock, J. P-Values are Random Variables. *The American Statistician*, 62(3):242–245, 2008.

Ojala, M. and Garriga, G. C. Permutation Tests for Studying Classifier Performance. *The Journal of Machine Learning Research*, 11:1833–1863, 2010.

Ojala, M., Vuokko, N., Kallio, A., Haiminen, N., and Mannila, H. Randomization of real-valued matrices for assessing the significance of data mining results. In *Proceedings of the 8th SIAM International Conference on Data Mining (SDM)*, volume 8, pages 494–505. SIAM, 2008.

Payton, M. E., Greenstone, M. H., and Schenker, N. Overlapping confidence intervals or standard error intervals: What do they mean in terms of statistical significance? *Journal of Insect Science*, 3(34):1–6, 2003.

Politis, D. N. The Impact of Bootstrap Methods on Time Series Analysis. *Statistical Science*, 18(2):219–230, 2003.

Politis, D. N. and Romano, J. P. A Circular Block-Resampling Procedure for Stationary Data (Technical Report No. 370). 1991. Stanford University, Department of Statistics.

Porta, A., Guzzetti, S., Furlan, R., Gnecchi-Ruscone, T., Montano, N., and Malliani, A. Complexity and Nonlinearity in Short-Term Heart Period Variability: Comparison of Methods Based on Local Nonlinear Prediction. *IEEE Transactions on Biomedical Engineering*, 54(1):94–106, 2007.

Prichard, D. and Theiler, J. Generating Surrogate Data for Time Series with Several Simultaneously Measured Variables. *Physical Review Letters*, 73(7):951–954, 1994.

Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., 1993.

Raccuglia, P., Elbert, K. C., Adler, P. D., Falk, C., Wenny, M. B., Mollo, A., Zeller, M., Friedler, S. A., Schrier, J., and Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73–77, 2016.

Rakthanmanon, T., Keogh, E. J., Lonardi, S., and Evans, S. MDL-based time series clustering. *Knowledge and information systems*, 33(2):371–399, 2012.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144. ACM, 2016.

Scargle, J. D. Astronomical Time Series Analysis: New Methods for Studying Periodic and Aperiodic Systems. In *Astronomical Time Series: Proceedings of The Florence and George Wise Observatory 25th Anniversary Symposium*, volume 218, pages 1–12, 1997.

Schenker, N. and Gentleman, J. F. On Judging the Significance of Differences by Examining the Overlap Between Confidence Intervals. *The American Statistician*, 55(3):182–186, 2001.

Schreiber, T. Constrained Randomization of Time Series Data. *Physical Review Letters*, 80(10):2105–2108, 1998.

Schreiber, T. and Schmitz, A. Improved surrogate data for nonlinearity tests. *Physical Review Letters*, 77(4):635–638, 1996.

Schreiber, T. and Schmitz, A. Surrogate time series. *Physica D: Nonlinear Phenomena*, 142(3):346–382, 2000.

Sweeney, L. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5): 557–570, 2002.

Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology. Heart Rate Variability. Standards of Measurement, Physiological Interpretation, and Clinical Use. *Circulation*, 93(5):1043–1065, 1996.

Tatti, N. Are your Items in Order? In *Proceedings of the 11th SIAM International Conference on Data Mining (SDM)*, pages 414–425. SIAM, 2011.

Theiler, J. and Prichard, D. Constrained-realization Monte-Carlo method for hypothesis testing. *Physica D: Nonlinear Phenomena*, 94(4):221–235, 1996.

Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., and Farmer, J. D. Testing for nonlinearity in time series: the method of surrogate data. *Physica D: Nonlinear Phenomena*, 58(1–4):77–94, 1992.

Turner, R. A model explanation system. In *26th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.

Ustun, B., Tracà, S., and Rudin, C. Supersparse Linear Integer Models for Interpretable Classification. *arXiv preprint arXiv:1306.6677*, 2014.

Vuokko, N. and Kaski, P. Significance of Patterns in Time Series Collections. In *Proceedings of the 11th SIAM International Conference on Data Mining (SDM)*, pages 676–686. SIAM, 2011.

Westfall, P. H. and Young, S. S. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. John Wiley & Sons, 1st edition, 1993.

Xu, X., Schuckers, S., Group, C. S., et al. Automatic detection of artifacts in heart period data. *Journal of Electrocardiology*, 34(4):205–210, 2001.

Ying, X. and Wu, X. Graph Generation with Prescribed Feature Constraints. In *Proceedings of the 9th SIAM International Conference on Data Mining (SDM)*, volume 9, pages 966–977. SIAM, 2009.

Zhao, Z. and Liu, H. Searching for Interacting Features. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI)*, pages 1156–1161, 2007.

Zhao, Z. and Liu, H. Searching for interacting features in subset selection. *Intelligent Data Analysis*, 13(2):207–228, 2009.

Aalto-DD 58/2017

ISBN 978-952-60-7361-3 (printed)
ISBN 978-952-60-7360-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

**Aalto University**
**School of Science**
**Department of Computer Science**
**www.aalto.fi**

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

**DOCTORAL**
**DISSERTATIONS**