Permutation Distribution Clustering and Structural Equation Model Trees

Dissertation zur Erlangung des Grades des
Doktors der Naturwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten der
Universität des Saarlandes

von
Andreas Markus Brandmaier

Saarbrücken
2011

Tag des Kolloquiums: 21.12.2011

Dekan: Prof. Dr. Holger Hermanns

Berichterstatter:	Prof. Dr. Antonio Krüger
	Prof. Dr. Timo von Oertzen
	Prof. Dr. Steven M. Boker
Vorsitz	Prof. Dr. Thorsten Herfet
Akad. Mitarbeiter	Dr. Ralf Jung

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Berlin, 05.01.2012 _____

Abstract

The primary goal of this thesis is to present novel methodologies for the exploratory analysis of psychological data sets that support researchers in informed theory development. Psychological data analysis bears a long tradition of confirming hypotheses generated prior to data collection. However, in practical research, the following two situations are commonly observed: In the first instance, there are no initial hypotheses about the data. In that case, there is no model available and one has to resort to uninformed methods to reveal structure in the data. In the second instance, existing models that reflect prior hypotheses need to be extended and improved, thereby altering and renewing hypotheses about the data and refining descriptions of the observed phenomena. This dissertation introduces a novel method for the exploratory analysis of psychological data sets for each of the two situations. Both methods focus on time series analysis, which is particularly interesting for the analysis of psychophysiological data and longitudinal data typically collected by developmental psychologists. Nonetheless, the methods are generally applicable and useful for other fields that analyze time series data, e.g., sociology, economics, neuroscience, and genetics.

The first part of the dissertation proposes a clustering method for time series. A dissimilarity measure of time series based on the permutation distribution is developed. Employing this measure in a hierarchical scheme allows for a novel clustering method for time series based on their relative complexity: Permutation Distribution Clustering (PDC). Two methods for the determination of the number of distinct clusters are discussed based on a statistical and an information-theoretic criterion.

Structural Equation Models (SEMs) constitute a versatile modeling technique, which is frequently employed in psychological research. The second part of the dissertation introduces an extension of SEMs to Structural Equation Modeling Trees (SEM Trees). SEM Trees describe partitions of a covariate-space which explain differences in the model parameters. They can provide solutions in situations in which hypotheses in the form of a model exist but may potentially be refined by integrating other variables. By harnessing the full power of SEM, they represent a general data analysis technique that can be used for both time series and non-time series data. SEM Trees algorithmically refine initial models of the sample and thus support researchers in theory development.

This thesis includes demonstrations of the methods on simulated as well as on real data sets, including applications of SEM Trees to longitudinal models of cognitive development and cross-sectional cognitive factor models, and applications of PDC on psychophysiological data, including electroencephalographic, electrocardiographic, and genetic data.

Zusammenfassung

Ziel dieser Arbeit ist der Entwurf von explorativen Analysemethoden für Datensätze aus der Psychologie, um Wissenschaftler bei der Entwicklung fundierter Theorien zu unterstützen. Die Arbeit ist motiviert durch die Beobachtung, dass die klassischen Auswertungsmethoden für psychologische Datensätze auf der Tradition gründen, Hypothesen zu testen, die vor der Datenerhebung aufgestellt wurden. Allerdings treten die folgenden beiden Situationen im Alltag der Datenauswertung häufig auf: (1) es existieren keine Hypothesen über die Daten und damit auch kein Modelle. Der Wissenschaftler muss also auf uninformierte Methoden zurückgreifen, um Strukturen und Ähnlichkeiten in den Daten aufzudecken. (2) Modelle sind vorhanden, die Hypothesen über die Daten widerspiegeln, aber die Stichprobe nur unzureichend abbilden. In diesen Fällen müssen die existierenden Modelle und damit Hypothesen verändert und erweitert werden, um die Beschreibung der beobachteten Phänomene zu verfeinern. Die vorliegende Dissertation führt für beide Fälle je eine neue Methode ein, die auf die explorative Analyse psychologischer Daten zugeschnitten ist. Gleichwohl sind beide Methoden für alle Bereiche nützlich, in denen Zeitreihendaten analysiert werden, wie z.B. in der Soziologie, den Wirtschaftswissenschaften, den Neurowissenschaften und der Genetik.

Der erste Teil der Arbeit schlägt ein Clusteringverfahren für Zeitreihen vor. Dieses basiert auf einem Ähnlichkeitsmaß zwischen Zeitreihen, das auf die Permutationsverteilung der eingebetteten Zeitreihen zurückgeht. Dieses Maß wird mit einem hierarchischen Clusteralgorithmus kombiniert, um Zeitreihen nach ihrer Komplexität in homogene Gruppen zu ordnen. Auf diese Weise entsteht die neue Methode der Permutationsverteilungs-basierten Clusteranalyse (PDC). Zwei Methoden zur Bestimmung der Anzahl von separaten Clustern werden hergeleitet, einmal auf Grundlage von statistischen Tests und einmal basierend auf informationstheoretischen Kriterien.

Der zweite Teil der Arbeit erweitert Strukturgleichungsmodelle (SEM), eine vielseitige Modellierungstechnik, die in der Psychologie weit verbreitet ist, zu Strukturgleichungsmodell-Bäumen (SEM Trees). SEM Trees beschreiben rekursive Partitionen eines Raumes beobachteter Variablen mit maximalen Unterschieden in den Modellparametern eines SEMs. In Situationen, in denen Hypothesen in Form eines Modells existieren, können SEM Trees sie verfeinern, indem sie automatisch Variablen finden, die Unterschiede in den Modellparametern erklären. Durch die hohe Flexibilität von SEMs, können eine Vielzahl verschiedener Modelle mit SEM Trees erweitert werden. Die Methode eignet sich damit für die Analyse sowohl von Zeitreihen als auch von Nicht-Zeitreihen. SEM Trees verfeinern algorithmisch anfängliche Hypothesen und unterstützen Forscher in der Weiterentwicklung ihrer Theorien.

Die vorliegende Arbeit beinhaltet Demonstrationen der vorgeschlagenen Methoden auf realen Datensätzen, darunter Anwendungen von SEM Trees auf einem längsschnittlichen Wachstumsmodell kognitiver Fähigkeiten und einem querschnittlichen kognitiven Faktor Modell, sowie Anwendungen des PDC auf verschiedenen psychophsyiologischen Zeitreihen.

Acknowledgements

This thesis would not have been possible unless Timo von Oertzen, head of the Formal Methods project, had convinced me in the first place that the Center for Lifespan Psychology at the Max Planck Institute for Human Development is a great place to work and that it offers the possibility to discover multidisciplinary challenges that are waiting to be tackled with creative ideas. His constant personal encouragement and scientific enthusiasm helped me to find the long way through a dense forest of trees.

I am most grateful to Ulman Lindenberger, who supported me and my ideas from the first day on. It has been a great honor and pleasure to work with him and all the great colleagues in the Center for Lifespan Psychology.

I am deeply grateful to Antonio Krüger for supporting and encouraging me in this challenging endeavor to jointly answer psychological and computer scientific questions in a multidisciplinary approach.

I am most thankful to Steven Boker for supporting me and believing in my ideas. I hope we keep being synchronized.

I am deeply indebted to John J. McArdle for never ceasing to inspire me and being a shining example in his ardent enthusiasm to make good things happen! In addition, Jack kindly provided the WAIS-R and WISC data sets that are presented in the applications chapter.

Indeed, I am indebted to many of my colleagues for supporting me. Especially, I'd like to express my gratitude to the following people:

Michael Schellenbach. Some say that scientist turn coffee into ideas. Michael has provided me constantly with company, encouragement, and coffee, and I am most thankful for that. Also, I want to thank him for helping me to collect the walking speed data set.

Markus Werkle-Bergner, who has constantly supported and encouraged me by showing interest in new methods to analyze data.

Manuel Völkle, who has been supportive all the time, and never ceased to show deep interest in methodological musings.

Julia Delius, for her time and her valuable insights into the deep matters of orthography, punctuation, and understandability. On a related topic: She was the only one who could spot the difference between an oyster and an ostrich.

Cornelia Wrzus, for her constant encouragement, and support, and for letting me win occasional table tennis games to cheer me up. Certainly, those wins could not have been due to my skills.

Julius Verrel and Thomas Grandy for inspiring and elated jazz sessions. Remember: *Pa*-na-ma *Pa*-na-ma *Cu*-ba (pst) *Bo*ssa *No*va (pst)

Florian Schmiedek, for his belief in exploratory methods, and for kindly providing the EEG data set.

Goran Papenberg, for inspiring discussions and for pointing me to Elfriede Jelinek's books.

Anna Kleinspehn-Ammerlahn, for providing a great data set of interpersonal action synchronization, which unfortunately did not make it into this thesis.

Tim Brick, for many fruitful and fun discussions and the awkward self-insights about how I behave in conversations.

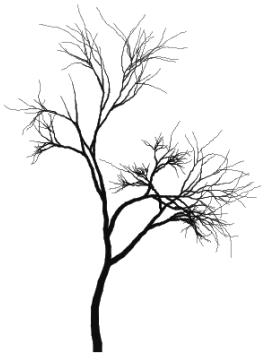
The student assistants of the Formal Methods project, with whom I very much enjoyed working: Julian Karch, Michael Beckmann, Florian Schilling, Denis Tapyshpan and Matthias Guggenmos.

The remaining guys from the computer science research circle, the primordial ooze of our ideas: Uwe Czienskowski, Paul Czienskowski, Dominik Jednoralski, Sebastian Schröder, Berndt Wischniewski, and the LIP support group.

My deep gratitude goes to all my friends in Berlin and around the world, who put up with me making myself scarce over the final period of my dissertation time. I will be back.

I would like to thank my family, who have always supported me in my endeavors and will probably stop reading this thesis about here. I am sorry, but believe me, the content is really cool!

I do not know how I can ever properly thank Myriam Sander: Your encouragement and support helped me in the successful completion of this work. You bore all my joy, my doubts, my enlightenments, and despair. If it wasn't for you, I would not have had a life beyond this thesis.



Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Motivation

In psychological research, enormous amounts of data are collected every day. Building and developing informed theories from multivariate and multi-method data sets is a major challenge in today's research. Instruments for data collection span a vast range of technical equipment and testing tools. Data are collected in the behavioral domain, obtained from participants completing tests or filling out questionnaires, in the physiological domain, obtained for example by measuring electrical currents on the human scalp with electroencephalography or blood-level oxygenation from high resolution brain imaging, or, increasingly, in the genetic domain, which can easily reveal information about 800,000 markers in each person's DNA. In developmental psychology, data sets are typically not only assessed by observing a single point in time, but participants are preferred to be followed longitudinally, which further enlarges the data sets. An outstanding example represents the multidisciplinary Berlin Aging Study (Lindenberger, Smith, Mayer, & Baltes, 2010) that followed 516 people over the course of more than 20 years, assessing facets of aging with instruments and measures from psychology, psychiatry, sociology, medicine and socioeconomics. A second prominent example is the COGITO study that assessed cognitive measures of about 100 young and 100 old adults across 100 daily training sessions (Lindenberger, Li, Lövdén, & Schmiedek, 2007), which represents an unprecedented dense longitudinal measurement of cognitive development on multiple time scales. The sheer number of variables collected in a single study can be overwhelming, and the number of possible interactions and mutual influences are enormous. In the presence of these unruly masses of data, one might feel the same bafflement as the character James from the novel "Moneyball: The Art of Winning an Unfair Game" by Lewis (2004), who reflects about baseball statistics and wonders "if we haven't become so numbed by all these numbers that we are no longer capable of truly assimilating any knowledge which might result from them" (p. 234). Indeed, it seems impossible to make sense of these data sets with the tool box that is taught as the major part of "statistics for psychology" text books, since it is based on confirming prior hypotheses

about the data. These methods are usually built around the general linear model and include the well-known analysis of variances, linear regression, correlations, and the t-test.

As an offshoot of the computer sciences, the field of data mining has been gaining interest in recent years. Fayyad, Piatetsky-Shapiro, and Smyth (1996b) defined the aim of this field the development of "computational theories and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data" (p. 38). Particularly the field of neurosciences has not hesitated to employ data mining methods, most prominently in the classification of brain responses (Formisano, De Martino, & Valente, 2008), for example to facilitate "mind reading" by decoding mental states in distributed representation patterns from functional magnet resonance imaging (fMRI) (Haynes & Rees, 2006). With the advent of large-scale computational resources and sophisticated algorithms, an automatic search for patterns in data has become possible and has been described with a variety of terms, including data mining, knowledge extraction, information discovery and knowledge discovery. Apart from the prominent examples from neurocognitive studies, data mining methods can be helpful in all subfields of psychology. Importantly, in order to facilitate true knowledge gain, the employed method must remain accessible and the results interpretable in such a way that the algorithm's inferences can be understood by the researcher (Fayyad et al., 1996b). In the context of data mining in the health sciences, Shillabeer and Roddick (2006) argued that data mining techniques can only be successful if they are adapted to the domain of interest. With an eye on the data analysis process in psychology, this thesis outlines two new methods to construct, extend and refine exploratory hypotheses about the data, in addition to confirming prior hypotheses, and thereby support the researcher in informed theory building.

1.2 Rationale

In everyday research situations, investigators are faced with large corpora of data and are required to make sense of them. This thesis focuses on two particular situations that are commonly encountered by researchers: (1) situations in which there is no model about the data available and (2) situations in which a model is available but needs to be extended by additional variables that have not yet been integrated into it. This thesis provides solutions for both situations with a focus on knowledge representation by hierarchical tree structures which are created by applying statistical and information-theoretic tests to a novel complexity-based clustering algorithm and a new type of model-based decision-tree.

Whenever a set of time series is available and no prior information or models are available, clustering techniques can help to reveal structure in the data set. The first proposed method, Permutation Distribution Clustering (PDC), is a clustering algorithm for time series. It is based on a dissimilarity concept relying on the complexity of time series. It allows an exploratory analysis of multidimensional time series of differing lengths and can, in addition, be used as a method for time series segmentation. Particularly, I propose the use of an α-divergence (Amari & Nagaoka, 2007; Chernoff, 1952) between the Permutation Distribution (PD) of time series as a dissimilarity measure. This distribution was introduced by Bandt and Pompe (2002), who interpret the entropy of this distribution as a measure of complexity of an univariate time series. Using a divergence based on this distribution leads to a measure of dissimilarity based on the relative complexity of time series. The α-divergence is a generalization of a set of common dissimilarity measures, particularly, the important information-theoretically

motivated Kullback-Leibler divergence (Kullback & Leibler, 1951). Using the α-divergence between PDs has several advantages: (1) the distance measure can be calculated linearly in the length of the time series, (2) it is invariant to monotonic transformations, particularly scaling and shifting, of the time series, and (3) it can be related to the Bayes-optimal error rate of discriminating between two classes. Combining the permutation distribution, the α-divergence, and a hierarchical agglomerative clustering scheme leads to a new and important clustering method for time series.

The resulting algorithm requires the choice of a parameter that is crucial to obtain distinctive patterns for clustering. This parameter defines the embedding dimension of the time series that determines the size of the internal representation. In the literature, this has not been solved, and authors merely report ranges of appropriate embedding dimensions without a clear rule of how to choose one out of them. Based on information theory, I introduce a criterion of how to choose this parameter automatically. A further important problem in clustering is the choice of the number of distinct clusters in a data set. Based on the log-likelihood ratio test, I will derive a statistical criterion that sequentially splits a hierarchical clustering into separate clusters until there is no more statistical evidence for further splitting clusters. This criterion still requires the choice of a parameter, the type-I error rate of the statistical test. Therefore, I derive a variant of this criterion that is parameter-free and determines the number of clusters based on a generalized information criterion that can be instantiated with Akaikes Information Criterion (AIC; Akaike, 1974) and the Bayesian Information Criterion (BIC; Schwarz, 1978). Taking all this together, we obtain an important parameter-free clustering technique for the analysis of multivariate time series.

A second situation is common when analyzing large data sets: Hypotheses about the data or parts of it exist and can be formalized in a model that determines relations between variables. In the behavioral and social sciences, Structural Equation Models (SEMs) have become widely accepted as a modeling tool for the relation of latent and observed variables (Bollen, 1989). It can be perceived as a unification of several multivariate analysis techniques (cf. Fan, 1997). Traditionally, SEM is a confirmatory technique, that is, the method confirms or rejects models that were devised prior to data collection. It is often the case that such a-priori created models do not reflect the observed data well and there is a need for model modification. Over the years, various methods for model modification have been proposed, e.g., modification indices that are heuristics of how to add parameters to a SEM (Bollen, 1989). Researchers may modify their models when parameter estimates are statistically non-significant or goodness-of-fit indices do not range within established bounds. This modification is typically accomplished by a re-specification of the model, an inclusion of further variables, or a partition of the data set into multiple groups. In the worst case, this involves a simple trial-and-error procedure. Researchers need to be aware that they are moving from a confirmatory to an exploratory approach. MacCallum, Roznowski, and Necowitz (1992) already warned "when an initial model fits well, it is probably unwise to modify it to achieve even better fit because modifications may simply be fitting small idiosyncratic characteristics of the sample" (p. 501). Although such model selection and modification may be effective in practice, the process lacks a formal setting, and it is often unclear how prone the selection is to overfitting the sample. More importantly, such processes are generally susceptible to implicit and explicit selection biases (cf. Kriegeskorte, Simmons, Bellgowan, & Baker, 2009). SEM Trees offer a formal setting for this model selection process. The proposed method is a combination of SEMs and the decision tree

paradigm to trees of SEMs, thereby forming Structural Equation Model Trees (SEM Trees). Given a template SEM, which reflects prior hypotheses about the data, the data set is recursively partitioned into subsets that explain largest differences of parameters of a SEM with respect to variables that were previously not integrated into the model. This allows the detection of heterogeneity of a data set with respect to these variables. SEM Trees provide means to find covariates and covariate interactions that predict differences in structural parameters in observed as well as latent space and facilitate theory-guided exploration of empirical data. With SEM as an undercarriage, SEM Trees can provide an exploratory analysis technique for a large range of models, including regression models (McArdle & Epstein, 1987), factor analysis (Jöreskog, 1969), autoregressive models (McArdle & Aber, 1990; Jöreskog, 1979), latent growth curve models (McArdle & Epstein, 1987) or latent difference score models (McArdle & Hamagami, 2001), and finally, for any model that can be described as a linear combination of observed and latent variables. SEM Trees algorithmically refine initial models of the observed sample while only maintaining findings that generalize to the population. In a process of gaining knowledge, SEM Trees and PDC extract generalizable patterns in the data and thus support researchers in the development of informed theory.

1.3 Outline

In the following chapter, I will review the permutation distribution and the invariance properties that define its high potential for clustering time series. I will show that the α-divergence, a generalization of the important information-theoretic Kullback-Leibler divergence, is a qualified choice as a dissimilarity measure between permutation distributions of time series. This applies because we can find an instance of the divergence that is symmetric and bears an association to the fundamental Bayes error rate. Combining the permutation distribution, the α-divergence, and a hierarchical agglomerative clustering scheme, we obtain PDC as a new clustering method. The permutation distribution requires the choice of an appropriate embedding dimension for a time series. I introduce a novel criterion to detect this parameter automatically in a data-driven fashion. This makes PDC a parameter-free algorithm. A common problem in clustering is the determination of the number of distinct clusters in a data set. An information-theoretic criterion is derived to automatically choose the embedding dimension, a crucial parameter to obtain distinct representations of time series in the permutation distribution. Based on statistical and information-theoretic criteria, rules will be derived that allow the determination of the number of different clusters present in a set of time series.

The third chapter begins with a review of important concepts in SEM. I start by introducing formal and graphical representations of SEM, parameter estimation, and evaluation of goodness-of-fit of the models. I then present details about important representative types of models, the regression model, the autoregressive model, the latent growth curve model, and the factor model. In the following, SEM is united with decision trees, forming SEM Trees. These are tree structures that have a SEM with unique parameter estimates at each node and describe a recursive partitioning of the observations that explain largest differences in the model parameters. I will introduce and discuss four criteria for the evaluation of split candidates in the generation of a tree, and I will prove that these criteria select covariates that maximize information gain and mutual information between the selected covariate and the model-implied distributions. I will talk about how the resulting trees can be interpreted, evaluated, and further

generalized by a pruning technique. A problem in real-world data sets is that they often include missing values. I will show that SEM Trees can deal with missing variables and even present a method to impute missing values.

In the fourth chapter, I will discuss idiosyncrasies of SEM Trees based on three particular model types. In psychology, the factor model has long been of high interest. The concept of measurement invariance, which is crucial in factor analyses, is added to SEM Trees, thus enabling measurement invariant factor-analytic SEM Trees that adhere to the requirements in the field. A similar model is defined by Principal Component Analysis. With a few modifications, we can formulate a PCA as a SEM and build Principal Component Trees that find differences in principal subspaces with respect to covariates. Finally, I will conclude with a fusion of the permutation distribution, which has previously been used for clustering, with SEM Trees, leading to Permutation Distribution Trees that find differences in the permutation distribution with respect to covariates.

The fifth chapter presents demonstrations of the two new methods. A simulation of autoregressive processes will show a first demonstration of the way PDC works. I then demonstrate its capabilities by clustering physiological time series of electrocardiographic and electroencephalographic data. Finally, a data set of eighteen pairs of diverse time series is successfully clustered by PDC and an application of PDC to textual and genetic data is sketched. Whenever results are available in the literature, PDCs performance is compared against those. In other cases, I present comparisons to other clustering algorithms. These examples are followed by applications of SEM Trees to an econometric data set, which is compared to a previous analysis with regression trees. This is followed by applications of SEM Trees to data sets that have been previously analyzed with SEM, which allows a discussion of the SEM Tree results in the light of the previous findings. Indeed, SEM Trees replicate findings in all cases. In the Conclusion, I summarize both methodologies, categorize the approach as an essential building block in data mining and in the process of knowledge gain in the behavioral sciences, and emphasize their importance in supporting researchers in their pursuits of gaining knowledge from data.

2
Permutation Distribution Clustering

In this section, I will introduce Permutation Distribution Clustering (PDC) as a novel solution to the problem of time series clustering that adresses the question of how time series can be distributed into groups of similar objects. A central building block for clustering algorithms is a measure of dissimilarity between the elements that are analyzed. I will review the permutation distribution and derive a dissimilarity measure based on the α-divergence of permutation distributions. This divergence is employed to create a hierarchical clustering. The permutation distribution crucially relies on the choice of a parameter that determines the embedding dimension of the time series. I propose a novel criterion to automatically determine this parameter in a data-driven way. Furthermore, I suggest two criteria based on a likelihood ratio test and on an information-theoretic model selection procedure, to determine the number of clusters by partitioning the hierarchical clustering into informative, distinct clusters. A broad range of applications of PDC is presented in Section 5.

2.1 Problem Setting

Clustering is an unsupervised technique to partition a data set into groups of similar objects. Similarity is formalized via suitable similarity or dissimilarity measures, e.g., the Euclidean distance between time series in a vector representation. Clustering algorithms unravel commonality structures with respect to the dissimilarity measure. Generally, there is a distinction between clustering algorithms and dissimilarity measures on which the algorithms operate. A number of different algorithms exist to this end, mainly differing in whether the number of clusters must be specified beforehand or not, and whether the clustering is flat or hierarchical. Among the most common and well-understood clustering algorithms are k-means (MacQueen, 1967), spectral clustering (cf. von Luxburg, 2007) and agglomerative hierarchical clustering (S. Johnson, 1967). A much larger number of formalizations of dissimilarity between objects exist. If time series are subject to clustering, most approaches reduce data sets to a meaningful description in a low-dimensional feature space or cluster the parameters of a model-based

description of the time series (Liao, 2005). Many such transformations have been described in the literature, among these, Fourier coefficients (Faloutsos, Ranganathan, & Manolopoulos, 1994; Chan & Fu, 1999), Wavelet coefficients (Chan & Fu, 1999), Singular Value Decomposition (Chan & Fu, 1999), or Adaptive Piecewise Constant Approximation (Keogh, Chakrabarti, Pazzani, & Mehrotra, 2001). A third class of clustering approaches derives distance measures, from the mutual information of the time series, e.g., based on an approximation of the Kolmogorov complexity of time series (M. Li, Chen, Li, Ma, & Vitányi, 2004; M. Li & Vitányi, 2008) or on measures of the cost of transforming one time series into the other, like Dynamic Time Warping (DTW; Berndt & Clifford, 1994).

Clustering is sometimes conceived of as an ill-defined problem (Caruana, Elhawary, Nguyen, & Smith, 2006) because real-world applications of clustering lack a ground truth. Even worse, what constitutes a "good" or "correct" clustering is often unclear as the perception of a good clustering differs across users. For example, Caruana et al. (2006) point out that a database containing diverse information about a country's population might be expected to produce different clusters, depending on the query by someone interested in consumer behavior or someone else interested in medical research.

Clustering methods are often evaluated by their application to simulated or real-world data sets with a particular known ground truth. A formal operationalization of clustering quality allows comparisons between different clustering algorithms and dissimilarity measures for specific tasks. In summary, the essential problem of time series clustering consists in finding such a dissimilarity measure that is (1) applicable under a minimal number of assumptions, (2) efficiently computable, and (3) interpretable.

Generally, finding the optimal set of clusters is NP-hard (Aloise, Deshpande, Hansen, & Popat, 2009). Therefore, clustering algorithms typically solve a reduced problem and provide an approximate solution to the problem. Many clustering methods resort to one of the standard clustering mechanisms, such as k-means or sequential hierarchical agglomerative clustering (S. Johnson, 1967). In the following, I will focus on the use of hierarchical clustering since it makes it possible to obtain a hierarchical structure of similarity between all elements that can help investigators to detect outliers and group structures at a glance.

2.2 Sequential Agglomerative Hierarchical Clustering

Sequential agglomerative hierarchical non-overlapping clustering (S. Johnson, 1967) builds a tree structure of clusters by sequentially merging objects to clusters until a single top cluster including all elements is obtained. Initially, each time series is assigned to a cluster with a single member. Based on a chosen dissimilarity measure, clusters are constructed by iteratively merging the two closest clusters until there is only a single top cluster left. This leads to a hierarchy of clusters. Distances between clusters are calculated via a linkage function. In the following, common linkage functions (Halkidi, Batistakis, & Vazirgiannis, 2001) are reiterated. These consider a set of elements O, clusters $C_i \subseteq O$, and define distance functions between clusters based on an arbitrarily chosen distance function between elements $d : O \times O \to \mathbb{R}$.

The *average linkage* dissimilarity calculates the distance between clusters as the average distance between all pair-wise distances of the cluster elements:

$$d_{average}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x, y) \tag{2.2.1}$$

Other common linkage functions include the *single linkage* that defines the distance between two clusters as the minimum distance between elements of each cluster:

$$d_{single}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \tag{2.2.2}$$

In contrast, *complete linkage* combines the clusters with the shortest distance of the elements that are farthest apart:

$$d_{complete}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \tag{2.2.3}$$

The resulting binary tree is visualized as a dendrogram. A dendrogram is a tree structure that represents the hierarchical clusterings. Leaf nodes represent the elements that are to be clustered. Each inner node has two children, which are either other inner nodes or leafs, and represents a merge of the two respective clusters. Typically, the distance between clusters is reflected by the height of the node that represents the merge. Dendrograms in this thesis were generated with the *hcluster* package (Eads, 2008). In this package, nodes are not explicitly depicted but merely show up as vertical lines that join horizontal lines leading to the subclusters.

The essential building block for hierarchical clustering is the choice of the distance function. In the following, I will derive a distance function based on the permutation distribution.

2.3 Notation

In the following, I will talk about discrete random variables, denoted by upper case latin characters, e.g., P,Q, or R. The probability of the elementary outcomes are denoted by the corresponding lower-case characters, e.g., p_i is the probability of the i-th event of P. The outcomes of some random variables will be permutations. In that case, I will use the permutation π as an index. The set of all $m!$ permutations of length m will be called S_m.

2.4 Permutation Distribution

In order to achieve our goal of constructing an algorithm for clustering time series, we require a dissimilarity measure between time series. I will base this measure on differences in the permutation distributions of time series. In the following, I will reiterate the concept of the permutation distribution of a time series.

Given an ordinal time series $X = \{x_t | t = 0 \dots T\}$, we create an embedding of this time series into an m-dimensional space,

$$X' = \{[x_t, x_{t+1}, x_{t+2}, ..., x_{t+(m-1)}] | t = 0 \dots T - (m-1)\} \tag{2.4.1}$$

Our goal is the description of differences between time series as differences between the structures of the embedded patterns. Observe that each element $[x_t, x_{t+1}, x_{t+2}, ..., x_{t+(m-1)}]$ of X'

has a distinct order pattern between the x_i. Bandt and Pompe (2002) suggested counting the frequencies of distinct order patterns in X'. In order to obtain the order patterns, we sort the m values of each element $x' \in X'$ in ascending order. If two elements x'_i,x'_j,$i \neq j$ have the same value, they will keep their original order relative to each other.

Definition 1. Let $x' = [x_1, \ldots, x_m]$ represent a section of an ordinal time series. The permutation obtained by sorting x' in ascending order is called the *codeword* of length m or *m-codeword* of x'.

Since there are $m!$ distinct permutations of length m, there are $m!$ distinct m-codewords. The set of all m-codewords is denoted by the symmetric group S_m. By counting the relative frequencies of codeword occurrences in a time series, we obtain a representation of a time series as a random variable that describes the probability of observing a codeword when uniformly drawing from X':

Definition 2. Let X' be the embedding of a time series and let $\Pi(x)$ be the permutation that x undergoes when being sorted. The permutation distribution of X' is

$$p_\pi = \frac{\#\{x' \in X' | \Pi(x') = \pi\}}{T'} \tag{2.4.2}$$

In the following, we denote the random variable with this distribution by P. We call the dimension of the elements of X' the *embedding dimension*.

The Permutation Distribution (PD) is also referred to as a *codebook*. Note that, in the codebook, the order of the codewords in the original time series is discarded and it solely represents the frequency of unique patterns in the time series. Bandt and Pompe (2002) introduced the entropy of the permutation distribution as a complexity measure of a time series:

Definition 3. The permutation entropy of order $m \geqslant 2$ is the entropy of its codebook P:

$$H(P) = -\sum_{\pi \in S_m} p_\pi \log p_\pi \tag{2.4.3}$$

2.5 Properties of the Permutation Distribution

Previously, we have demanded three properties of a dissimilarity measure for time series clustering: It should be (1) applicable under a minimal number of conditions, (2) efficiently computable, and (3) interpretable. This section summarizes why the permutation distribution is a reasonable choice as a building block for a clustering algorithm.

As we have seen in the previous Section, the permutation distribution is efficiently computable. A single sweep over the time series is sufficient to obtain the codebook representation.

The permutation distribution can be interpreted as a distribution representing the occurrence of distinct order patterns in a time series. The order patterns describe patterns occurring in the time series independently of the absolute values but by the order of the values obtained when sorting them. Intuitively, this interpretation makes sense for clustering time series. Furthermore, the codebooks have important invariance properties. The permutation distribution of a time

series is invariant to all monotonically increasing transformations since these operations do not affect the sorting of the elements. In particular, the permutation distribution is invariant to changing a time series' mean and is invariant to linear scalings of the original time series (Bandt & Pompe, 2002). Two time series that differ only by shifting and scaling are therefore invariant in their PDs. This relieves the investigator of the choice of an appropriate normalization of the data, e.g., subtracting the mean, dividing by the standard deviation, or rescaling the values to a certain range. Monotonically decreasing transformations alter the distribution in a way that could be loosely described as flipping the distribution, but do not affect the distance measure if consistently applied to all time series under consideration.

The permutation distribution is applicable to time series of multiple dimensions. More importantly, sets of time series that are subject to clustering may even differ in length since they are compared by means of their PD only. Also, the comparison of time series is scale-free and independent of a zero point. All these properties make the permutation distribution an important building block for a clustering algorithm.

2.6 Data Structure for Permutations

When calculating the permutation distribution, a data structure that allows efficient random access to the elements is needed. For a chosen embedding dimension, the number of codewords is known and an array is the obvious choice as efficient data structure. In order to get a feeling for the required storage space, assume an embedding dimension of 9, which is already quite large, with more than 300,000 codewords, and a 4 byte unsigned integer representation. In that case, the size of the array per dimension per time series would be 1.45 megabyte, which could handle permutation distributions up to the length of $2^{32} - 1 \approx 10^9$ elements. Furthermore, this means that, in today's computers, several hundred time series can be stored in memory in their permutation distribution representation.

In order to be able to use arrays to store the permutation distribution, we need a canonical assignment of permutations to array indices. I suggest using a bijective mapping from permutations to the natural numbers that was originally introduced by Lehmer (1960). The Lehmer code is based on a factorial numbering system. In a factorial number system, the i-th place in a number represents $i!$. Valid numbers for the i-th place in a factorial number are the numbers from 0 to i. Pseudocode for the calculation of the Lehmer code is given in Algorithm 2.1, and an exemplary calculation of the Lehmer code for a permutation of length 5 is given in Table 2.1. Here, the Lehmer code for the permutation $(4, 2, 3, 0, 1)$ is calculated. resulting in a sorted list of the indices $(0, 1, 2, 3, 4)$. An index iterates through the permutation from left to right, which is depicted as an underscore on the respectively selected symbol in the first column of the table. The currently maintained sorted list is kept in the second column. In each iteration, the position of the selected index in the permutation is determined in the sorted list and then multiplied by the factorial place corresponding to that index. The position is denoted in the third column, the factorial place is denoted in the fourth column. In the first iteration, the position of the first index of the permutation "4" in the sorted list is 4, when counting from zero. This position is the first digit in the factorial number system. The 4 is removed from the sorted list. In the next iteration the "2" is found on position 2, which forms the next digit in the factorial numbering system, and so on. The resulting Lehmer-code of the permutation $(4, 2, 3, 0, 1)$ in the factorial system is given as $4_{4!}2_{3!}2_{2!}0_{1!}0_{0!}$ with subscripts indicating

Permutation	Sorted List	Position	Factorial Place
42301	01234	4	4!
42301	0123	2	3!
42301	013	2	2!
42301	01	0	1!
42301	1	0	0!
Resulting code: $4_{4!}2_{3!}2_{2!}0_{1!}0_{0!} = 4 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! = 112$			

Table 2.1: Example of the calculation of the Lehmer code for the permutation (4,2,3,0,1). There are 120 permutations of length 5. The permutation in this example has a Lehmer code of 112.

the weight of the digit in the factorial numbering system. The resulting number in the decimal system is $4_{4!}2_{3!}2_{2!}0_{1!}0_{0!} = 4 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 0 \cdot 1! + 0 \cdot 0! = 112$. Note that by construction, the digit representing 0! will always be zero, and therefore, no ambiguity arises between digits 0! and 1!.

Algorithm 2.1 Pseudocode to derive the Lehmer code of a codeword w

```
function LEHMER-CODE(w)
    π ← permutation of the indices of w after having sorted w
    m ← length of w
    list ← [1 ... m]
    lehmer ← 0
    for j in 0 ... (m − 1) do
        idx ← position of π_j in list          ▷ first element has position zero
        remove π_j from list
        lehmer ← lehmer + (m − j − 1)! · idx
    end for
    return lehmer
end function
```

2.7 Example: Calculating a Codebook

To illustrate the calculation of the permutation distribution, I present a short example. Let X be a univariate time series as follows

$$X = [1, 5, 8, 3, 6, 0, 3, 2]$$

In this example, the time series will be converted to a codebook by building codewords with an embedding $m = 3$. The following table illustrates the process of obtaining the permutation distribution. By shifting a window of the length of the embedding dimension over the time series, the embedded sequences (left column) are obtained. By sorting the embedded sequences, a permutation of the positions of the elements in the embedded sequence is obtained (center

column). This permutation is used to generate the Lehmer code (right column), a bijective mapping of the m-codeword to the interval $0 \dots (m! - 1)$.

Embedded Sequence	Permutation	Lehmer code
1,5,8	0,1,2	0
5,8,3	1,2,0	3
8,3,6	2,0,1	4
3,6,0	1,2,0	3
6,0,3	2,0,1	4
0,3,2	0,2,1	1

The resulting Permutation Distribution $\hat{P}$ is obtained by counting the frequency with which each Lehmer code occurs. By dividing this number by the total number of codewords, we obtain estimates for the probability of seeing a codeword:

Lehmer code i	0	1	2	3	4	5
$\hat{p}_i$	1/6	1/6	0	2/6	2/6	0

2.8 Dissimilarity Based on the Permutation Distribution

In the following, I derive a dissimilarity measure between time series based on their codebooks. For clustering, dissimilarity measures are usually *semimetrics* or *metrics*:

Definition 4. Let M be a set. A *semimetric* on M is a function $d : M \times M \to \mathbb{R}$ such that the following conditions hold:

- Symmetry
  $\forall x, y \in M, d(x, y) = d(y, x)$
- Identity
  $\forall x, y \in M, d(x, y) = 0 \Leftrightarrow x = y$
- Non-Negativity
  $\forall x, y \in M, d(x, y) \geq 0$

A semimetric is called a *metric* if it satisfies the triangle inequality

$$\forall x, y, z \in M, d(x, y) + d(y, z) \geq d(x, z) \tag{2.8.1}$$

A function satisfying only the axioms of non-negativity and identity is a *premetric*. In statistics and information theory, metrics are also referred to as *distances*, and premetrics as *divergences*.

2.8.1 The Kullback-Leibler Divergence

Shannon's concept of entropy (Kullback & Leibler, 1951; Shannon, 1951) was previously used by Bandt and Pompe (2002) to determine the complexity of a time series based on their permutation distribution. The natural choice for a dissimilarity between codebooks is the Kullback-Leibler divergence or *relative entropy*, which is closely related to the concept of entropy:

Definition 5. The Kullback-Leibler divergence between two codebooks P and Q with the embedding dimension m is defined as

$$D_{KL}(P\|Q) = \sum_{\pi \in S_m} p_\pi \log \frac{p_\pi}{q_\pi} \tag{2.8.2}$$

Since the Kullback-Leibler divergence in the above definition is not symmetric, we can employ a symmetrized version to obtain a dissimilarity measure for codebooks. Note that there are other possibilities to symmetrize the Kullback-Leibler divergence (D. H. Johnson & Sinanovic, 2001).

Definition 6. The symmetric Kullback-Leibler divergence between two codebooks P and Q with embedding dimension m is defined as

$$D_{KLS}(P||Q) = \frac{1}{2}\left(D_{KL}(P\|Q) + D_{KL}(Q\|P)\right) \tag{2.8.3}$$

There is a computationally more efficient representation:

Lemma 7. *The symmetric Kullback-Leibler divergence simplifies to*

$$D_{KLS}(P||Q) = \frac{1}{2}\sum_{\pi \in S_m} (p_\pi - q_\pi) \log\left(\frac{p_\pi}{q_\pi}\right) \tag{2.8.4}$$

Proof.

$$\begin{aligned}
D_{KLS}(P||Q) &= \frac{1}{2}\left(D_{KL}(P\|Q) + D_{KL}(Q\|P)\right) && (2.8.5)\\
&= \frac{1}{2}\left(\sum_{\pi \in S_m} p_\pi \log\left(\frac{p_\pi}{q_\pi}\right) + \sum_{\pi \in S_m} q_\pi \log\left(\frac{q_\pi}{p_\pi}\right)\right) && (2.8.6)\\
&= \frac{1}{2}\left(\sum_{\pi \in S_m} p_\pi \log\left(\frac{p_\pi}{q_\pi}\right) - \sum_{\pi \in S_m} q_\pi \log\left(\frac{p_\pi}{q_\pi}\right)\right) && (2.8.7)\\
&= \frac{1}{2}\sum_{\pi \in S_m} (p_\pi - q_\pi) \log\left(\frac{p_\pi}{q_\pi}\right) && (2.8.8)
\end{aligned}$$

□

Lemma 8. *The symmetric Kullback-Leibler divergence is a semimetric, but not a metric.*

Proof. Identity and Non-negativity are obvious. Symmetry follows by construction. The triangle inequality is violated, as can be seen in the following example: Let P, Q, R be codebooks with an embedding dimension of 2 and probabilities $p_1 = \frac{1}{2}, p_2 = \frac{1}{2}, r_1 = \frac{3}{4}, r_2 = \frac{1}{4}, q_1 = 1 - 2^{-10}, q_2 = 2^{-10}$

$$
\begin{aligned}
D_{KLS}\left(P||R\right) + D_{KLS}\left(R||Q\right) &\geq D_{KLS}\left(P||Q\right) \\
\sum_{i=1}^{2}\left(p_i - q_i\right)\log\left(\frac{p_i}{q_i}\right) + \left(r_i - q_i\right)\log\left(\frac{r_i}{q_i}\right) &\geq \sum_{i=1}^{2}\left(p_i - q_i\right)\log\left(\frac{p_i}{q_i}\right) \\
\sum_{i=1}^{2}\left(p_i\log p_i + q_i\log q_i - p_i\log q_i - q_i\log p_i\right) & \\
+\sum_{i=1}^{2}\left(r_i\log r_i + q_i\log q_i - r_i\log q_i - q_i\log r_i\right) &\geq \sum_{i=1}^{2}\left(p_i\log p_i + q_i\log q_i - q_i\log p_i - p_i\log q_i\right) \\
\sum_{i=1}^{2} r_i\left(2\log r_i - \log p_i - \log q_i\right) & \\
+\sum_{i=1}^{2}\left(p_i\left(\log q_i - \log r_i\right) + q_i\left(\log p_i - \log r_i\right)\right) &\geq 0
\end{aligned}
$$

Using the logarithm to the base 2, the left part of the above inequality evaluates to -2.50, which violates the inequality.

□

The symmetric KL-divergence remains problematic inasmuch as the measure is undefined if one or more codewords are not observed in a time series, i.e., it is undefined for a codebook P, with one or more $p_i = 0$. This problem also arises in other domains, e.g., when estimating the occurrence of n-grams in natural languages, and is known as the zero-frequency problem (Witten & Bell, 1991). A typical remedy is the application of the Laplacian rule of adding one to the number of observations of each codeword (Jeffreys, 1948), which can be seen as setting a prior to the multinomial probability distribution and in fact derives from the Bayesian rule. However, Gale and Church (1994) report that this estimation can be severely biased. Despite the choice of the correction for this bias in estimation, ideally, we would like to consider a divergence measure that is not only symmetric but can also naturally deal with zero frequencies while still being related to the notion of relative entropy. Therefore, I will consider a generalization of the Kullback-Leiber divergence that will eventually lead to a divergence that can naturally handle zero frequency values. Nevertheless, the problem remains that larger embedding dimensions lead to less reliable estimates of codeword frequencies. This is discussed later in this chapter when I discuss determining the embedding dimension.

2.8.2 The Family of α-Divergences

A generalization of the Kullback-Leiber divergence is the concept of α-divergence (Amari, 2007; Amari & Nagaoka, 2007), which was originally proposed by Chernoff (1952). There are notational differences in the literature. In some cases, normalizing constants are added and the restriction to random variables is released. Since the codebooks are discrete random variables that are normalized by definition, we define a simplified version for discrete random variables.

Definition 9. The *α-divergence* of order $\alpha \in \mathbb{R}$ between two codebooks P and Q with embedding dimension m is defined as

$$D_\alpha = \frac{\sum_{\pi \in S_m} \alpha p_\pi + (1-\alpha) q_\pi - p_\pi^\alpha q_\pi^{1-\alpha}}{\alpha\,(1-\alpha)} \tag{2.8.9}$$

We can simplify this to

$$\begin{aligned} D_\alpha\,(P||Q) &= \frac{1 - \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha}}{\alpha\,(1-\alpha)} && (2.8.10) \\ &= \frac{1 - \sum_{\pi \in S_m} \left(\frac{p_\pi}{q_\pi}\right)^\alpha q_\pi}{\alpha\,(1-\alpha)} && (2.8.11) \end{aligned}$$

The α-divergence is a premetric (Amari, 2007). It is apparent from the definition that the α-divergence is in general not symmetric. However, we will see later that for a certain choice of α it will become symmetric. First, we can discover interesting relations of the α divergence to the Kullback-Leibler divergence, the χ^2-divergence, and the Hellinger distance (cf. Minka, 2005). In the following, the formal proofs are given:

Theorem 10. *Let P and Q be two codebooks. For $\alpha \to 0$ and $\alpha \to 1$, the Kullback-Leibler divergence is obtained as a special case of the α-divergence of P and Q*

$$D_{\alpha=1}\,(P||Q) = \lim_{\alpha \to 1} D_\alpha\,(P||Q) = KL\,(P||Q)$$

$$D_{\alpha=0}\,(P||Q) = \lim_{\alpha \to 0} D_\alpha\,(P||Q) = KL\,(Q||P)$$

Proof. A similar proof can be found in Cevhe and Beirami (2008). Since both numerator and denominator have a limit of zero for $\alpha \to 0$, we can apply L'Hôpital's rule to Equation 2.8.10 and thus differentiate the nominator with respect to α

$$\begin{aligned} \frac{\partial\left(1 - \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha}\right)}{\partial \alpha} &= -\sum_{\pi \in S_m} \left(p_\pi^\alpha \log\left(p_\pi\right) q_\pi^{1-\alpha} + \log\left(q_\pi\right) q_\pi^{1-\alpha} p_\pi^\alpha\right) \\ &= \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} \left[\log\left(q_\pi\right) - \log\left(p_\pi\right)\right] \\ &= \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} \log\left(\frac{q_\pi}{p_\pi}\right) \end{aligned}$$

And likewise, we differentiate the denominator with respect to α

$$\frac{\partial \alpha(1-\alpha)}{\partial \alpha} = 1 - 2\alpha$$

Combining both

$$\begin{aligned}\lim_{\alpha\to 0} D_\alpha\left(P||Q\right) &= \frac{\sum_{\pi\in S_m} p_\pi^\alpha q_\pi^{1-\alpha}\log\left(\frac{q_\pi}{p_\pi}\right)}{1-2\alpha}\\ &= \sum_{\pi\in S_m} q_\pi \log\left(\frac{q_\pi}{p_\pi}\right)\\ &= KL\left(Q||P\right)\end{aligned}$$

Analogously, this proof can be found for $\alpha = 1$. □

The α-divergence also includes the χ^2-divergence which is the same as the Pearson χ^2-test statistic usually used in hypothesis testing to reject the hypothesis that two multinomial distributions are equal.

Theorem 11. *Let P and Q be two codebooks. For the choices of $\alpha = -1$ and $\alpha = 2$, the χ^2-divergence is obtained from the α-divergence of the two codebooks*

$$D_{\alpha=2}\left(P||Q\right) = \frac{1}{2}\sum_{\pi\in S_m}\frac{\left(p_\pi - q_\pi\right)^2}{q_\pi}$$

$$D_{\alpha=-1}\left(P||Q\right) = \frac{1}{2}\sum_{\pi\in S_m}\frac{\left(p_i - q_i\right)^2}{p_i}$$

Proof. For $\alpha = 2$,

$$\begin{aligned}D_{\alpha=2}\left(P||Q\right) &= \sum_{\pi\in S_m}\frac{\alpha p_\pi + (1-\alpha)q_\pi - p_\pi^\alpha q_\pi^{(1-\alpha)}}{\alpha\left(1-\alpha\right)}\\ &= -\frac{1}{2}\sum_{\pi\in S_m} 2p_\pi - q_\pi - p_\pi^2 q_\pi^{-1}\\ &= \frac{1}{2}\sum_{\pi\in S_m} q_\pi^{-1}\left(p_\pi^2 - 2p_\pi q_\pi + q_\pi^2\right)\\ &= \frac{1}{2}\sum_{\pi\in S_m}\frac{\left(p_\pi - q_\pi\right)^2}{q_\pi}\end{aligned}$$

which is the Pearson χ^2-test statistic. Analogously, the proof is found for α=-1. □

Lemma 12. *Let P and Q be two codebooks. The α-divergence of P and Q for the choice of $\alpha = \frac{1}{2}$ is*

$$D_{\alpha=\frac{1}{2}} = 4\left(1 - \sum_{\pi\in S_m}\sqrt{p_\pi q_\pi}\right)$$

Proof. The α-divergence for $\alpha = \frac{1}{2}$ can be written as

$$\begin{aligned} D_{\alpha=\frac{1}{2}} &= \sum_{\pi \in S_m} \frac{\alpha p_\pi + (1-\alpha) q_\pi - p_\pi^\alpha q_\pi^{(1-\alpha)}}{\alpha (1-\alpha)} \\ &= \sum_{\pi \in S_m} \frac{\frac{1}{2} p_\pi + \frac{1}{2} q_\pi - (p_\pi q_\pi)^{\frac{1}{2}}}{\frac{1}{4}} \\ &= 2 \sum_{\pi \in S_m} p_\pi + 2 \sum_{\pi \in S_m} q_\pi - 4 \sum_{\pi \in S_m} (p_\pi q_\pi)^{\frac{1}{2}} \\ &= 4 - 4 \sum_{\pi \in S_m} \sqrt{p_\pi q_\pi} \\ &= 4 \left(1 - \sum_{\pi \in S_m} \sqrt{p_\pi q_\pi} \right) \end{aligned}$$

An interesting similarity to the Squared Hellinger distance (Kailath, 1967; Rao, 1995) can be revealed: □

Definition 13. The Squared Hellinger distance is

$$D^2_{Hellinger}(P||Q) = \sum_{\pi \in S_m} (\sqrt{p_\pi} - \sqrt{q_\pi})^2$$

Theorem 14. *$D_{\alpha=1/2}$ is twice the Hellinger divergence.*

$$D_{\alpha=\frac{1}{2}} = 2 \cdot D^2_{Hellinger}$$

Proof. The relation is the following

$$\begin{aligned} \sum_{\pi \in S_m} (\sqrt{p_\pi} - \sqrt{q_\pi})^2 &= \sum_{\pi \in S_m} p_\pi - 2\sqrt{p_\pi q_\pi} + q_\pi \\ &= \sum_{\pi \in S_m} p_\pi - 2 \sum \sqrt{p_\pi q_\pi} + \sum_{\pi \in S_m} q_\pi \\ &= 2 \left(1 - \sum_{\pi \in S_m} \sqrt{p_\pi q_\pi} \right) \\ &= \frac{1}{2} \left(4 \left(1 - \sum_{\pi \in S_m} \sqrt{p_\pi q_\pi} \right) \right) \\ 2 \cdot D^2_{Hellinger} &= D_{\alpha=\frac{1}{2}} \end{aligned}$$

□

Theorem 15. *$D_{\alpha=1/2}(P||Q)$ and the Squared Hellinger distance are metrics*

Proof. $D_{\alpha=1/2}(P||Q)$ satisfies the triangle inequality

$$D_{\alpha=1/2}(P||R) + D_{\alpha=1/2}(R||Q) \geqslant D_{\alpha=1/2}(P||Q)$$

$$\begin{aligned} 1 - \sqrt{p_\pi r_\pi} + 1 - \sqrt{r_\pi q_\pi} &\geq 1 - \sqrt{p_\pi q_\pi} \\ \sqrt{p_\pi r_\pi} + \sqrt{r_\pi q_\pi} - \sqrt{p_\pi q_\pi} &\leq 1 \end{aligned}$$

In the dimension of r_π this function is maximal for $r_\pi = 1$ (remember that $0 \leq r_\pi \leq 1$), that is, we possibly break this condition maximally with respect to b. That leaves us with

$$\begin{aligned} \sqrt{p_\pi} + \sqrt{q_\pi} - \sqrt{p_\pi q_\pi} - 1 &\leq 0 \\ \left(\sqrt{p_\pi} - 1\right)\left(\sqrt{q_\pi} - 1\right) &\geq 0 \end{aligned}$$

This is always true since $p_\pi \geqslant 0$ and $q_\pi \geq 0$.

It follows from Theorem 14 that the Squared Hellinger distance also satisfies the triangle inequality. □

I have discussed several choices of divergences between probability distributions. A summary is given in Table 2.2, and an illustration of the behavior of the presented divergences in two simplified cases is given in Figure 2.8.1. It shows divergences of codebooks with an embedding dimension of 2. The first example shows divergences between a codebook with distribution $[0.5, 0.5]$ and a codebook with distribution $[\beta, 1-\beta]$. The second example features divergences between a codebook with distribution $[0.8, 0.2]$ and again the codebook with uniform distribution.

To extend the presented methods to multivariate time series, codebooks are separately calculated for each of the n dimensions of each time series. After having calculated the n divergences between corresponding dimensions of two time series, the $L2$-norm of the resulting point in n-space is assigned as the resulting total dissimilarity. This dissimilarity measure assumes independence of the dimensions of the time series and assumes equal importance of the time series in the clustering process.

Definition 16. Let X, Y be two n-dimensional time series. Let P_i be the codebook of the dimension i of X and let Q_i be the codebook of dimension i of Y. Let $D(P_i, Q_i)$ be a measure of dissimilarity between the codebooks. The multivariate dissimilarity is

$$D_{multivariate}(P, Q) = \sqrt{\sum_i^n D(P_i, Q_i)^2}$$

I continue with further theoretical motivations for choosing the α-divergence.

	α	symmetric	triangle inequality	premetric
Kullback-Leibler Divergence	0/1	-	-	•
χ^2-Divergence	-1/2	-	-	•
Squared Hellinger Divergence	$\frac{1}{2}$	•	•	•
Euclidean Distance	-	•	•	•
Symmetrized KL Divergence	-	•	-	•

Table 2.2: An overview of divergences between codebooks. If divergences are special cases of the α-divergence, the first column lists the respective α-parameter. The other columns indicate metric properties of the divergences.

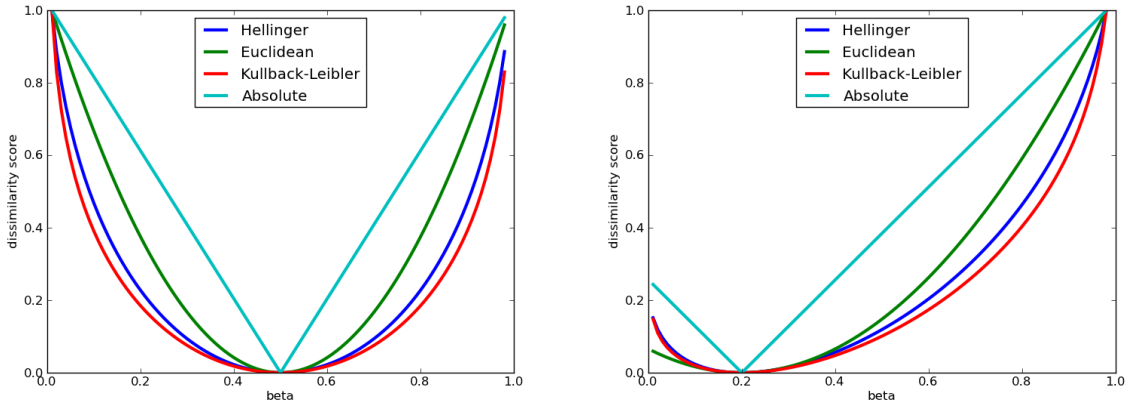


Figure 2.8.1: An illustration of the behavior of the Squared Hellinger distance, Euclidean distance, absolute distance and the symmetric Kullback-Leibler divergence. Left: Dissimilarity scores between a distribution of embedding dimension 2 with distribution $[0.5, 0.5]$ versus a distribution $[\beta, 1-\beta]$. Right: Dissimilarity scores between a distribution of embedding dimension 2 with distribution $[0.8, 0.2]$ versus a distribution $[\beta, 1-\beta]$.

2.8.3 Relation Between the Chernoff Bound and α-Divergence

The *Chernoff bound* in Bayesian decision theory describes a bound for the error of an optimal probabilistic classifier. It is closely related to the α-divergence as described by Hero, Ma, Michel, and Gorman (2001) and Nielsen (2011). The following derivations of the Chernoff bound can similarly found in Duda, Hart, and Stork (2001).

In Bayesian decision theory, the optimal classification decision is based on finding the class with the maximum posterior probability of seeing the class label given the observation. Up to a scaling factor, the posterior distribution is the prior probability of the class multiplied with the likelihood of the observations. This is a simple usage of the fundamental law of Bayes. Let X be a random variable representing observations with x being a realization of X, and the random variable Y represent classes y to which the observations belong. The posterior probability of the class y given the observations x is

$$Pr\left(Y = y|X = x\right) = \frac{Pr\left(X = x|Y = y\right) Pr\left(Y = y\right)}{Pr\left(X = x\right)} \tag{2.8.12}$$

Assuming that the full probabilistic structure of the problem is known, classification of an observation x according to the rule

$$\arg\max_{y} Pr\left(Y = y|X = x\right) \tag{2.8.13}$$

is the Bayes-optimal decision rule. Particularly, each newly observed data point is classified as belonging to the class with the maximum posterior probability. The decision rule minimizes the average probability of misclassification (cf. Duda et al., 2001), which we term $P(Error)$. In practice, the probabilistic structure of the problem is not fully known. This paves the way for numerous classification algorithms based on different grades of prior knowledge and structural assumptions about the data.

In the following, we consider the cases in which a decision has to be made between only two classes, that is, the elementary outcomes of Y are y_1 for one class and y_2 for the other class. For a two-class problem in the Bayesian framework, the probability of making a misclassification after having seen a datapoint x is given by

$$Pr(Error|x) = \min\left[Pr\left(Y = y_1|x\right), Pr\left(Y = y_2|x\right)\right]$$

Clearly, we cannot perform better than guessing the most likely class given the observation, and this will lead to an error that corresponds to the posterior probability of the class that we have not chosen. By approximating the minimum function by a smooth function, an interesting relation to the α-divergence appears.

Lemma 17. *The minimum function can be lower-bounded by the following smooth function for* $\alpha \in [0..1]$ *and* $x, y \geqslant 0$

$$\min\left(x, y\right) \leq x^{\alpha} y^{1-\alpha}$$

Proof. for $\alpha \in [0 \ldots 1]$ and $x, y \geqslant 0$. The proof is easily obtained as

$$\begin{aligned} x \geqslant y \Rightarrow & \left(\frac{x}{y}\right)^{\alpha} \geqslant 1 \\ & \left(\frac{x}{y}\right)^{\alpha} y \geq y \\ & x^{\alpha} y^{1-\alpha} \geqslant y \\ & x^{\alpha} y^{1-\alpha} \geqslant \min(x, y) \end{aligned}$$

□

Theorem 18. *Let $P(Error)$ be the error of an optimal two-class classifier as above. A bound for this error is given as*

$$Pr(Error) \leq Pr(Y = y_1)^{\alpha} Pr(Y = y_2)^{1-\alpha} \sum_x Pr(X = x|Y = y_1)^{\alpha} Pr(X = x|Y = y_2)^{1-\alpha}$$

Proof.

$$\begin{aligned} Pr(Error) &= \sum_x Pr(Error|x) Pr(x) \\ &= \sum_x \min\left[Pr(Y = y_1|x), Pr(Y = y_2|x)\right] Pr(x) \\ &= \sum_x \min\left[\frac{Pr(x|Y = y_1) Pr(Y = y_1)}{Pr(x)}, \frac{Pr(x|Y = y_2) Pr(Y = y_2)}{Pr(x)}\right] Pr(x) \\ &= \sum_x \min\left[Pr(x|Y = y_1) Pr(Y = y_1), Pr(x|Y = y_2) Pr(Y = y_2)\right] \\ &\leq \sum_x \left(Pr(x|Y = y_1) Pr(Y = y_1)\right)^{\alpha} \cdot \left(Pr(x|Y = y_2) Pr(Y = y_2)\right)^{1-\alpha} \\ &\leq Pr(Y = y_1)^{\alpha} Pr(Y = y_2)^{1-\alpha} \sum_x Pr(X = x|Y = y_1)^{\alpha} Pr(X = x|Y = y_2)^{1-\alpha} \end{aligned}$$

□

If we assume that the permutation distribution represents a class-conditional probability distribution, and we assume non-informative prior probabilities of the classes, we obtain a bound on the probability of misclassification, and we can derive an optimal value of α.

Theorem 19. *Let the codebooks P and Q describe class-conditional probability distributions for two different classes with equal prior probabilities. The average error probability for a two-class classification is bounded by*

$$Pr(Error) \leq \frac{1}{2} \sum_{\pi \in S_m} p_{\pi}^{\alpha} q_{\pi}^{1-\alpha}$$

Proof. By substituting $Pr(Y = y_1) = Pr(Y = y_2) = \frac{1}{2}$ in Theorem 18 and interpreting codewords as conditional probabilities on their classes $p_\pi = Pr(X = \pi|Y = y_1)$ and $q_\pi = Pr(X = \pi|Y = y_2)$, we obtain

$$\begin{aligned} Pr(Error) &\leq Pr(Y = y_1)^\alpha Pr(Y = y_2)^{1-\alpha} \sum_x Pr(X = x|Y = y_1)^\alpha Pr(X = x|Y = y_2)^{1-\alpha} \\ &\leq \left(\frac{1}{2}\right)^\alpha \left(\frac{1}{2}\right)^{1-\alpha} \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} \\ &\leq \frac{1}{2} \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} \end{aligned}$$

□

Theorem 20. *Let $P(Error)$ be the average error probability as above. The following bound relates the α-divergence and the average error probability under equal priors*

$$Pr(Error) \leq \frac{1}{2}\left[1 - D_\alpha(P||Q) \cdot \alpha \cdot (1 - \alpha)\right]$$

Proof.

$$\begin{aligned} Pr(Error) &\leq \frac{1}{2} \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} \\ \frac{1 - 2 \cdot Pr(Error)}{\alpha(1-\alpha)} &\geq \left[\frac{1 - \sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha}}{\alpha(1-\alpha)}\right] \\ \frac{1 - 2 \cdot Pr(Error)}{\alpha(1-\alpha)} &\geq D_\alpha(P||Q) \\ Pr(Error) &\leq -\frac{1}{2}\left[D_\alpha(P||Q)\,\alpha(1-\alpha) - 1\right] \\ Pr(Error) &\leq \frac{1}{2}\left[1 - D_\alpha(P||Q)\,\alpha(1-\alpha)\right] \end{aligned}$$

□

Particularly, an intuitive relation becomes clear here: the larger the α-divergence between two codebooks, the more the misclassification error will decrease.

Theorem 21. *Let P and Q be two codebooks. The choice of $\alpha = \frac{1}{2}$ is the only value of α for which the α-divergence $D_\alpha(P||Q)$ is symmetric.*

Proof.

$$\begin{aligned}
\sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} &= \sum_{\pi \in S_m} q_\pi^\alpha p_\pi^{1-\alpha} \\
\sum_{\pi \in S_m} p_\pi^\alpha q_\pi^{1-\alpha} - q_\pi^\alpha p_\pi^{1-\alpha} &= 0 \\
\left(\frac{p_\pi}{q_\pi}\right)^\alpha q &= \left(\frac{p_\pi}{q_\pi}\right)^{-\alpha} p \\
\left(\frac{p_\pi}{q_\pi}\right)^{2\alpha} &= \frac{p_\pi}{q_\pi} \\
\alpha &= \frac{1}{2}
\end{aligned}$$

□

Therefore, we choose the $D_{1/2}$ divergence as a general choice for α that should be appropriate for our goals, being a metric, arising from the concept of entropy, having a formal appeal by bounding the Bayes-optimal error, and allowing the calculation of dissimilarity between codebooks with zero frequencies.

Taking together, the dissimilarity between permutation distributions and the hierarchical clustering scheme that was initially motivated, we obtain a novel approach to time series clustering: Permutation Distribution Clustering (PDC).

2.9 Time Complexity

An important consideration for the practicality of an algorithm is an analysis of its asymptotic time complexity. Let N be the number of time series under investigation and let T be the length of the longest time series in that set. Let D be the dimensionality of the time series and m the embedding dimension. Calculating the permutation distribution of a time series is linear in the length of the time series. Calculating the Lehmer code for a time series segment of length m requires sorting the segment and is therefore in $O(m \log(m))$. Shifting a window over the D dimensions of N time series and counting the resulting Lehmer codes is thus $O(DTNm \log(m))$. The hierarchical clustering algorithm is polynomial in the number of the resulting N codebooks. Constructing the hierarchical clustering requires $N-1$ iterations, in which two clusters are merged each time, until only a single cluster is left. After each merge, the distance matrix must be searched for the minimum element, which again takes $O(N^2)$. Therefore, the clustering step has a complexity of $O(N^3)$. Eppstein (1998) describes an efficient data structure for the distance matrix that allows updating in $O(N)$ instead of $O(N^2)$, and therefore, this enables the generation of a hierarchical clustering in $O(N^2)$. Merging clusters will be performed at each iteration. Evaluating the distance measure depends on the dimensionality D of the time series. For average, single, and complete linkage, the evaluation of the distance function iterates all current cluster members, and therefore, each evaluation is in $O(ND)$. There are a total of $N-1$ such iterations. Summing up, the time complexity of permutation distribution clustering is $O(DTNm \log(m) + DN^2)$.

2.10 Minimum Entropy Heuristic for Choosing the Embedding Dimension

The permutation distribution is essentially determined by the choice of the embedding dimension, which determines the number of codewords and ultimately the complexity of the representation. Bandt and Pompe (2002) reported that they choose an embedding dimension between 3 and 7 depending on the time series. Cao, Tung, Gao, Protopopescu, and Hively (2004) report that they empirically found an embedding dimension of 3 and 4 to be insufficient to detect changes in the complexity of time series and rather recommend a choice of 5 to 7. Choosing the criterion is critical for the use of PDC. A small embedding dimension could result in a codebook with too few distinct codewords, such that differences between time series cannot be detected. On the other hand, large embedding dimensions decrease the reliability of the frequency estimates of the permutation distribution. The problem is even more severe if large embedding dimensions lead to cases where frequency estimates of codewords are zero. It has been discussed earlier that this renders some dissimilarity measures, such as the Kullback-Leibler divergence, unusable, and naïve corrections can lead to severe biases in the dissimilarity measure (Gale & Church, 1994). In the following, I will derive a criterion for determining the embedding dimension if no other information about an appropriate choice of embedding dimension is given. Importantly, employing this criterion makes PDC a parameter-free clustering algorithm.

In order to formalize the observations about embedding dimensions and their consequences on the representation above, I resort to entropy to describe the distinctiveness of the permutation distribution of time series. We can observe two interesting consequences from inappropriate choices of the embedding dimension:

1. If an embedding dimension is too low to represent distinct order patterns in the data set, the distribution of the permutations will approximate a uniform distribution;

2. For embedding dimensions approaching the number of observations, we observe that the permutation distribution flattens out.

A simple probabilistic argument illustrates this observation. When moving from an embedding dimension $m-1$ to an embedding dimension m, the frequency of each codeword is distributed to up to m codewords in the codebook with the larger embedding. In a finite sample, increasing the embedding dimension decreases the number of observed codewords, and therefore, frequency estimates are less reliable. Eventually, a maximum embedding dimension is reached with only a single observed codeword and the large number of other codewords having an estimated zero frequency. We observe that both extremely small and large embedding dimensions tend to reveal only low information content about the time series.

It seems reasonable that entropy will help us to find the best trade-off between representational power of patterns and reliability of the estimated frequencies. I suggest operationalizing this measure as the average entropy of all codebooks that are subject to clustering. A low entropy indicates that the chosen number of codewords for the codebooks yield on average the most distinctive representation of time series.

Generally, the unnormed entropy measure for a discrete random variable X increases with the number of discrete outcomes of X. It attains its maximum for a uniform distribution, i.e., $x_i = \frac{1}{N}$ for all outcomes i. In order to be able to compare the entropy estimate across

different embedding dimensions with different numbers of codewords, we must normalize by the maximally achievable entropy. Thus, we obtain the following normalization term for the entropy measure, depending on the embedding dimension m

$$\begin{aligned} H(X) &= -\sum_{\pi \in S_m} x_\pi \log x_\pi \\ &= -\sum_{\pi \in S_m} \frac{1}{m!} \log \frac{1}{m!} \\ &= -\log \frac{1}{m!} = \log m! \end{aligned}$$

The essential idea that facilitates an unbiased estimate of entropy is the reduction of codebooks to those that do not contain zero frequencies. This can be interpreted as deleting all codewords from the representation that were never observed. The crucial point is that the correct normalization of the entropy of an m-embedding is not based on dividing by the total number of codewords $m!$ but by the number of non-zero-frequency codewords. Thus, we determine the entropy on the effective distribution without over-penalizing increasing embedding dimensions that do not exploit all possible codewords.

Definition 22. Let P be a codebook of embedding dimension m. Let $log_0(x)$ be 0 if $x = 0$, and $\log(x)$ otherwise. The normed entropy of the codebook is

$$e_P(m) = -\frac{\sum_{\pi \in S_m} p_\pi \log_0 p_\pi}{\log\left(\#\left\{x_\pi > 0 | \pi \in S_m\right\}\right)}$$

In order to find the appropriate embedding dimension for clustering a set of time series, we calculate the average entropy of a set of codebooks.

Definition 23. Let X be a set of codebooks of embedding dimension m. The average entropy of these codebooks is

$$e_X(m) = \sum_{P \in X} e_P(m)$$

Finding an appropriate embedding dimension is now reduced to finding the minimum average entropy of the set of codebooks that are subject to clustering. In other words, the criterion recommends increasing the embedding dimension as long as the information gain increases, that is, as long as the entropy of the representation decreases when increasing the embedding dimension. A schematic drawing in Figure 2.10.1 illustrates the heuristic's mechanisms. The embedding dimension with the minimum average entropy is a representation of the codewords that is most distinct from an uniform distribution of the codewords and thereby promises an advantageous representation to calculate dissimilarities between codebooks. The above heuristic is unchanged if multidimensional time series are under investigation. Definition 23 remains unchanged, and the average entropy for each embedding dimension is calculated across the codebooks of each dimension of each time series.

I call this heuristic the *Minimum Entropy* (MinE) heuristic. The MinE algorithm is given in pseudocode in Algorithm 2.2. The algorithm investigates each dimension of each time series. In

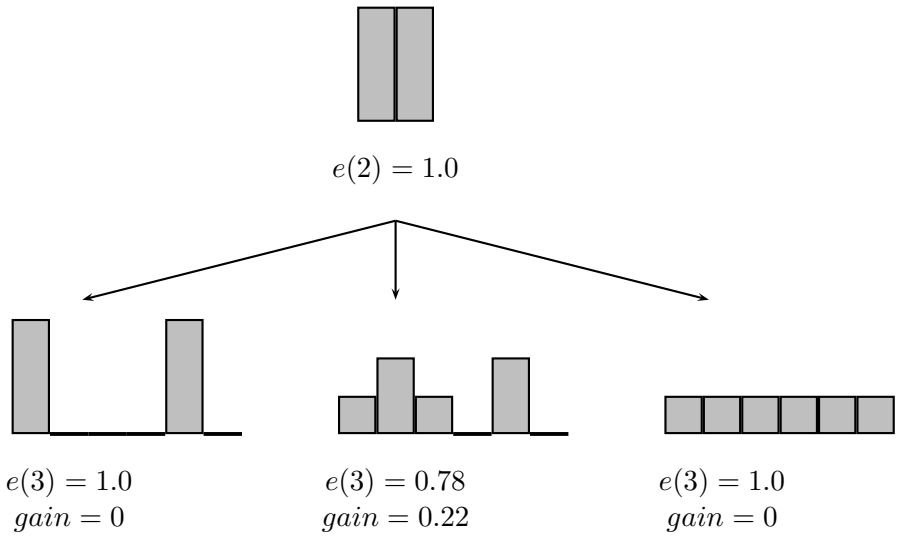


Figure 2.10.1: Schematic diagram of information gain, which is the difference in entropy when increasing the embedding dimension. The upper diagram represents a codebook of embedding dimension 2 with uniform distribution of two codewords. The entropy achieves a maximum value. When increasing the embedding dimension to 3, codebooks contain 6 distinct codewords. There is no information gain if distributions with the same entropy are obtained, independently of how many codewords are used (left and right case, below). However, if a more distinct representation is achieved, the entropy will decrease (center, below) and a more successful clustering can be expected.

each iteration, building the codebook and the calculation of the entropy for a time series is in $O(T)$. Therefore, the time complexity of the algorithm is $O(DTN)$. Strictly speaking, the time complexity of the algorithm multiplicatively depends on the number of searched embedding dimensions. By means of the pigeon-hole principle it can be argued that if the number of codewords exceeds the number of observations, at least one codeword will have a zero frequency, and therefore the embedding dimension is potentially too large. On this assumption, the number of embedding dimensions depends on the number of observations by $m! \leq T$, and therefore the runtime complexity is multiplied by a factor of the inverse Γ function, the continuous extension of the factorial function, of T. However, in compliance with the findings of Bandt and Pompe (2002) and Cao et al. (2004), I have never encountered a data set requiring an embedding dimension of larger than 8. It seems safe to search potential embedding dimensions between 3 and a maximum fixed embedding dimension of 9 and to treat this factor as constant in the consideration of time complexity.

In this section, we have obtained a heuristic for the crucial choice of the embedding dimension for PDC. The heuristic is based on selecting the embedding dimension that maximizes the information in the permutation distribution. In Chapter 5, applications of PDC to various data sets are shown. There, I will compare the performance of PDC on different embedding dimensions and will find that this heuristic allows excellent choices of the embedding dimension that maximize clustering performance.

Algorithm 2.2 Pseudocode algorithm for the detection of the embedding dimension with the MinE criterion, given a set of time series T and a range of embedding dimensions that is searched, given by m_{min} and m_{max}.

```
function MINE-HEURISTIC(T, m_min,m_max)
    m_best ← m_min
    e_best ← 1.0
    for m in m_min to m_max do
        entropies ← []
        for j in 1 to length(T) do
            P ← codebook of T_j with embedding dimension m
            P' ← P without zero-frequency codewords
            ent ← entropy of P' / log(number of codewords in P')
            append ent to entropies
        end for
        e ← mean(entropies)
        if e < e_best then
            m_best ← m
            e_best ← e
        end if
    end for
    return m_best
end function
```

2.11 Determining the Number of Clusters

Determining the number of clusters is not trivial and poses an interesting problem for most clustering approaches. Hierarchical clustering tries to avoid this problem by the presentation of a hierarchy of similarities. Nevertheless, in data analysis, it can be helpful to know how many distinct clusters form an appropriate representation of a data set. This information can be gained from a hierarchical clustering by traversing the tree and cutting the clustering tree at each child node into distinct sub-trees until a stopping criterion is reached. There are a number of solutions to the problem of finding the right number of clusters, among these is typically finding a cut-off at which the total explained variance increases only marginally by adding a cluster (Duda et al., 2001; Salvador & Chan, 2004), a principle which can also be based on a statistical test (Tibshirani, Walther, & Hastie, 2001) or on information theoretic approaches (Fraley & Raftery, 1998).

The permutation distribution allows an elegant formulation of a criterion for the determination of the number of clusters. The permutation distribution of two clusters and a merging of the two clusters can each be represented as multinomial models. This allows the formulation of a likelihood ratio test that can be employed to reject a null hypothesis that the two sub-clusters are drawn from the same distribution. In other words, significant values of the respective test statistics indicate that the number of clusters should be increased since they are likely to be drawn from different populations. This allows a straightforward formulation of a statistical criterion that determines when to stop increasing the number of clusters. In the following, I derive a statistical test for the determination of the numbers of clusters when using PDC and present a variant of the criterion based on the Bayesian Information Criterion (Schwarz, 1978) and Akaike's Information Criterion (Akaike, 1973).

2.11.1 Likelihood and Likelihood Ratio of Codebooks

A cluster in PDC represents a set of codebooks, which in turn represent the original time series. Each codebook is, in fact, a multinomial distribution over the codewords. Joining a cluster effectively joins the respective codebooks to an average representative. Based on multinomial likelihood ratio tests, we can obtain a statistical rule for the determination of the number of clusters.

In the following, I will derive a theoretical framework and present an algorithm to determine the number of clusters based on this notion. At first, we need the likelihood of observing codewords under a given codebook:

Theorem 24. *Let $x \in \mathbb{R}^{|S_m|}$ be a vector of observed frequencies for the codewords π with respective frequencies x_π. The likelihood of observing x under a codebook P is given by*

$$L(x|P) = \left(\frac{N!}{\prod_{\pi \in S_m} x_\pi} \right) \prod_{\pi \in S_m} p_\pi^{x_\pi} = N! \prod_{\pi \in S_m} \frac{p_\pi^{x_\pi}}{x_\pi!}$$

with N being the total number of observed codewords. Accordingly, the negative log-likelihood is given as

$$-LL(x|P) = -\log(N!) - \sum_{\pi \in S_m} x_\pi \log(p_\pi) + \sum_{\pi \in S_m} \log(x_\pi!) \tag{2.11.1}$$

Proof. This theorem is an adaptation of the likelihood of a multinomial distribution to the notation for permutation distributions. The multinomial distribution arises from the multinomial theorem. The proof can be found for example in Goldberg (1986). □

Let x be a vector of observed frequencies as above. We refer to the maximum likelihood estimates of a codeword as $\hat{p}_\pi = \frac{x_\pi}{N}$ and the resulting estimated codebook as $\hat{P}$. An interesting reformulation of the multinomial likelihood is given as:

Theorem 25. *Let P and x be as above. Let $\hat{P}$ be the codebook estimated from x. For large values of x_π, the negative log-likelihood of observing x under a model P is approximated by a term involving the Kullback-Leibler divergence*

$$-LL(x) \approx N \cdot KL\left(\hat{P}||P\right)$$

Proof. This proof is based on the usage of the famous Stirling approximation by Stirling and Tweddle (2003), which is given as

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{2.11.2}$$

which allows us to rewrite logarithms of factorials as

$$\log(n!) \approx n\log n - n + O(\log n) \tag{2.11.3}$$

It follows from Equation 2.11.1

$$\begin{aligned}
-LL(x|P) &= -\log(N!) - \sum_{\pi\in S_m} x_\pi \log(p_\pi) + \sum_{\pi\in S_m} \log(x_\pi!) \\
&= -N\log N + N - \sum_{\pi\in S_m} [N\hat{p}_\pi \log(p_\pi)] + \sum_{\pi\in S_m} [N\hat{p}_\pi \log(N\hat{p}_\pi) - N\hat{p}_\pi] \\
&= -N\log N + N - \sum_{\pi\in S_m} N\hat{p}_\pi \log(p_\pi) + N\sum_{\pi\in S_m} \hat{p}_\pi \log(\hat{p}_\pi) \\
&\quad +N\sum_{\pi\in S_m} \hat{p}_i \log(N) - N\sum_{\pi\in S_m} \hat{p}_\pi \\
&\quad -N\log N + N - N\sum_{\pi\in S_m} \hat{p}_\pi \log(p_\pi) + N\sum_{\pi\in S_m} \hat{p}_\pi \log(\hat{p}_\pi) + N\log N - N \\
&= N\left[\sum_{\pi\in S_m} \hat{p}_\pi \log(\hat{p}_\pi) - \sum_{\pi\in S_m} \hat{p}_\pi \log(p_\pi)\right] \\
&= N\left[\sum_{\pi\in S_m} \hat{p}_\pi \log\left(\frac{\hat{p}_\pi}{p_\pi}\right)\right] \\
&= N \cdot KL\left(\hat{P}||P\right)
\end{aligned}$$

□

However, the above formulation of the log-likelihood is undefined if one or more $p_\pi = 0$. This comes down to the same problem as encountered previously, when calculating the Kullback-Leibler divergence (cf. Subsection 2.8.1). The same solutions apply here. Either a deletion of zero-frequency codewords or an estimation via codeword-buckets needs to be applied. The deletion of codewords effectively modifies the log-function to the following

$$\log_0\left(\frac{a}{b}\right) = \begin{cases} 0 & if\, a = 0 \vee b = 0 \\ \log\left(\frac{a}{b}\right) & otherwise \end{cases}$$

For large embedding dimension m, the above approximation can be problematic, since the number of codewords will grow in the factorial of the embedding dimensions, and for short time series, the observed frequencies of many codewords will be close to zero or even equal to zero. I suggest improving this approximation for smaller x_i by using a corrected Stirling formula by Gosper (1978).

Theorem 26. *Let $P, \hat{P}$ and x be as above. An approximation of $-LL(x)$ for small x_i is given by*

$$-LL(x|P) \approx N \cdot KL\left(\hat{P}||P\right) + C$$

with an correction factor

$$C = \frac{1}{2}\sum_{\pi \in S_m} \log\left(2\pi x_i\right) - \frac{1}{2}\log\left(2\pi N\right) + (m-1)! \cdot \frac{1}{2}\log\left(\frac{\pi}{3}\right)$$

Proof. Gosper (1978) improved Stirling's approximation by replacing $\sqrt{2\pi n}$ by $\sqrt{2\pi\left(n + 1/6\right)}$, which approximates the remainder of the Stirling series instead of truncating it, resulting in the expression

$$n! \approx \sqrt{2\pi\left(n + 1/6\right)}\left(\frac{n}{e}\right)^n$$

For the correction of our approximation to $\log(n!)$, we need the logarithm of the additional term as an additive factor, which we simplify to

$$\log(n!) \approx \frac{1}{2}\log\left(2\pi n\right) + \frac{1}{2}\log\left(\frac{\pi}{3}\right) + n\log n - n \tag{2.11.4}$$

In order to correctly handle the case of $n = 0$, we have to define the logarithm in the above Equation as $\log(0) = 0$.

In particular, this approximation is an extension of the approximation in Equation 2.11.3 by an addition of the first two terms. We can approximate the last term of Equation 2.11.1 by

$$\sum_{\pi \in S_m} \log\left(x_\pi!\right) \approx \sum_{\pi \in S_m}\left[x_\pi \log\left(x_\pi\right) - x_\pi + \frac{1}{2}\log\left(2\pi x_\pi\right) + \frac{1}{2}\log\left(\frac{\pi}{3}\right)\right]$$

Following the proof in Theorem 25, this allows the approximation of the negative log-likelihood function

$$
\begin{aligned}
-LL(x|P) &\approx -\log(N!) - \sum_{\pi\in S_m} x_\pi \log(p_\pi) + \sum_{\pi\in S_m} \log(x_\pi!) \\
&\approx -N\log N + N - \frac{1}{2}\log(2\pi N) - \frac{1}{2}\log\left(\frac{\pi}{3}\right) \\
&\quad - \sum_{\pi\in S_m} N\hat{p}_\pi \log(p_\pi) + N \sum_{\pi\in S_m} \hat{p}_\pi \log(\hat{p}_\pi) \\
&\quad + N \sum_{\pi\in S_m} \hat{p}_i \log(N) - N \sum_{\pi\in S_m} \hat{p}_\pi + \frac{1}{2} \sum_{\pi\in S_m} \log(2\pi x_i) + m! \cdot \frac{1}{2}\log\left(\frac{\pi}{3}\right) \\
&\approx N \cdot KL\left(\hat{P}||P\right) + \frac{1}{2} \sum_{\pi\in S_m} \log(2\pi x_i) - \frac{1}{2}\log(2\pi N) + (m-1)! \cdot \frac{1}{2}\log\left(\frac{\pi}{3}\right)
\end{aligned}
$$

□

Theorem 27. *Let P and Q be two codebooks. Let x be a vector of observations as above. The log-likelihood ratio of seeing x under P and x under Q is given by*

$$\Lambda(x|P,Q) = \sum_{\pi\in S_m} x_\pi \log(q_\pi/p_\pi) \tag{2.11.5}$$

Proof. This proof follows the generally known proof for the log-likelihood ratio of two multinomial distributions.

By definition the log-likelihood ratio of observations x under models P and Q, is the logarithm of the ratio of $L(x|P)$, the likelihood of x under P, and $L(x|Q)$, the likelihood of seeing x under Q, resulting in

$$
\begin{aligned}
\Lambda(x|P,Q) &= \log\left(\frac{-L(x|P)}{-L(x|Q)}\right) && (2.11.6)\\
&= (-LL(x|P)) - (-LL(x|Q)) && (2.11.7)\\
&= -\log(N!) - \sum_{\pi\in S_m} x_\pi \log(p_\pi) + \sum_{\pi\in S_m} \log(x_\pi!) && (2.11.8)\\
&\quad + \log(N!) + \sum_{\pi\in S_m} x_\pi \log(q_\pi) - \sum_{\pi\in S_m} \log(x_\pi!) && (2.11.9)\\
&= \sum_{\pi\in S_m} x_\pi \log(q_\pi) - \sum_{\pi\in S_m} x_\pi \log(p_\pi) && (2.11.10)\\
&= -\sum_{\pi\in S_m} x_\pi \log\left(\frac{q_\pi}{p_\pi}\right) && (2.11.11)\\
&= \sum_{\pi\in S_m} x_\pi \log\left(\frac{p_\pi}{q_\pi}\right) && (2.11.12)
\end{aligned}
$$

□

Most importantly, we note that there is no need for a correction term since it would cancel out. Interestingly, the likelihood ratio between multinomial distributions is again closely related to the concept of entropy and relative entropy as can be seen in the following.

Lemma 28. *Let P and Q be codebooks. The cross-entropy $H(P \wedge Q) = -\sum_i p_i \log q_i$ can be related to the Kullback-Leibler divergence via the following relation*

$$D_{KL}(P||Q) = -H(P) + H(P \wedge Q)$$

Proof. This relationship is generally known and can be easily shown

$$\begin{aligned} D_{KL}(P||Q) &= \sum_i x_i \log\left(\frac{x_i}{y_i}\right) \\ &= \sum_i x_i \log x_i - \sum_i x_i \log y_i \\ &= -H(P) + H(P \wedge Q) \end{aligned}$$

□

Theorem 29 (Likelihood ratio of codebooks)**.** *The likelihood ratio of observing observations x with MLE $\hat{P}$ between two Permutation Distributions P and Q can be approximated by*

$$\Lambda(x|P,Q) = N \cdot \left[H\left(\hat{P} \wedge P\right) - H\left(\hat{P} \wedge Q\right)\right]$$

Proof. Let the MLE estimates of the observations x be $\hat{P}$ with $\hat{p}_i = x_i/n$. We obtain

$$\begin{aligned} \sum_{i=1}^{n} x_i \log(q_i/p_i) &= N \sum_{i=1}^{N} \hat{p}_i \log(q_i/p_i) \\ &= N \cdot \left[\sum_{i=1}^{N} \hat{p}_i \log q_i - \sum_i \hat{p}_i \log p_i\right] \\ &= N \cdot \left[H\left(\hat{P} \wedge P\right) - H\left(\hat{P} \wedge Q\right)\right] \end{aligned}$$

Under the null hypothesis, this likelihood ratio converges to a χ^2-distribution with $n-1$ degrees of freedom. □

A hierarchical clustering structure is a tree structure representing an increasing number of partitions of the data set. The root node represents a unified cluster that comprises all observed codebooks. When traversing the tree structure, each level splits off one or more data points into a new cluster until each cluster has only a single member at the final level. Finding the number of clusters, given a hierarchical clustering, can thus be interpreted as finding a rule when to stop cutting clusters in a top-down fashion. The likelihood ratio test provides us with an objective criterion for stopping. The null hypothesis for an appropriate statistical test claims that there is no difference between the permutation distributions of two clusters. In other words, we establish the null hypothesis that the observations of both clusters were generated by the same process. Formally, we write

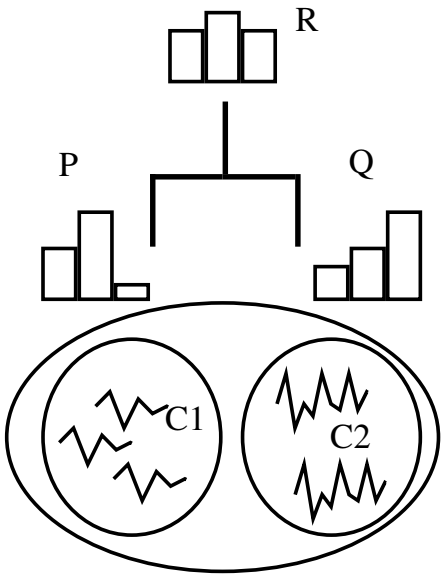


Figure 2.11.1: Schematic drawing of the hierarchical clustering of time series in a codebook representation. Determining the number of clusters based on likelihood ratio tests of the multinomial distribution. In this example, clusters C1 and C2 are hierarchically nested in a super-cluster. Analogously, the joint codebooks P of all elements in C1 and the codebook Q of all elements in C2 are nested in a super-codebook R. Likelihood-ratio tests can serve as a statistical tool to determine whether a single codebook R is sufficient to represent the data or whether two separate codebooks P and Q are more appropriate.

$$H_0 : P = Q$$

for two permutation distributions P and Q whose distributions we estimate from a finite sample. For each node in the tree, we can test whether two codebooks representing the sub-trees are a better representation than a single codebook representing both sub-trees. Figure 2.11.1 illustrates this setting. Formally, we need the notion of joint codebooks that represent a single codebook for a set of time series in a cluster. With this notion, I suggest a likelihood ratio test that indicates whether such a joint codebook of two clusters is a better representation of the data than a single cluster.

Definition 30. Let O be a cluster of m time series, $O = \{X_i | i \dots m\}$. The joint codebook P for a cluster with elements O is given by summing up the individual codebooks of the X_i

$$p_\pi = \frac{\sum_{X' \in O} \#\{x' \in X' | x' \text{ has type } \pi\}}{\sum_{X' \in O} |X'|}$$

Lemma 31. *Let P and Q be codebooks, and N_P and N_Q be the number of observations from which P and Q were estimated. The joint codebook R of two joint clusters is the weighted*

average of the permutation distributions of P and Q. The codewords of R are obtained from P and Q by

$$r_\pi = \frac{N_P}{N_P + N_Q} p_\pi + \frac{N_Q}{N_P + N_Q} q_\pi$$

Proof. Let $x \in \mathbb{R}^{|S_m|}$ and $y \in \mathbb{R}^{|S_m|}$ be two vectors of observed codebooks that are used to estimate the codebooks P and Q respectively. We denote the total number of observed codewords in x with N_P, and the total number in y with N_Q. In particular, this means that the codebooks are estimated with $p_\pi = \frac{x_\pi}{N_P}$ and $q_\pi = \frac{y_\pi}{N_Q}$. A joint codebook estimated from all observations in x and y is

$$\begin{aligned} r_\pi &= \frac{x_\pi + y_\pi}{N_P + N_Q} \\ &= \frac{x_\pi}{N_P + N_Q} + \frac{y_\pi}{N_P + N_Q} \\ &= \frac{N_P}{N_P + N_Q} p_\pi + \frac{N_Q}{N_P + N_Q} q_\pi \end{aligned}$$

□

Based on the likelihood of observations given a codebook, we can formulate a likelihood ratio test, whose distribution is known under the null hypothesis that two codebooks that are merged to a single cluster are actually the same. Whenever we can reject this test, we have evidence that the two clusters indeed represent different processes.

Theorem 32. *Let x be a vector of observations as above. Let C_1 and C_2 be clusters with respective joint codebooks P and Q. Let R be the joint codebook of P and Q. The log-likelihood ratio between a model of two individual clusters and a model of a single joint cluster is given by*

$$LLR\left(x|P,Q\right) = 2 \sum_{\pi \in S_m} x_\pi \log\left(\frac{r_\pi}{p_\pi}\right) + 2 \sum_{\pi \in S_m} x_\pi \log\left(\frac{r_\pi}{q_\pi}\right)$$

Λ is asymptotically χ^2-distributed with $m! - 1$ degrees of freedom under the null hypothesis that the true P and Q are equal.

Proof. The codebook P and Q are two multinomial models nested in two multinomial models representing the joint clusters that have equality constraints across the two models.

$$\begin{aligned} LLR\left(x|P,Q\right) &= -2\left[LL\left(x|P\right) - LL\left(x|P \cap Q\right)\right] - 2\left[LL\left(y|Q\right) - LL\left(y|P \cap Q\right)\right] \\ &= -2\Lambda(x|P, P \cap Q) - 2\Lambda\left(y|Q, P \cap Q\right) \\ &= 2 \sum_{\pi \in S_m} x_\pi \log\left(\frac{r_\pi}{p_\pi}\right) + 2 \sum_{\pi \in S_m} x_\pi \log\left(\frac{r_\pi}{q_\pi}\right) \end{aligned}$$

Wilk's (1938) theorem says that the likelihood ratio of two nested models is asymptotically χ^2-distributed under the null hypothesis that the parameters of the two models P and Q are actually the same. □

Note that $\Lambda(x|P,Q)$ denotes the log-likelihood ratio of observing x under P and x under Q, whereas $LLR(x|P,Q)$ denotes the log-likelihood ratio of observing x under the two models P and Q and a joint model of P and Q. The log-likelihood ratio test statistic asymptotically follows a χ^2-distribution for large sample sizes, but finite sample sizes might indeed deviate from this distribution (Lawley, 1956). Therefore, I adapt the suggested correction by Lawley (1956), which corrects the first order moment of the distribution by dividing the test statistic by the following factor

$$LLR'(x|P,Q) = LLR(x|P,Q) \,/\, \left(1 + \frac{\sum_{\pi \in S_m} \left[\frac{N}{x_i}\right] - 1}{3\,(m! - 1)\,N}\right)$$

This allows the derivation of the following decision rule to determine whether a cluster should be split into two non-overlapping clusters. The decision rule is a stopping rule, that is, whenever the rule evaluates as true, cluster splitting should be stopped. Vice versa, as long as the rule is not met, splitting clusters is recursively continued.

Theorem 33 (Likelihood Ratio Stopping Rule)**.** *Let P and Q be two codebooks representing the joint codebooks of two clusters. Let x be an observation as above. The* likelihood ratio stopping rule *for a chosen significance level α is*

$$LLR(x|P,Q)' < c$$

with c being the critical value derived from the α-quantile of the χ^2-distribution as $P(\chi^2 > c) = \alpha$.

Based on the likelihood of a multi-group multinomial model for the permutation distributions of the clusters, another selection method is popular in the literature. Instead of likelihood ratios, the model selection process could be based on penalized likelihoods of competing models with growing complexity. In the particular case of PDC, models of growing complexity arise by successively increasing the number of clusters. The likelihood of observing the data under the models will increase with growing model complexity. In order to avoid overfitting, this likelihood will be penalized according to a chosen information criterion. Typical criterions include Akaike's Information Criterion (AIC; Akaike, 1974) and the Bayesian Information Criterion (BIC; Schwarz, 1978). Kuha (2004) presents a comparison between AIC and BIC, and mentions that the preference of the information criterion depends on the field. For example, the AIC is often used in economics, whereas the BIC is often used in sociology. Generally, I recommend the AIC for its appealing derivation from the expected Kullback-Leibler divergence between the true model and estimated model. Nevertheless, I will derive the following results for a generic penalized likelihood function that we can afterwards use with a specific instance of information criterion according to individual preferences.

Definition 34. Let P be a codeword and x a vector of observed frequencies as above. Let N be the total number of observations in x. Let $\Psi : N \times N \to R$ be a *penalty.* A *penalized likelihood function* with penalty Ψ is

$$-2LL'(x|P) = -2LL(x|P) + \Psi(N, df)$$

A penalty that satisfies $\Psi(N,a) + \Psi(N,b) = \Psi(N, a+b)$ is a *df-additive penalty.*

Theorem 35. *Let P and Q be two uni-dimensional codebooks of embedding dimension m, representing the joint codebooks of two clusters. Let N be the total number of observed codewords in both codebooks, and x_1 and x_2 be the respective observed frequencies of codewords. Let $\Psi(N, df)$ be a df-additive penalty. The stopping criterion based on the penalized likelihood is*

$$LLR(x|P,Q) \leq \Psi(N, m! - 1)$$

Proof. A model that represents two codebooks individually, has $2(m! - 1)$ degrees of freedom since each model has $m!$ freely estimated parameters under the constraint that the probabilities in each model sum up to 1, which costs one degree of freedom for each model. We define the penalized likelihood function of observing the data under two separate models

$$-2LL'_{separate} = -2LL(x_1|P) - 2LL(x_2|Q) + \Psi(N, 2 \cdot (m! - 1))$$

The likelihood of observing x under a joint model is

$$-2LL'_{joint} = -2LL(x_1 + x_2|P \cap Q) + \Psi(N, m! - 1)$$

Again, whenever a split of a cluster in two clusters is supported by this criterion, the penalized negative likelihood of the two separate models is lower than the penalized negative likelihood of the joint model, or equivalently, the penalized likelihood of the two separate models is larger than the penalized likelihood of the joint model. In contrast, if the difference of the penalized negative likelihoods is negative, there is no evidence according to the criterion Ψ for further cluster splitting

$$\begin{aligned} -2LL'_{joint} - \left(-2LL'_{separate}\right) &< 0 \\ +2LL(x_1|P) + 2LL(x_2|Q) - \Psi(N, 2 \cdot (m! - 1)) & \\ -2LL(x_1 + x_2|P \cap Q) + \Psi(N, m! - 1) &< 0 \\ 2LL(x_1|P) + 2LL(x_2|Q) - 2LL(x_1 + x_2|P \cap Q) - \Psi(N, m! - 1) &< 0 \end{aligned}$$

With the approximation of Theorem 26, this calculation remains tractable for large frequencies if it is based on the calculation of the log-likelihoods of the individual models. However, this is not required. A stopping criterion based on penalized likelihoods can also be interpreted as defining a critical value for the likelihood ratio test statistic without being grounded on explicit distributional assumptions of the null hypothesis. This becomes clear with the following transformation

$$\begin{aligned} 2LL(x_1|P) + 2LL(x_2|Q) - 2LL(x_1 + x_2|P \cap Q) & \\ -\Psi(N, m! - 1) &< 0 \\ -2LL(x_1|P) - 2LL(x_2|Q) + 2LL(x_1 + x_2|P \cap Q) &< -\Psi(N, m! - 1) \\ LLR(x|P,Q) &> \Psi(N, m! - 1) \end{aligned}$$

Since this rule implies continuing splitting, the stopping rule is defined as the opposite condition

$$LLR\left(x|P,Q\right) \leq \Psi\left(N, m! - 1\right)$$

This notion can be extended to multidimensional codebooks: □

Theorem 36. Let P and Q be two d-dimensional codebooks. The stopping criterion based on a penalty is

$$LLR\left(x|P,Q\right) \leq d \cdot \Psi\left(N, m! - 1\right)$$

Proof. Under the assumption that the dimensions of the time series are independent, we obtain a joint model of the d multinomial models for each dimension. Due to independence, the log-likelihood of the joint models is the sum of the individual likelihoods. Likewise, the degrees of freedom of the joint model are d times the degrees of freedom of every individual model, which we previously found to be $m! - 1$. Analogously to the previous proof, we define the BIC for the multidimensional time series as

$$\begin{aligned} -2LL_{separate} &= \sum_{i=1}^{d} \left(-LL\left(x_1|P_d\right) - LL\left(x_2|Q_d\right)\right) + d\Psi\left(N, 2\left(m! - 1\right)\right) \\ -2LL_{joint} &= \sum_{i=1}^{d} \left(-LL\left(x_1 + x_2|P_d \cap Q_d\right)\right) + d\Psi\left(N, m! - 1\right) \end{aligned}$$

Following the previous proof, we obtain the criterion that indicates that cluster splitting should be terminated

$$LLR\left(x|P,Q\right) \leq d \cdot \Psi\left(N, m! - 1\right)$$

□

The BIC (Schwarz, 1978) and the AIC (Akaike, 1973) are instances of a generic penalized likelihood:

Theorem 37. *The BIC penalty function is* $\Psi_{BIC}\left(df, N\right) = df \log\left(N\right)$. *The AIC penalty function* $\Psi\left(df, N\right) = 2 \cdot df$. *Both penalty functions are df-additive.*

Proof. The AIC is defined (Akaike, 1973) as $AIC = -2LL + 2df$, therefore the penalty term is $\Psi_{AIC} = 2df$. The BIC (Schwarz, 1978) is $BIC = -2LL + df \cdot \log N$, therefore the penalty term is $\Psi_{BIC} = df \cdot \log\left(N\right)$.

The Ψ_{BIC} is df-additive:

$$\Psi_{BIC}\left(a, N\right) + \Psi_{BIC}\left(b, N\right) = 2a + 2b = 2(a + b) = \Psi_{BIC}\left(a + b, N\right)$$

The Ψ_{AIC} is df-additive:

$$\Psi_{AIC}\left(a, N\right) + \Psi_{AIC}\left(b, N\right) = a \log N + b \log N = (a + b) \log N = \Psi_{AIC}\left(a + b, N\right)$$

□

This allows us to formulate a stopping rule based on the BIC:

Corollary 38. *Let P and Q be two uni-dimensional codebooks of embedding dimension m, representing the joint codebooks of two clusters. Let N be the total number of observed codewords in both codebooks, and x_1 and x_2 be the respective observed frequencies of codewords. The relative-BIC stopping rule is*

$$LLR\left(x|P,Q\right) \leq \frac{d}{2}\left(m!-1\right)\log N \tag{2.11.13}$$

Analogously, we can derive this criterion based on the AIC:

Corollary 39. *Let P and Q be two uni-dimensional codebooks of embedding dimension m, representing the joint codebooks of two clusters. Let N be the total number of observed codewords in both codebooks, and x_1 and x_2 be the respective observed frequencies of codewords. The relative-AIC stopping rule is*

$$LLR\left(x|P,Q\right) \leq d\left(m!-1\right)$$

In this section, I derived two methods to determine the number of clusters in a hierarchical clustering. Based on likelihoods of observing codewords under a codebook, I derived a criterion based on a likelihood ratio test between a codebook representing a super-cluster and two codebooks representing a partition of this super-cluster. The distribution of the resulting test statistic is known under the null hypothesis that the partitioned codebooks are the same in the population. This formulation allows the determination of the numbers of clusters by employing a sequence of hypothesis tests. Algorithm 2.3 summarizes the steps to determine clusters from a hierarchical clustering based on the likelihood ratio criterion and the BIC criterion. I derived an alternative stopping rule based on information-theoretic criteria, the BIC and the AIC, that can be applied in the same way and do not require the setting of a parameter.

Algorithm 2.3 Pseudocode algorithm for splitting a hierarchical clustering based on either (a) sequential tests of nested multinomial likelihood ratio tests or on (b) sequential application of the AIC or BIC criterion. Arguments of the algorithm are the top cluster C of a hierarchical clustering of time series, the embedding dimension m, and the significance level α for the likelihood ratio test.

```
1: procedure SPLIT-HIERARCHICAL-PDC(C,m,α)
2:     if the number of elemens in C is zero then
3:         return
4:     end if
5:     C_1,C_2 ← partition of C according to hierarchical clustering
6:     P_1 ← joint codebook of all time series in C_1
7:     P_2 ← joint codebook of all time series in C_2
8:     P ← joint codebook after merging P_1 and P_2
9:     N_1 ← 0; N_2 ← 0
10:    x_1 ← empty array of size m!
11:    x_2 ← empty array of size m!
12:    for all time series T in C_1 do
13:        for all codeword c in m-embedding of T do
14:            x_1[c] ← x_1[c] + 1; N_1 ← N_1 + 1
15:        end for
16:    end for
17:    for all time series T in C_2 do
18:        for all codeword c in m-embedding of T do
19:            x_2[c] ← x_2[c] + 1; N_2 ← N_2 + 1
20:        end for
21:    end for
22:    N ← N_1 + N_2
23:    Λ ← −2LL(x_1|P) + 2LL(x_1|P_1) − 2LL(x_2|P) + 2LL(x_2|P_2)
24:    if use likelihood ratio criterion then
25:        crit ← (1 − α)-quantile of χ^2-distribution with m − 1 df
26:    else if use AIC criterion then
27:        crit ← (2m! − 2)
28:    else if use BIC criterion then
29:        crit ← (m! − 1) · log(N)
30:    end if
31:    if Λ > crit then
32:        permanently split C_1 and C_2
33:        SPLIT-HIERARCHICAL-PDC(C_1,m,α)
34:        SPLIT-HIERARCHICAL-PDC(C_2,m,α)
35:    else
36:        return
37:    end if
38: end procedure
```

2.12 Summary

In this section, I discussed dissimilarity measures between time series that are based on an α-divergence between the permutation distributions (PD) of time series. The PD is invariant to all monotonic transformations of the underlying time series. In particular, addition and multiplication of positive constants to the time series do not change its PD, hence the PD is invariant to normalizations, e.g., standardization by subtracting the mean and dividing by standard deviation. This property relieves the researcher of choosing normalization as preprocessing which often has a serious impact on clustering with commonly used metric dissimilarity measures. For example, Kalpakis, Gada, and Puttagunta (2001) report a comparison of the Euclidean distance on differently transformed representations of the data, including Fourier coefficients, wavelet coefficients, principal components, and raw data, and they find that normalization improves the clustering performance on some data sets, whereas it decreases the performance on others. Another advantage gained from this property is the robustness against drift in a signal, which often occurs due to the physical properties of the measurement device. In electroencephalography (EEG), amplifier drifts are observed (Fisch & Spehlmann, 1999) and a common problem in accelerometry is thermal drift of the sensors. Drifts of these kinds can be thought of as a local offset in the relatively small embedding window which does not alter the PD. These properties make PDC an interesting complexity-based alternative to other dissimilarity measures for clustering. In conjunction with a divergence between probability distributions, I proposed applying this measure in a hierarchical clustering scheme to obtain tree-like partitions of a set of multivariate time series. A likelihood ratio test of multinomial distributions provides a statistical criterion for the choice of how many distinct clusters are an appropriate representation of the data set. This test requires setting a parameter in order to determine the critical value of the test. A variant of the same criterion was derived from an information-theoretic perspective and provides a parameter-free criterion for the determination of the number of clusters.

Given the large number of existing clustering algorithms, one might ask why there is a need for a further clustering algorithm. Indeed, there are several reasons. The permutation distribution has important properties for the clustering of time series, most importantly it does not depend on the moments of a time series and is therefore robust to shifts and scalings. Second, the codebook representation allows the comparison of time series that differ in length. Third, many heuristics for the determination of the number of clusters are developed independently of the time series' features or the measure of dissimilarity. Fourth, it is efficient to compute. The distributional model of the codebooks allows a consistent and straightforward formulation of such heuristics based on statistical and information-theoretic tests. All these points make a strong case for the PDC methodology. Applications to simulated and real data sets are presented in Chapter 5 and will demonstrate the versatility of the approach, and provide empirical evidence for the MinE criterion that automatically detects an appropriate embedding dimension.

We have discovered a criterion for choosing the embedding dimension and the number of clusters based on information-theoretic criteria. Importantly, this makes PDC a parameter-free clustering approach. For future applications, it is potentially useful to apply the embedding heuristic separately to each dimension. Choosing different embedding dimensions for different dimensions of the time series, might increase the discriminatory power of the clustering and might help in selecting informative dimensions of the data set. An advantage of keeping equal embedding dimensions across different time series dimensions is the possibility of forming cross-

product codewords that allow modeling interdependencies between multiple dimensions.

Time series segmentation is a challenging problem in many domains. Beyond clustering a set of different time series, clustering algorithms can generally be employed to detect anomalies in time series (Keogh, Lonardi, & Ratanamahatana, 2004) or segment time series into different parts by sliding a window over the time series and treating the resulting shorter segments as independent time series (Keogh, Chu, Hart, & Pazzani, 2004). Alternatively, top-down and bottom-up approaches or combinations thereof can be used (Keogh, Chu, et al., 2004). The window approach adds another parameter, the window size w, to the algorithm. Preselecting an appropriate window size is often considered as feasible (Keogh, Lonardi, & Ratanamahatana, 2004). In segmentation approaches, the MinE criterion can be applied to not only choose an appropriate embedding dimension but also to determine the combination of embedding dimension and window size that maximizes the distinctiveness of the order pattern representation.

As a further potential modification of PDC, we can adopt other divergences between probability distributions. Csiszár f-divergences (Csizár, 1974), sometimes also Ali–Silvey divergences (Ali & Silvey, 1966), describe another family of divergences that also subsume the Kullback-Leibler-divergence as special case. Recently, Cichocki, Cruces, and Amari (2011) presented a further generalization of α-divergences to the broad class of α-β-γ-divergences. These might provide further interesting measures of dissimilarity for future research.

One could argue that, in some cases, PDC is not an appropriate representation of the time series. For example, if the moments of the time series are thought to represent discriminative information about time series, this information is discarded by PDC. Although this can be advantageous for some problems, it might be problematic for others. A famous example is the synthetic control chart data set, which was described by Alcock and Manolopoulos (1999). They describe a task that needs the discrimination between six noisy patterns: The six basic patterns are a constant, a cyclic pattern, an increasing trend, a decreasing trend, a sudden upward shift, and a sudden downward shift. The final patterns are obtained by adding white noise to the basic patterns. PDC can in principle succeed in determining the constant, the cycle, and the trends. However, both patterns with a sudden shift in either direction are not captured by PDC. In domains where such a shift is considered as irrelevant information, for example arising from a sensor misreading, this property is clearly beneficial. Whenever information of the moments should be preserved, there is no reason that speaks against taking into account multiple features for clustering that capture the specific definition of similarity in the idiosyncratic context and extending the dissimilarity function to also consider further features, such as the signal's mean and the variance. Wang, Smith, Hyndman, and Alahakoon (2004) present such a feature-based approach, which also included complexity measures, even though they did not use the permutation distribution. It would be highly interesting to examine the extent to which PDC can improve their cluster results.

Many time series approaches are based on a wavelet decomposition and perform clustering in the frequency space. Again, complexity measures like the PD could be applied to the different resolution levels of such basis transformations.

Hierarchical clustering as used in this work is relatively slow due to its quadratic time complexity. On the other hand, codebook generation from the time series and the calculation of the pair-wise distances is highly parallelizable. When the number of clusters is known beforehand, it might be advantageous to switch to more efficient clustering methods like k-means or spectral clustering while being aware of their particular limitations.

In the next chapter, I turn the focus of my thesis to SEM Trees, which combine SEMs and decision trees to a novel method of exploratory data analysis.

3
Structural Equation Model Trees

SEM Trees are placed in a quite different setting than the clustering approach. In that context, we assumed that neither a model about the observed data nor any additional information is available. However, this is often not the case. Indeed, researchers often hold a large set of prior assumptions about their data and have models at their disposal that formalize their preconceptions. It is interesting to observe how these initial models are refined when they do not perform as expected, e.g., when they explain to little variance in the data or simply indicate by measures of goodness-of-fit that the model is an inappropriate representation of the sample. In these cases, exploratory approaches can be taken, i.e., approaches that are data-driven and extend models so as to account for the observed phenomena. However, these model selection processes bear the risk of implicit and explicit selection biases. In these cases, model refinements represent mere idiosyncrasies of the sample rather than relations that generalize to the population. Whenever a SEM is available along with a set of covariates whose influence on the model is not yet clear, SEM offers a formal approach to find covariates that find partitions of the data set that maximally differ with respect to the estimated parameters. SEM Trees are a method to systematically explore refinements of initial models in order to guide informed theory building.

SEM Trees are a consolidation of Structural Equation Models (SEMs) and decision trees. In the following, a review of the most important concepts of SEM is given. Afterwards, the SEM concept is united with the decision tree paradigm, resulting in SEM Trees. Important theoretical and practical aspects of SEM Trees are discussed. Applications of SEM Trees are presented in Chapter 5.

3.1 Structural Equation Modeling

Structural Equation Modeling is a frequently used multivariate analysis technique in the behavioral and social sciences. SEMs have been discussed extensively in the literature. For an introduction, see, for example, the textbook by Bollen (1989). SEMs constitute a widely applied

multivariate analysis technique and can be conceived of as a unification of several multivariate statistics (cf. Fan, 1997).

In SEM, models are constructed by specifying relations between observed and latent variables. Under the assumption of multivariate normality of the model variables, the resulting distribution is also multivariate normal. Therefore, the model can be fully described by a model-implied expectations vector $\mu(\theta)$ and a model-implied covariance matrix $\Sigma(\theta)$, with θ being a vector of free parameters. By imposing or freeing restrictions on the model variables' distributions or the relations between the variables, different concepts and hypotheses about the data can be integrated into a model. By minimizing a discrepancy function between the model-implied covariance matrix and expectation vector, and a sample covariance matrix and a sample expectation vector, population parameters can be estimated from a finite sample, and goodness-of-fit indices can be derived.

The original SEM framework only considered analyses of covariance structures and assumed the expected values to be zero. In the following, when we talk about SEM, we consider the extension which can be found in Sörbom and Jöreskog (1982), which also considers first-order moments of the data.

There is a variety of software for modeling SEMs available. Among the most prominent representatives are LISREL (Jöreskog & Sörbom, 1996), Mplus (L. Muthen & Muthen, 2007), SAS (SAS, 1999), Mx (Neale, Boker, Xie, & Maes, 1999) and recently the open source framework OpenMx (Boker et al., 2011).

In the following, I will recapitulate a graphical and a matrix-based representation of SEMs. Based on this covariance algebra, I repeat maximum likelihood estimation of model parameters and review measures of goodness of fit that are found in the literature. Finally, basic types of SEM are briefly presented.

3.1.1 Latent Variable Models

SEM are a class of latent variable models. Latent variable models supplement models of observed variables by introducing latent variables, which are not measured directly. The observed variables are sometimes referred to as manifest variables when interpreted as manifestations of a latent construct. Latent variables are sometimes also called hidden variables. By imposing structural restrictions about the relation of the latent and observed variables, we can both estimate the relations between the variables and infer distributions and individual values of the latent variables. In the machine learning literature, presumably the most prominent latent variable model is the Bayesian Network or Belief Network (cf. Russel & Norvig, 2003). SEM and Bayesian Networks differ in a central aspect. Latent variables in a SEM represent linear combinations of random variables, whereas Bayesian Networks represent conditional probability distributions. This allows Bayesian Networks to factorize complex joint distributions of observed variables into a smaller network of conditional distributions, which again makes fast inference feasible. SEM describe linear combinations of normally distributed variables. In the machine learning community, this class of models is also referred to as *linear Gaussian model*s, which subsumes a broad range of models with idiosyncratic estimation techniques, e.g., Principal Component Analysis, Hidden Markov Models, or the Kalman Filter (Roweis & Ghahramani, 1999).

In Structural Equation Modeling, a researcher is typically interested in retrieving the un-

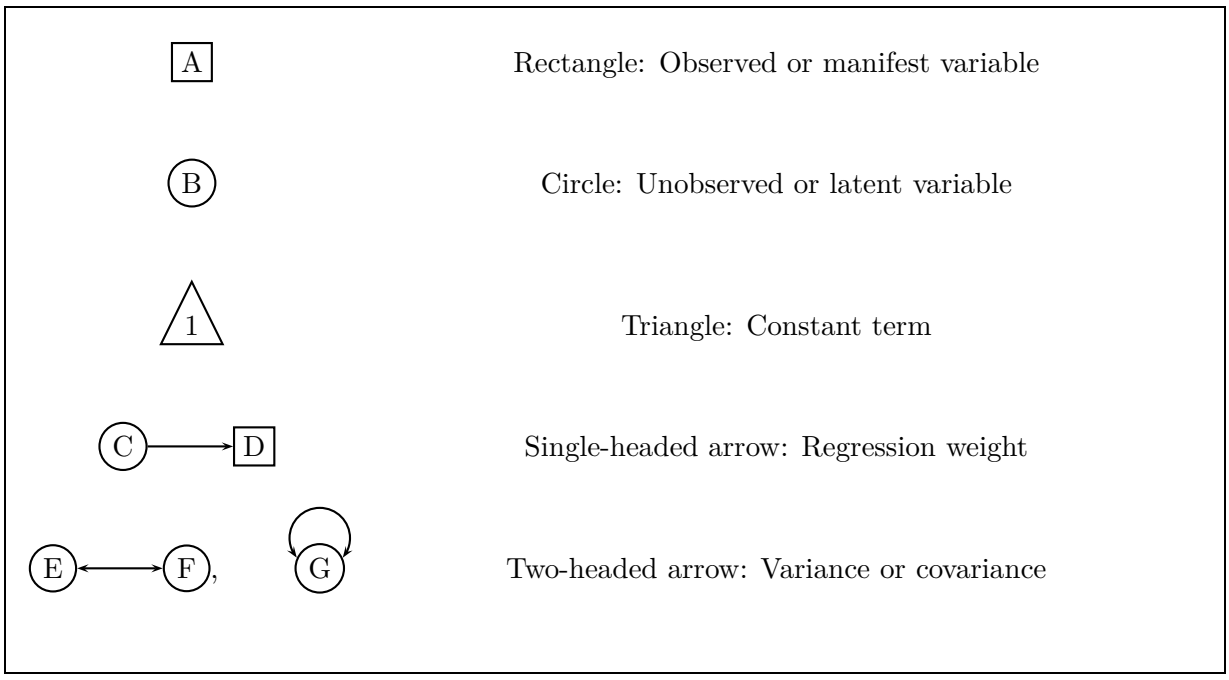


Figure 3.1.1: Primitive elements of the graphical representation of Structural Equation Models.

conditional distributions of hypothesized latent constructs. For example, intelligence can be modeled as a latent construct that cannot be measured directly. Rather, it is defined as the common ability to succeed on certain tests that are designed to measure cognitive capability. Modeling the relation between test outcomes observed for each participant and a common latent factor, a SEM can indicate to what extent each test score is predicted by a single common construct, how much residual variance remains unique to this test, and how much the score of this common factor varies across the population.

The introduction of latent variables has several advantages. Most obviously, latent variables allow the formulation of hypothetical constructs that cannot be measured directly, and allows hypothesis tests on these. Equally importantly, the introduction of measurement error terms as latent constructs allows the observed variables to be purged of this measurement error. This adjustment offers greater validity and generalizability of research designs (Little, Lindenberger, & Nesselroade, 1999).

3.1.2 Representation of SEM

SEMs are models of linear relations between latent and observed variables. SEMs are typically defined in a matrix-style notation and often visually represented as graphical models. There are different ways to describe a SEM. In this thesis, I adhere to the Reticular Action Model (RAM) notation (Boker, McArdle, & Neale, 2002; McArdle & Boker, 1990; McArdle & McDonald, 1984), which is used, for example, in Mx (Neale et al., 1999) and OpenMx (Boker et al., 2011).

A central aspect is common to all formal representations of SEM. The observed variables are defined to arise as a linear combination of observed and latent variables:

Definition 40. Let Y be a distribution over l model variables of a SEM defined by the covari-

ance matrix $S \in \mathbb{R}^{l \times l}$ and vector of expectations $m \in \mathbb{R}^l$. Let $A \in \mathbb{R}^{l \times l}$ be the matrix describing the linear relations between the model variables. The model-implied distribution $X \in \mathbb{R}^{l \times l}$ is the distribution satisfying $X = Y + AX$. This definition is equivalent to $X = (I - A)^{-1}Y$.

In the following, a description of the graphical notation and the matrix notation is given. SEMs are typically visualized as graphs with nodes being observed and latent variables, and edges, typically called arrows, representing different kinds of relations between variables. In particular, observed variables, also referred to as manifest variables, are depicted as rectangular boxes with the variable name inside the box. Unobserved or latent variables are depicted in circles. Regression weights between the variables are depicted as single-headed arrows. Variances and covariances are depicted as two-headed arrows. Particularly, a two-headed arrow with a single variable as both source and target depicts a variance, whereas a two-headed arrow connecting two different variables depicts a covariance. Arrows without labels are assumed to have unit weight. Otherwise, they are labeled with parameter symbols if freely estimated in the model, or with real values if they are fixed at that value. RAM introduces a third type of variable, the triangle, represents a constant. In the literature, the triangle is sometimes depicted with a double-headed arrow. Regression weights from a triangle to a variable determine expectations of the variable. In addition to the definition of variance and covariance structures, regression weights from the triangle to latent or manifest variables represent expected values of the respective target variables. An overview of the symbols can be found in Figure 3.1.1. Graphical representation of various exemplary models are illustrated in Section 3.2.

Definition 40 shows that the model-implied distribution of the observed variables in a SEM is essentially determined by a linear combination of the model variables. Under the assumption of multivariate normality of Y, the distribution X can be described by a vector of expectations and a covariance matrix. In the following, we denote with X^T the transpose of a matrix X. The observed model-implied vector of expectations and covariance matrix arise from RAM in the following way:

Theorem 41. Let a SEM with l variables, among those p observed variables, be defined by a structural matrix $A \in \mathbb{R}^{l \times l}$ describing linear relations between the variables, a covariance matrix $S \in \mathbb{R}^{l \times l}$ of the variables, a vector of expectations of the variable $m \in \mathbb{R}^l$, and a filter matrix $F \in \mathbb{R}^{p \times l}$ with ones on the diagonal entries. The model-implied observed covariance matrix $\Sigma \in R^{p \times p}$ is

$$\Sigma = F\,(I_l - A)^{-1}\,S\,(I_l - A)^{-T}\,F^T$$

and the model-implied vector of expectations $\mu \in \mathbb{R}^p$ is

$$\mu = F\,(I_t - A)^{-1}\,m$$

Proof. Following definition 40, we obtain the distribution that is described by the SEM as $X = (I - A)^{-1}Y$ with Y being the distribution over all model variables. By definition, the covariance of Y is defined by matrix $S \in R^{l \times l}$ and thus, using covariance algebra (see also Appendix A) the covariance of X is

$$\begin{aligned} Cov(X) &= (I_l - A)^{-1} Cov(Y) (I_l - A)^{-T} \\ &= (I_l - A)^{-1} S (I_l - A)^{-T} \end{aligned}$$

The implied covariance matrix $Cov(X)$ includes variances and covariances between latent and observed variables. Since we can observe only the covariance of observed variables in a sample from the population, S' has to be filtered so that the remaining matrix contains only the covariances of the observed variables. This filter matrix $F \in R^{p \times l}$ will have ones on the p diagonal entries, so that a multiplication of F with a vector $Fv = v'$ or a matrix $M' = FMF'$ representing latent and manifest variables will be filtered to be a vector of size p or a matrix of size $p \times p$ with rows and columns corresponding to the observed variables.

Therefore, the model-implied covariance matrix of the observed variables $\Sigma \in R^{p \times p}$ is given as

$$\Sigma = F (I_l - A)^{-1} S (I_l - A)^{-T} F^T \tag{3.1.1}$$

Analogously, it is easy to see that the model-implied vector of observed model expectations $\mu \in R^p$ is

$$\mu = F (I_t - A)^{-1} m$$

□

We have seen that the matrix-style notation of RAM uses three matrices to describe a SEM: A covariance matrix S ("symmetric relations"), a structural matrix A ("asymmetric relations"), and a filter matrix F which selects a sub-matrix from the model-implied matrix, such that the resulting matrix only contains rows and columns corresponding to observed variables. The matrices correspond closely to the previously described graph representation. Let a SEM have a total of l variables, among these p observed variables, the others being observed. The matrix A contains all asymmetric relations, i.e., all single-headed arrows, connecting both latent and manifest variables with each other. The symmetric matrix S contains all variances and covariances, i.e., all two-headed arrows.

3.1.3 Maximum Likelihood Estimation

Once data have been collected, researchers typically want to make inferences about the population they sampled from. Given a model that reflects their prior assumptions, they expect to estimate parameters in their models so that this model reflects the observations as closely as possible. In the SEM setting, this means that a set of parameters that minimizes the discrepancy between the model-implied covariance matrix and the sample covariance matrix is sought. At the same time, if expectations are modeled, the discrepancy between the model-implied expectations and the sample expectations have to be minimized. A convenient method for estimating parameters was proposed by Fisher (1922, 1925): Maximum Likelihood (ML) Estimation chooses parameters to maximize the likelihood of having observed the sample under the model. This method of estimation has many desirable properties, among these consistency (the parameter estimates are asymptotically unbiased) and efficiency (lowest possible variance

of parameter estimates). In order to estimate parameters in the model, the likelihood function of the sample, given the model, is established and parameters are chosen such that this function attains a maximum. In the following, the derivation of the Maximum Likelihood Estimation procedure for SEM is reiterated. Similar derivations can be found in SEM textbooks, e.g., in Bollen (1989) or in any other statistics textbook that introduces ML estimation for multivariate normal distributions. Typically, we will talk about covariance matrices and expectations vectors that are parameterized by a parameter vector θ. Formally, these are referred to as $\Sigma(\theta)$ and $\mu(\theta)$. Whenever it is unambiguous, we will use the shorthand notation Σ and μ for parameterized covariance matrices and expectation vectors.

For reasons of simplicity and computational efficiency, the negative log-likelihood is typically used instead of the likelihood function. Both functions have the same extrema. However, note that the negative log-likelihood function must be minimized whenever the likelihood function is subject to maximization.

Theorem 42. *Let M be a SEM with model-implied covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$ and model-implied vector of expectations $\mu \in \mathbb{R}^p$. The negative two log-likelihood of observing a set of independently drawn observations $x = \{x_i | x_i \in \mathbb{R}^p\}$ is*

$$-2LL(x_1, \ldots, x_n|\mu, \Sigma) = N \cdot p \log(2\pi) + N \log|\Sigma| + \sum_{i=1}^{N} \left[(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)\right]$$

Proof. Assuming that observed data is multivariate normal, the likelihood of observing a single observation $x \in R^p$ is given by the multivariate Gaussian probability distribution function:

$$L(x|\mu, \Sigma) = ((2\pi)^p |\Sigma|)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right]$$

To simplify the term, the log-likelihood is taken, which has the same maxima since the logarithm is a monotonic transformation

$$LL(x|\mu,\Sigma) = -\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)$$

The negative two log-likelihood is thus

$$-2LL(x|\mu, \Sigma) = p \log(2\pi) + \log|\Sigma| + (x - \mu)^T \Sigma^{-1} (x - \mu) \tag{3.1.2}$$

Observing multiple observations, assuming they are drawn independently and identically distributed, the product of the individual likelihoods and thus the sum over the $-2LL$s is

$$
\begin{aligned}
-2LL\left(x_1,\ldots,x_n|\mu,\Sigma\right) &= -2\log\left(\prod_{i=1}^{N} L\left(x|\mu,\Sigma\right)\right) && (3.1.3)\\
&= \sum_{i=1}^{N} -2LL\left(x|\mu,\Sigma\right) && (3.1.4)\\
&\quad \sum_{i=1}^{N}\left[p\log\left(2\pi\right)+\log\left|\Sigma\right|+\left(x_i-\mu\right)^T\Sigma^{-1}\left(x_i-\mu\right)\right] && (3.1.5)\\
&= N\cdot p\log\left(2\pi\right)+N\log\left|\Sigma\right| && (3.1.6)\\
&\quad +\sum_{i=1}^{N}\left[\left(x_i-\mu\right)^T\Sigma^{-1}\left(x_i-\mu\right)\right] && (3.1.7)
\end{aligned}
$$

Let S be the sample covariance matrix, obtained from the N observations x_i as □

$$S=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-m\right)\left(x_i-m\right)^T$$

This allows a reformulation of the negative two log-likelihood function in a condensed form, which is efficient to compute.

Theorem 43. *Let Σ be the model-implied covariance matrix and S be the empirical covariance matrix. If the true mean is known or assumed to be zero, the maximum likelihood fit function for parameter estimation simplifies to*

$$F_{ML}\left(\Sigma,S\right)=\log\left|\Sigma\right|+tr\left(\Sigma^{-1}S\right)$$

Proof. If the true mean is known, $\mu=m$, most often this is the case when the mean is assumed to be zero and is not modeled explicitly, the following trace trick, which is found in many textbooks, allows a simplification of the log-likelihood function

$$
\begin{aligned}
\textstyle\sum_{i=1}^{N}\left(x_i-\mu\right)^T\Sigma^{-1}\left(x_i-\mu\right) &= \textstyle\sum_{i=1}^{N} tr\left[\left(x_i-\mu\right)^T\Sigma^{-1}\left(x_i-\mu\right)\right]\\
&= \textstyle\sum_{i=1}^{N} tr\left[\Sigma^{-1}\left(x_i-\mu\right)\left(x_i-\mu\right)^T\right]\\
&= tr\left[\Sigma^{-1}\textstyle\sum_{i=1}^{N}\left(x_i-\mu\right)\left(x_i-\mu\right)^T\right]\\
&= tr\left(\Sigma^{-1}N\cdot S\right)\\
&= N\cdot tr\left(\Sigma^{-1}S\right)
\end{aligned}
$$

This simplification can be used to get rid of the summation and yields

$$-2LL\left(x_1,\ldots,x_n|\mu,\Sigma\right)=N\left[p\log\left(2\pi\right)+\log\left|\Sigma\right|+tr\left(\Sigma^{-1}S\right)\right]$$

Since N and $p\log(2\pi)$ do not depend on the free parameters, the function that has to be minimized is

$$F_{ML}=\log\left|\Sigma\right|+tr\left(\Sigma^{-1}S\right)$$

□

Theorem 44. *Let Σ be the model-implied covariance matrix, μ the model-implied expectations vector, S the empirical covariance matrix and m the empirical expectations vector. The fit function that has to be minimized to obtain a maximum-likelihood estimate is*

$$F_{ML}(\mu, \Sigma, m, S) = N\left[\log|\Sigma| + tr\left(\Sigma^{-1}S\right) + (m-\mu)^T\Sigma^{-1}(m-\mu)\right] \tag{3.1.8}$$

Proof. A variation of the trace trick is employed to obtain this derivation

$$\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) = \sum_{i=1}^{N}(x_i-m+m-\mu)^T\Sigma^{-1}(x_i-m+m-\mu)$$

With the binomial expansion we obtain the terms

$$\begin{aligned}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) &= \sum_{i=1}^{N}(x_i-m)^T\Sigma^{-1}(x_i-m) + \sum_{i=1}^{N}(x_i-m)^T\Sigma^{-1}(m-\mu)\\ &\quad + \sum_{i=1}^{N}(m-\mu)^T\Sigma^{-1}(x_i-m) + \sum_{i=1}^{N}(m-\mu)^T\Sigma^{-1}(m-\mu)\end{aligned}$$

The first term of this Equation can be related to the empirical covariance matrix

$$\begin{aligned}\sum_{i=1}^{N}(x_i-m)^T\Sigma^{-1}(x_i-m) &= \sum_{i=1}^{N}tr\left((x_i-m)^T\Sigma^{-1}(x_i-m)\right)\\ &= \sum_{i=1}^{N}tr\left((x_i-m)(x_i-m)^T\Sigma^{-1}\right)\\ &= N\cdot tr\left(S\Sigma^{-1}\right)\end{aligned}$$

The second term cancels out in the following way

$$\begin{aligned}\sum_{i=1}^{N}(x_i-m)^T\Sigma^{-1}(m-\mu) &= tr\left(\sum_{i=1}^{N}(m-\mu)(x_i-m)^T\Sigma^{-1}\right)\\ &= tr\left((m-\mu)\left(\sum_{i=1}^{N}(x_i-m)^T\right)\Sigma^{-1}\right)\\ &= tr\left((m-\mu)\cdot 0\cdot\Sigma^{-1}\right)\\ &= 0\end{aligned}$$

Analogously, the third term cancels out. The fourth term can be simplified by replacing the sum with a multiplication of N. This yields

$$\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) = N\cdot tr\left(S\Sigma^{-1}\right) + N\cdot\sum_{i=1}^{N}(m-\mu)^T\Sigma^{-1}(m-\mu)$$

We obtain

$$-2LL\left(x_1,\ldots,x_n|\mu,\Sigma\right) = N\left[p\log\left(2\pi\right)+\log|\Sigma|+tr\left(\Sigma^{-1}S\right)+\left(m-\mu\right)^T\Sigma^{-1}\left(m-\mu\right)\right]$$

Finally, we can drop the constant term that does not depend on the free parameters. □

Given a SEM M, I introduce a further simplification in notation. Whenever the likelihood of a model M is referred to, we simply write

$$-2LL\left(x_1,\ldots,x_N|M\right) = -2LL\left(x_1,\ldots,x_N|\mu,\Sigma\right)$$

Parameter estimates are obtained by finding the global maximum of the likelihood function, or the equivalent minimum of the respective simplified F_{ML} functions. Since it is generally difficult to find a closed-form solution for arbitrary models, numeric procedures are employed to find the maximum of the likelihood function. Commonly, SEM optimizers are gradient-based. The vector of the first partial derivatives of the likelihood function with respect to the model parameters is numerically estimated. Parameter estimates are then iteratively changed until a maximum of the likelihood function is found. An improvement of this approach also considers the matrix of the partial second derivatives, the so-called *Hessian matrix*. This class of methods is known as Newton methods. If the Hessian matrix is iteratively approximated rather than fully approximated in each iteration, this is called a quasi-Newton method. A famous representative of the later method is the *Broyden, Fletcher, Goldfarb and Shanno method* (BFGS), which was independently suggested by the four authors. A comprehensive overview of optimization algorithms is given by Fletcher (1994). BFGS is usually considered the most efficient unconstrained optimizer and is, for example, used by OpenMx to obtain maximum-likelihood estimates for SEMs. Von Oertzen, Ghisletta and Lindenberger (2009) report that a dampened Newton method works well in practice.

Finding the right starting values is crucial for the optimization process. Poor choices of these values can require a large number of iterations until the optimizer converges on the maximum. Some researchers prefer choosing starting values by hand. In empirical research situations, appropriate values are often not known a-priori; researchers may feel reluctant to feed an algorithm with prior values, and they may not be aware of the mechanisms of the underlying optimization process. Therefore, it is common to choose starting values for the ML optimization by unweighted least-squares estimates (ULS) that are obtained from a preceding optimization run (Jöreskog & Sörbom, 1982; von Oertzen et al., 2009). Other techniques include repeated model fits with perturbations of some randomly chosen starting values, a technique which can be found in MPlus (L. Muthen & Muthen, 2007).

3.1.4 Goodness-of-Fit and Information Indices

SEMs come with a large body of literature concerning the evaluation of goodness-of-fit. Cudeck and Henly (1991) proposed a general framework that separates the misfit of a hypothesized model to the unknown true model of the population. Essentially, the relation of four measures is described: (1) the true population covariance matrix Σ, (2) the hypothesized model with the unknown and true parameter γ given as $\Sigma\left(\gamma\right)$, (3) the estimated parameters from the observations $\Sigma\left(\hat{\gamma}\right)$, and (4) the sample covariance matrix S. Many traditional fit indices, of

which some are listed below, define measures of the *sample discrepancy* which is defined as a discrepancy between S and $\Sigma(\hat{\gamma})$. Within this framework, the *overall discrepancy* is defined as the discrepancy between the estimated model $\Sigma(\hat{\gamma})$ and the true model Σ. This can be partitioned into the *discrepancy of estimation* and the *discrepancy of approximation*. The former is the discrepancy between $\Sigma(\gamma)$ and $\Sigma(\hat{\gamma})$ that is due to the sample fluctuation and should decrease to zero with the number of observations going to infinity. The latter represents the discrepancy between the true model Σ and the hypothetical model with the optimal parameter $\Sigma(\gamma)$ that might be due to model misspecification and relationships that are not appropriately captured by a parsimonious model. It is advisable to use a set of conceptually different fit indices to obtain formal measures of model quality (Hu & Bentler, 1999; MacCallum, Browne, & Sugawara, 1996). Special attention has to be paid if only indices of sample discrepancy are used since they can easily lead to an overfitting to the sample.

In the following, we will emphasize the use of cross-validation techniques to evaluate the predictive quality of models. Only with such techniques can we guarantee that the model under investigation indeed has predictive performance beyond the collected data set.

A huge variety of fit indices can be found in the SEM literature, and opinions of what should determine a good fit also diverge. Most often, researchers have heuristically adopted thresholds and rules-of-thumb to decide whether a model sufficiently fits the data (Hu & Bentler, 1999). In the following, a brief overview on model fit indices and rules-of-thumb to determine fitness is given.

Overall model fit is determined with the *Root Mean Square Error of Approximation* (RMSEA) and *Standardized Root Mean Square Residual* (SRMR). The *Comparative Fit Index* (CFI) and *Non-Normed Fit Index* (NNFI) are incremental fit indices which compare the fit of an *independence model* to the candidate model. The independence model assumes all covariances between latent constructs are zero. Comparative fit indices indicate that there is a correlational structure that differs from the independence model. In contrast to relative fit indices, SRMR and the χ^2-index are absolute fit indices, measuring sample discrepancy. The last family of fit criteria, such as *Akaike Information Criterion* (AIC) and *Bayesian Information Criterion* (BIC), introduce penalty terms for model complexity and aim at estimating overall discrepancy.

A brief review of the most common fit indices is repeated in the following.

- χ^2 : Non-significant p-values of the χ^2-distribution are interpreted as non-significant differences between the model-implied covariance matrix and the sample covariance matrix and can serve as a tentative indicator for good model fit (Bollen, 1989). This statistic is dependent on sample size, and moderate sample sizes will detect small differences as significant deviations. Generally, this measure is considered inexpedient and has led to the development of a large variety of other measures.

- SRMR: The residual matrix is defined as the difference between the sample covariance matrix and the model-implied covariance matrix. Large residual values indicate a bad representation of the respective part of the model. Therefore, an average of the residuals can serve as an indicator of goodness of fit to the sample data. In order to compare this value across data sets and models, a standardization is required. The resulting fit index is called the *Standardized Root Mean Square Residual* (SRMR)

$$SRMR = \sqrt{\frac{\sum_{i=1}^{p}\sum_{j=1}^{p}\frac{(\sigma_{ij}-s_{ij})^2}{\sqrt{\sigma_{ii}s_{jj}}}}{(p+q)(p+q+1)}}$$

A SRMR of zero indicates a perfect model fit, a SRMR < 0.05 is interpreted as good model fit; and $0.05 <$ SRMR < 0.10 is believed as moderate model fit.

- CFI: The *Comparative Fit Index* also known as the *Bentler Comparative Fit Index* compares the model with an independence model (Bentler, 1990). The CFI measures the relative lack of fitness when going from the independence model with a diagonal covariance matrix to the proposed model. Thus, a CFI of .95 indicates a loss of 5% of model fit when postulating the restrictions made by the researcher's model in comparison to a pure measurement model. Generally, a CFI of .90 or greater indicates acceptable model fit.

$$CFI = 1 - \frac{\chi^2 - df}{\chi_i^2 - df_i} \tag{3.1.9}$$

- NNFI: *The Non-Normed Fit Index* (NNFI) (Tucker & Lewis, 1973; Bentler & Bonett, 1980) is also known as Tucker-Lewis Index (TLI) and is

$$NNFI = \left(\frac{\chi_b^2}{df_b} - \frac{\chi^2}{df}\right) \cdot \left(\frac{\chi_b^2}{df_b} - 1\right)^{-1}$$

with χ_b^2 being the fit index of a corresponding null model with df_b degrees of freedom. Generally, a perfect model fit is a NNFI of 1.0 and good model fit is attributed to models with NNFI > 0.97. A range of 0.95 to 0.97 is still considered an adequate model fit. The NNFI can exceed values of 1.0, which is regarded as an indicator of overfitting due to too many parameters in the model.

- RMSEA: The *Root Mean Squared Error of Approximation* is an estimator of the error of approximation as introduced by Steiger and Lind (1980). Under misspecification, the χ^2-test statistic is asymptotically non-central χ^2-distributed. The RMSEA estimates this non-centrality parameter by $\chi^2 - df$ with df being the degrees of freedom of the model.

$$\text{RMSEA} = \sqrt{\max\left(\frac{(\chi^2 - df)}{df\,(N-1)}, 0\right)} \tag{3.1.10}$$

A value below 0.05 is commonly interpreted as adequate fit, a value between 0.05 and 0.08 is considered a mediocre fit, and everything else a bad model fit. An advantage of the RMSEA over other measures of goodness-of-fit is that its asymptotic distribution is known. Therefore, confidence intervals can be given based on the rescaled approximate non-central χ^2-distribution. Beyond using a point-estimate to judge goodness-of-fit, a hypothesis test of close fit can be applied: model fitness is judged adequate if the lower confidence interval of RMSEA is below 0.05 (MacCallum & Austin, 2000).

- AIC: The *Akaike Information Criterion* (Akaike, 1974) is a criterion which is broadly applied according to the literature. AIC adds a penalty to the likelihood function that

more parsimonious models are favored, in the hope that overfitting is avoided and the generalizability of the model is increased. The rationale of the AIC is deeply rooted in information theory. It is an estimator of the Kullback-Leibler divergence between the true model and the estimated model (Burnham & Anderson, 2002). The resulting penalty term for model complexity is twice the number of free parameters *df* in the model. In contrast to most other indices, AIC allows the comparison of non-nested models. The model with the lowest AIC should generally be favored.

$$AIC = \chi^2 + 2 \cdot df$$

- BIC: The *Bayesian Information Criterion* or *Schwarz Criterion* (Schwarz, 1978). It defines an increasing function of the model complexity determined by the number of free parameters *df* and the number of training cases N. The BIC generally penalizes free parameters more strongly, depending on sample size.

$$BIC = \chi^2 + df \cdot \log(N)$$

Not surprisingly, Fan, Thompson, and Wang (1999) found that different fit indices react very differently to different types of model misspecification. Also, researchers can be led to very different conclusions about the adequacy of their model depending on their choice of fit index. Most commonly, the combination of RMSEA, CFI, and NNFI are advised (Hu & Bentler, 1999; MacCallum et al., 1996). However, the sheer number of different indices poses a danger in itself since researchers could unluckily adjust their models until some measure of fitness reaches the required level, whereas others could be neglected. Unwittingly, a strongly contradicting index may thus remain overlooked. This situation can lead to adverse selection biases and overfitting. Generally, I strongly suggest that model selection should be grounded on the predictive ability of a model, that is, on the evaluation of the model on an independent sample or, if this is not available, on a performance estimation based on cross-validation (Stone, 1974). The discussion of evaluation methods will be continued in the context of SEM Trees in Section 3.5.

3.1.5 Hypothesis Testing in SEM: Likelihood Ratio Test

A particular strength of the SEM framework is the availability of hypothesis tests to test parameter restrictions in the model. A versatile tool for hypothesis testing is the likelihood ratio test. The test indicates whether an algebraic parameter restriction in a model significantly decreases the likelihood of the data under the model. Intuitively, if a parameter restriction in a model, e.g., implemented by fixing a parameter to zero, reflects the true value in the population, the decrease of fit is only marginal. Asymptotically, there is no decrease of fit at all, since with growing sample size, a freely estimated parameter approaches the true value. However, since we can observe only finite samples from the population, we expect a maximum of the likelihood function somewhere in close vicinity to the true value. Therefore, fixing a parameter on the true value in the population will lead to a slight discrepancy. Following the same logic, if a restriction does not reflect the true population value, we expect a large misfit and therefore a much lower likelihood, respectively a much larger negative two log-likelihood value. Thus we have to decide on a threshold that discriminates between low values of the test statistic arising from finite sample differences and high values of the test statistic which are due to inappropriate

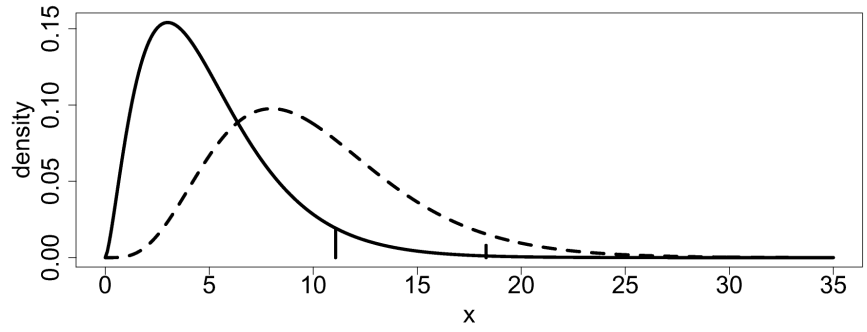


Figure 3.1.2: A χ^2-distribution with 5 degrees of freedom (solid line) and a χ^2-distribution with 10 degrees of freedom (dashed line). Vertical lines indicate critical values obtained from a significance level of $\alpha = 0.05$. In a classical hypothesis testing setting with a null hypothesis that is known to be χ^2-distributed, the null hypothesis is rejected if the observed statistic is larger than the critical value. Following the logic that a statistic that extreme is unlikely to be observed under the null hypothesis, the alternative hypothesis is accepted.

restrictions. Luckily, there is a fundamental theorem about nested models in statistics. A nested model is a model that can be derived from the baseline model by adding algebraic constraints, e.g., by demanding equality of pairs of parameters or by demanding equality of parameters and a constant. Each such constraint reduces the degrees of freedoms of the model by one. Wilks's (1938) theorem about nested models was used in Chapter 2 to determine the number of clusters in PDC. The theorem states that the log-likelihood ratio or, equivalently, the difference between the negative two log-likelihoods for nested models is asymptotically χ^2-distributed under the null-hypothesis claiming that the restriction in the nested model holds in the population. Let M be a SEM and x a set of observations. The log-likelihood ratio statistic is

$$LLR\left(x\right) = -2\log\left(\frac{L\left(x|M\left(\hat{\theta}_R\right)\right)}{L\left(x|M\left(\hat{\theta}_B\right)\right)}\right) \sim \chi^2$$

with $\hat{\theta}_R$ and $\hat{\theta}_B$ being maximum likelihood estimates, the former under a restriction and the latter without a restriction. An exemplary χ^2-distribution is depicted in Figure 3.1.2. Knowing the distribution allows the application of a hypothesis test to reject the null hypothesis that the restriction holds in the population. The null hypothesis can be formulated as

$$H_0 : \theta_R = \theta_B$$

where θ_R is the unknown true parameter for model M under the restriction and θ_B is the unknown optimal parameter for the model M without the restriction. In significance testing, two types of errors are typically discerned that are associated with the decision of rejecting the null hypothesis or not. Rejecting the null hypothesis despite its being true is called a *type-I error*. In contrast, no rejection of a false null hypothesis is referred to as a *type-II error*.

Typically, the type-I error rate α, i.e., the probability of falsely rejecting the null hypothesis, is chosen a priori by the researcher. By convention, this value is most often chosen to be either 0.05, 0.01, or 0.001. Under the assumption of fixed a-priori probabilities for the hypotheses, choosing α will effectively control the trade-off between type-I error and type-II error. Smaller values of α trade increasing type-II errors against decreasing type-I errors, and vice versa.

Given a choice of α, a *critical value* that serves as a threshold for the observed test statistic can be determined, allowing a decision whether or not to reject the hypothesis. This is exemplarily illustrated in Figure 3.1.2. The critical value is obtained by finding the value x for which the probability of drawing this value or a larger value from the given distribution is equal to the chosen α level. Formally, given a probability density function $P(X)$, we determine x such that $P(X \geq x) = \alpha$. Whenever the observed test statistic is larger than the critical value, the null hypothesis is rejected. This procedure allows comparison of nested models and enables important tests whether regression weights or expectations are different from zero and should effectively be included in the model. This test will play an essential role in the model comparison involved in the algorithm for generating SEM Trees.

3.1.6 Distribution of Parameter Estimates

In a SEM, parameter estimates $\hat{\theta}$ are approximately normally distributed in the neighborhood of the true and unknown parameter θ with the distribution $\mathcal{N}(0, H^{-1})$, with H being the Hessian matrix of θ, containing the second-order partial derivatives of the negative two log-likelihood function (Bollen, 1989)

$$H^{-1}(\theta) = \left(\frac{\partial^2 F_{-2LL}}{\partial \theta^2} \right)^{-1}$$

This matrix is usually directly available from the numerical optimization procedure that was used to obtain the ML estimates. The square roots of the trace of the Hessian matrix are the respective estimators of the standard errors of the parameter estimates:

$$se(\theta) = \sqrt{tr\left(H^{-1}(\theta)\right)}$$

It is common practice to mark parameter estimates with asterisks if they are significantly different from zero. This property is commonly tested with a Z-test, a test that approximates a test statistic with a standard normal distribution. We formulate a null hypothesis that the estimate of the i-th parameter $\hat{\theta}_i$ has the mean zero $m = 0$. The z-statistic under the null-hypothesis is assumed to be normally distributed and is given as

$$z = (\theta_i - m) / se(\theta_i)$$

Based on the quantiles of the standard normal distribution, we obtain the critical values for a one-sided significance tests as $crit_{0.01} = 2.33$ for a significance level of $\alpha = 0.01$ and $crit_{0.001} = 3.09$ for a level of $\alpha = 0.001$. If the z-statistic for a parameter is significant on the former level, it will be marked with a single asterisk ($*$). If it is significant on the later level, it will be marked with a double asterisk ($**$). It should be noted that this type of indication bears the potential for misunderstandings, among these the lacking correction for the issue of multiple testing. The question arises why a continuous measure should be reduced to arbitrary categories of significance. I suggest using the explicit standard errors instead of the asterisks.

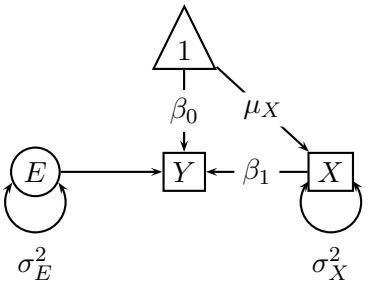


Figure 3.2.1: Univariate regression in a graphical representation. Variable Y is predicted by predictor X. The residual error E represents the unexplained variance in Y. All variables are assumed to be normally distributed. Y is distributed with the mean μ_X and variance σ^2_X, the residual has zero mean and a variance σ^2_E. The term β_0 is the regression intercept and β_1 is the regression weight of X on Y.

3.2 Classes of Structural Equation Models

In the following, common types of SEMs are briefly discussed. Knowing these basic model types is a pre-requisite for understanding the applications introduced in Chapter 5.

3.2.1 Regression Model

Regression models are the basic models to express and test linear relationships. Generally, regression models estimate the conditional expectation of one or more dependent variables given a set of independent variables.

Univariate and multivariate regression models can be represented as SEM in a straightforward way. The dependent and independent variables of the regression will be observed variables. The error term for each dependent variable will be represented by a latent variable. A matrix-style notation of a multivariate regression model follows. Let y_i be n dependent variables, x_i be m dependent variables and β_{ij} be the regression weight of x_i on y_j. The n residual error terms are denoted by ϵ_i. The relation of the variables in a regression model is

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{21} & \dots & \beta_{m1} \\ \beta_{12} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \beta_{1n} & \dots & \dots & \beta_{mn} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Figure 3.2.1 shows a graphical representation of an univariate regression model.

3.2.2 Latent Factor Model

The *latent factor model* is a formulation of factor analysis in the SEM context. Factor analysis is a widely used technique in psychology. A famous example is the hypothesis of intelligence

being a underlying hypothetical factor which can not be measured directly but as the common variance shared between several tests (Spearman, 1904). Essentially, in factor analysis, a set of one or more latent factors is sought that can be measured by observed variables. Typically, the latent means and variances of the latent factors, their correlations, and the factor loadings that determine to what extent a latent factor is measured by an observed variable, and, finally, the residual or unique variance of each observed variable are at the center of attention. The model is illustrated by the example of a two-factor model. Let y_i be random variables representing n observations, and ϵ_t the random variables representing the residual variances. Let ξ_1 and ξ_2 be two latent factors and $\lambda_{i,j}$ the entries of the structural matrix representing the regression weights between factor i and observation j. These are also referred to as *factor loadings*. A two-factor model featuring observations that are manifestations of a unique factor can algebraically be formulated as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \lambda_{1,1} & 0 \\ \vdots & \vdots \\ \lambda_{1,k} & 0 \\ 0 & \lambda_{2,(k+1)} \\ \vdots & \vdots \\ 0 & \lambda_{2,n} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \epsilon_n \end{bmatrix}$$

An illustration of this model is found in Figure 3.2.2. The above model allows the modeling of covariances between latent factors by allowing $cov(\xi_1, \xi_2)$ to be a freely estimated parameter. Analogously, for models with more than two factors, the latent covariance matrix can be freely estimated.

When introducing latent factors to a model of observed variables, the scaling of this newly added hypothetical construct is left undefined. If all factor loadings from the latent factor to the observed scores are freely estimated, the scaling of the latent variable is arbitrary. This is also referred to as the "normalization problem" (McArdle & Prescott, 1992). In practice, there are two solutions to this problem. Either the variance is normed to unity, which standardizes the factor loadings relative to the latent score, or a single loading is fixed to unity so that the remaining loadings are factor loadings relative to the chosen reference variable. A scaling of this kind is generally a prerequisite for each latent variable. Scaling the latent variables is especially important if latent factor models are to be compared across groups (Jöreskog, 1971).

3.2.3 Latent Growth Model

In social and behavioral sciences, the analysis of longitudinal data has become an increasingly important topic. Instead of testing only pre- and post-effects of a treatment, the analysis of trajectories of variables and the analysis of change over time certainly offers more information about the investigated process. The following section follows descriptions by Bollen, Curran, and Wiley (2006).

Latent Growth Curve Models (LGCM) are a re-representation of random effects trajectory models (Meredith & Tisak, 1990) which was extended by several authors (McArdle, 1988; B. Muthen & Curran, 1997). The random coefficient model models trajectories over time by including an intercept and a linear slope variable, both of which were assumed to be normally

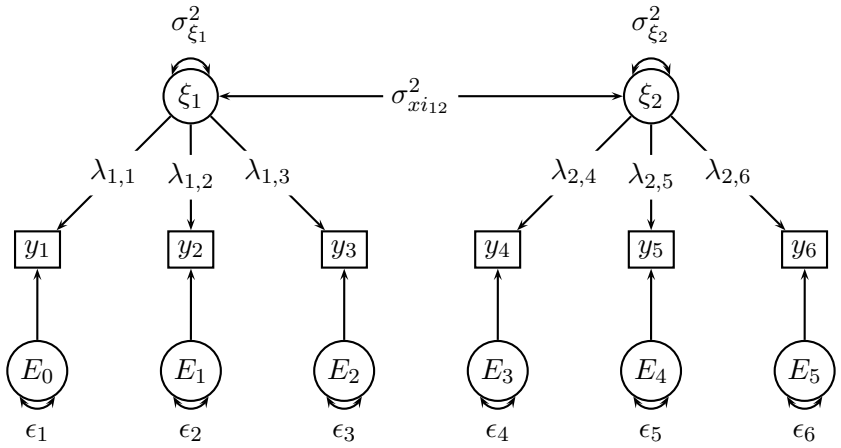


Figure 3.2.2: Latent factor model with six observed variables y_1-y_6 that are manifestations of two latent variables ξ_1 and ξ_2. The residual variances for the six observations are modeled as ϵ_1-ϵ_6. Latent variances are modeled as $\sigma^2_{\xi_1}$ and $\sigma^2_{\xi_2}$. The covariance between both latent factors is $\sigma^2_{\xi_{12}}$. Expectations are not modeled in this example. The factor loadings from latent variable i to observed variable j are modeled as six parameters $\lambda_{i,j}$.

distributed across the population. The intercept variable was assumed to have a constant influence on each observation but could vary across people. Similarly, the slope described a linear trend over time. This can be formulated in the SEM context. Let y_t be a random variable representing the observation at time point t of a total of T time points, ε_t being a random variable with the residual variance at time point t, I a random variable representing the intercept of the growth curves, and S a random variable representing the slope, that is, the linear increment between two observations. The essential growth equation can be written as

$$y_t = I + S \cdot t + \varepsilon_t \tag{3.2.1}$$

Equivalently, in matrix notation

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & T-1 \end{bmatrix} \begin{bmatrix} I \\ S \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{bmatrix}$$

By modifying the factor loadings of the slope term, non-linear growth functions can be modeled while the hypothesized model will remain linear in the parameters. Other types of growth functions can be employed easily. For example, a quadratic trajectory can be modeled by taking the square of the time, resulting in the following growth equation

$$y_t = I + S \cdot t^2 + \varepsilon_t$$

Analogously, exponential growth can be incorporated in a model with an exponential transformation of the factor loadings of the slope component

$$y_t = I + S \cdot \exp(\delta \lambda_t) + \varepsilon_t$$

where δ represents a rate of change. Following the same logic, cyclic functions can be incorporated in a LGCM by using trigonometric functions. For a frequency f and phase offset p, we obtain

$$y_t = I + S \cdot \cos[2\pi f(t+p)]$$

Finally, factor loadings can also be freely estimated, resulting in models that can represent arbitrary non-linear functions. However, the interpretation of the slope component becomes difficult since the term "slope" is no longer connected to a functional form.

Often, growth is hypothesized to be a mixture of a set of elementary growth functions. For example, scores of participants in a cognitive training study can be expected to increase linearly at the beginning of a study. Growth then saturates towards the maximum level the participant will be able to achieve. This increase could be modeled as a combination of linear and inverse exponential growth. For another example, an investigated variable could exhibit seasonal changes and a long-term quadratic growth. This could be modeled effectively by a cyclic and a quadratic component. The LGCM model can easily be extended to incorporate an arbitrary number of latent variables representing growth functions through the respective factor loadings as described above. Figure 3.2.3 shows an example of a LGCM with a linear and an exponential decay. The corresponding matrix notation is:

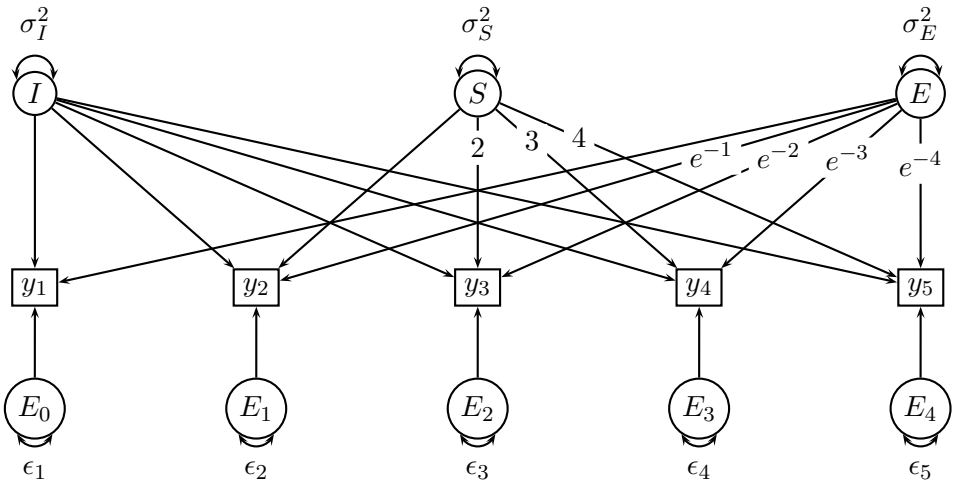


Figure 3.2.3: Latent growth curve model with three latent variables representing intercept (I), linear slope (S), and exponential slope (E). In this example, five equidistant observations are modeled with independent and individual residual error variances.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & e^{-1} \\ \vdots & \vdots & \vdots \\ 1 & T-1 & e^{-T+1} \end{bmatrix} \begin{bmatrix} I \\ S \\ E \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{bmatrix}$$

3.2.4 Autoregressive Model

The *auto-regressive model* (AR) is a model for time series of the class of the so-called *Box-Jenkins models* (Box, Jenkins, & Reinsel, 1976). The underlying principle is simple. Observations are linear combinations of a fixed number of previous observations and an additive noise component. Formally, we write

$$x_t = \sum_{i=1}^{m} \beta_i \cdot x_{t-1} + \varepsilon_t$$

with x_t being the observations of a time series x at time point t, β_i being the weights of the past observations, and $\epsilon \sim N\left(0, \sigma_\epsilon^2\right)$ a Gaussian noise component. For a choice of all $\beta_i = 0$, all observations are independent and normally distributed and the process will collapse to a white-noise process.

When dealing with multivariate time series, autoregressive models are often encountered as cross-lagged models. They extend separate AR models by introducing additional linear cross-influences between the time series. This model class is also known as Vector Autoregression (cf. Holden, 1995). The bivariate case of such a model could be formulated as

$$
\begin{aligned}
x_t &= \sum_{i=1}^{m} \beta_{xi} \cdot x_{t-i} + \gamma_{yi} \cdot y_{t-i} + \varepsilon_t \\
y_t &= \sum_{i=1}^{m} \beta_{yi} \cdot y_{t-i} + \gamma_{xi} \cdot x_{t-i} + \varepsilon_t
\end{aligned}
$$

A graphical representation of a bi-variate AR model is depicted in Figure 3.2.4.

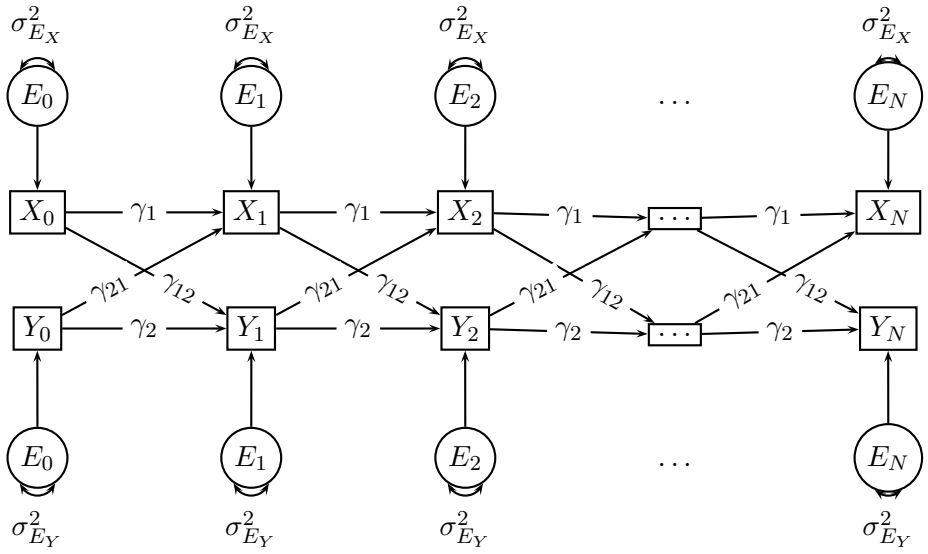


Figure 3.2.4: Modeling two observed time series and a lag-1 relation within and between the time series using a bi-variate auto-regressive model. The time series X and Y are observed on N time points. The autoregressive weights are γ_1 and γ_2, and the cross-lagged weights are γ_{12} and γ_{21}.

3.3 Extending SEMs to SEM Trees

When searching for a model, researchers ideally build an initial model based on their prior hypotheses and evaluate the goodness-of-fit of the data for the model. In applied research, initial models often do not fit adequately and are therefore subsequently adapted and modified. In the course of statistical analysis of a data set, multiple models may serve as possible candidates from which the best model is chosen, according to the researcher's criteria and hypotheses. It is not a rare occurrence that only the final model is reported, along with inferential statistics that are based on the assumptions of a single-shot analysis. One way to increase the predictive quality of a model is to account for heterogeneity in a data set by splitting it into multiple groups. If represented by the same structural model, these groups can be compared by differences in the estimated parameters. In a new approach, I suggest exploit this process systematically by (1)

exhaustively searching all available covariates and choosing the one that maximizes the information about the observed variables in the model, and (2) continuing this process recursively as long as covariates that explain differences in the observed variables can be determined. This leads to a systematic approach that detects covariates and covariate interactions with respect to the parameters of a chosen SEM. The result is a tree structure of covariates; and the method can be seen as a decision tree approach to SEMs. In the following, I will briefly review the decision tree paradigm and, then, formally introduce SEM Trees.

3.3.1 The Decision Tree Paradigm

Trees are data structures that represent hierarchical organizations. Formally, trees are directed graphs, consisting of nodes and connecting directed edges, without loops. A non-empty tree has a single node without incoming edges. In a tree, each node has at most one incoming edge and the edges do not form circles. All nodes without outgoing edges are called *leaves*. All nodes that are not leaves are called *inner nodes*. Trees can effectively represent partitions of a data set if each node represents a split rule and the branches are associated with the resulting split data.

Decision trees constitute an early supervised learning scheme. Assume, for each subject, an observed categorial outcome y and a vector of covariates x were collected. A decision tree describes a partition of the covariate space that describes significant differences in the outcome variable. The partitions of the covariate space are described by inequalities on the individual dimensions of the covariate space. This enables decision trees to be read like rule sets, for example, a rule could be read as "if $x_1 > 5$ and $x_2 < 2$ then $y = 0$ else $y = 1$". Formally, decision trees describe partitions of the covariate space that are orthogonal to the axes of the covariate space (see Figure 3.3.1). The problem of learning a decision tree is solved by finding a tree that represents a partition of the feature space predicting the target variable as well as possible. There are various formalizations of what constitutes a good split, e.g., a maximal reduction of entropy of the outcome variable (Quinlan, 1986). Presumably, the earliest representative of this idea is the *Chi-square Automatic Interaction Detector* (CHAID; Sonquist & Morgan, 1964), which uses the χ^2-test of independence to find hierarchies of variable interactions. The paradigm gained popularity with Breiman's *Classification and Regression Trees* (CART; Breiman, Friedman, Olshen, & Stone, 1984) and the *ID3* algorithm (Quinlan, 1986). The algorithms have in common that they employ greedy approaches to improve a target function. For example, in ID3, the information gain of a target variable is maximized, that is, the entropy of the target's class distributions in the resulting hypercubes is minimized.

The decision tree paradigm has been improved constantly and taken to more sophisticated levels. The original idea was designed only for categorial outcomes. Later extensions incorporated models that could describe more complex outcomes. These so-called model-based trees have appeared in many fields, among them M5 (Quinlan, 1992) that allows multivariate regression models in the leave nodes, Naive Bayes Trees, Logistic regression trees (Landwehr, Hall, & Frank, 2005) or MARS (Friedman, 1991), which builds decision trees with spline models. A recent comprehensive framework for model-based recursive partitioning was presented by Zeileis, Hothorn, and Hornik (2008) with a corresponding freely available package *party* for the statistical programming language R (R Development Core Team, 2011). Nodes of the decision trees are represented by parametric models whose parameters are estimated from the data set.

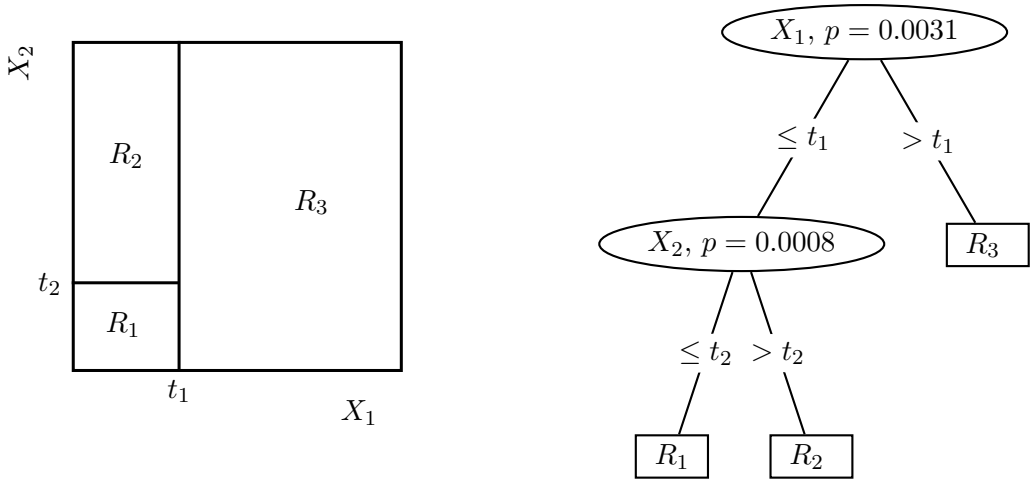


Figure 3.3.1: Decision trees describe partitions of the covariate space that explain differences in the outcome. Right: A decision tree describing partitions of the covariates X_1 and X_2 into the areas R_1, R_2, and R_3. In this example, covariate X_1 was split at value t_1 with a significance $p = 0.0008$. Additionally, a subsequent split of covariate X_2 at value t_2 was found with a significance of $p = 0.0034$. Left: A two-dimensional plot of the partitioned covariate space in the areas R_1, R_2, and R_3.

The originally specified model is recursively split into submodels whenever these fit the data significantly better. Zeileis, Hothorn, and Hornik (2006) present applications of such recursive partitioning with linear regression models, logistic regression models, and Weibull regression for censored survival data.

3.3.2 Structural Equation Model Trees

Model-based trees enhance the decision tree paradigm by partitioning the data set with respect to differences of a hypothesized model. Model-based trees, despite their variety, are models of observed variables. In this Chapter, I propose SEM Trees, which extend this line of research to models with an observed and a latent structure. SEM Trees can be conceived of as a hierarchical structure of models in which each model is a SEM with unique parameter estimates. Due to the high flexibility and generality of SEM, SEM Trees offer tree-based analyses of a large range of models under a single theoretical framework, including regression models, factor analytic models (Jöreskog, 1969), autoregressive models (Jöreskog, 1970), latent growth curve models (McArdle & Epstein, 1987), latent difference score models (McArdle & Hamagami, 2001), or latent differential equations (Boker, Neale, & Rausch, 2004). Questions that can be answered with SEM Trees using these models include: "Is there a significant difference in the expected values of a latent construct in subpopulations of the data set with respect to the covariates?" or "Does the relation between two hypothetical constructs differ with respect to interactions of covariates?". Importantly, tree structures enable the interpretability of the results by virtue of a set of decision rules (Quinlan, 1993) and thereby guide the researcher in acquiring accessible knowledge about the data set at hand.

A problem that is often encountered in daily research life is the heterogeneity of data sets. SEM relies on the implicit assumption that the sample is homogeneous. However, this notion might often be misleading. If all observations come from the same model with the same parameter set, the data set is considered homogeneous. And if the data set originated from two or more different models or same models with different parameter sets, it is considered heterogeneous. In the context of SEM Trees, homogeneity is defined in the context of a generative hierarchical model that is thought to have produced the observed samples. Oftentimes, researchers split their data sets according to a variable that explains a large amount of heterogeneity in a data set. For example, lifespan researchers might split their data set into two parts, young and old, on an a-priori basis and fit separate models to the data set. However, it is unclear whether a different split according to another known covariate or a split at a different age could explain more heterogeneity. On the other hand, a researcher might be misled to split the data set at all, despite there being no apparent heterogeneity. Also, if a single partition of the data set is approved, why should researchers stop partitioning? There might indeed be further partitions of each subpopulation that could explain important differences.

Indeed, researchers often perform a more or less extensive model selection procedure. Often, initially hypothesized models are successively extended by covariates or are split into different groups with respect to a single covariate. The lack of a formal setting for this process poses a significant danger for scientific theory building. Test statistics and p-values may be reported without corrections although in fact, multiple models were previously compared. In this situation, the likelihood that the findings were indeed random increase successively. Most importantly, this process is susceptible to implicit and explicit selection bias (Kriegeskorte et al., 2009).

To recapitulate the situation: Initial hypotheses about a data set exist and can be formalized in a SEM. In addition, there is a set of covariates that describe properties of the data set and are not yet part of the model. The emerging question is how these covariates can be used to refine the model. From the point of view of building psychological theory, this question ultimately becomes: "How can we algorithmically guide informed theory construction?"

SEM Trees offer a formal setting for the described model selection process. Within a single framework, they combine exploratory and confirmatory approaches. Heterogeneity in a data set is detected by automatically finding covariates inducing partitions of the data set that explain largest differences in the model parameters. By applying this procedure recursively, covariates and covariate interactions that ultimately partition the data set into homogeneous subgroups with unique parameter estimates are detected. Theory-based hypotheses can be included in the model while the model is successively refined in an exploratory fashion, and the resulting refined hypothesis is confirmed in a final step.

3.3.3 Graphical Representation

SEM Trees are depicted as tree graphs with nodes and edges. Inner nodes are drawn as ovals and represent splits of the data set with respect to a specific covariate. Each node is labeled with the name of the covariate that is split at that level. If the split was chosen according to a statistical criterion, a p-value representing the probability of observing a test statistic of the observed value or larger under the hypothesis that the split is uninformative is available. If available, the p-value associated with this split is appended to the node label to reflect the

significance of the node. Leaves of the tree present the template model's estimated set of free parameters for the subset of the data according to the performed data set partition. Leave nodes are represented as rectangles containing information about the estimated parameters of the sub-model. This includes the number of data points N of the original sample that are contained in the respective partition, all parameter estimates, and if required, estimates of their standard error indicating the significance of differing from zero. Often, parameter estimates are marked with symbols representing the level of significance of a test against the null hypothesis that the true parameter value is zero. This serves as a first indicator of whether the parameter is relevant to the model or not. Usually, significance on the .01-level is marked by a single asterisk, and significance on the .001-level is marked by two asterisks. However, this practice is debated and others prefer to report the standard errors or confidence intervals. I recommend reporting standard errors, but, ultimately, this is a choice to be made by the researcher using SEM Trees. This graphical representation is schematically depicted in Figure 3.3.2.

To understand the formation of partitions, it can be insightful to examine parameter estimates of inner nodes. Therefore, inner nodes can be annotated with their internal node identification number which is an integer number starting at zero for the root node and a depth-first ordering of all other nodes. If desired, this number can be rendered as a small boxed annotation above the respective inner node.

SEM Trees can grow large with numerous covariates and data sets with many observations. To retain clarity of graphical depictions of large SEM Trees, an alternative graphical display is proposed: All leaf nodes are collapsed to only their node identification number and parameter estimates are listed in a tabular display with references to the node identification numbers below the graph.

3.3.4 An Exemplary SEM Tree

For illustration of the interpretation of a SEM Tree, I would first like to present a hypothetical example. Assume a data set consisting of scores on a cognitive test observed for 400 participants was collected. Half of each group participated in a cognitive training program expected to attenuate cognitive decline with increasing age. Therefore, participants were recruited from two age groups, young and older adults. Beyond the cognitive test results for each participant, it was noted whether they were in the young or the old group, whether they participated in the training program or were assigned to a control group, and whether they were male or female.

In our example, the decline process is assumed to be linear and thus modeled with a linear LGCM, as earlier described in Section 3.2.3. The employed model measures a hypothetical cognitive score on four occasions of measurement (cf. McArdle, Ferrer-Caja, Hamagami, & Woodcock, 2002, for a comparative longitudinal structural analysis of intelligence). Model parameters include the mean μ_I and variance σ_I^2 of the latent intercept and the mean μ_S and variance σ_S^2 of the latent slope and without a correlation between both latent constructs. The corresponding SEM is shown in Figure 3.3.2.

SEM Tree analysis can now be employed to detect differences in the growth curves of subgroups in the sample described by the covariates. Assume the following situation: Cognitive decline affects younger adults only mildly, whereas older adults are affected more strongly. The treatment only affects older participants by reducing the slope of their cognitive decline. The treatment affects participants of both sexes equally, and there are no baseline differences between

males and females in the two age groups. In this situation, a SEM Tree finds the respective covariate interaction of the covariates "age" and "treatment." In this case, three significantly different groups are expected, consisting of the young participants with their prototypical decline curve, while the old training group and the old control group differ both between each other and with regard to the first group. A data set was simulated in this way. The resulting SEM Tree is illustrated in Figure 3.3.2. The SEM Tree splits the data set according to age group. Then it recursively splits the older one into two subgroups that significantly differ in their growth curve parameters. Since the group of younger is homogenous, it is not selected for a further split. The tree does not include the variable "sex" as it has no explanatory power with respect to differences in the parameter estimates for any of the found subgroups.

3.3.5 Formal Definition

In the following, I formally introduce SEM Trees. This includes a definition of the tree and a traversal function that allows retrieval of a tree's leaf node that corresponds to a particular observation. This enables the calculation of a negative two log-likelihood of observing a data set given a tree. For the generation of SEM Trees, I derive the likelihood ratio test as a tool for the evaluation of when to split a tree and when to stop the process.

First, we need a formal definition of a data set that consists of rows of observations and columns representing observed variables. Furthermore, we make a distinction of two types of observed variables: (1) Variables that are included in the model and (2) candidate variables that are potential candidates for the inclusion in the model.

Definition 45. A data set D is a tuple $D = (O, C)$ such that the observation matrix O is a $(n \times o)$-matrix and the covariate matrix C is a $(n \times c)$-matrix, with elements $o_{ij} \in \mathbb{R}$ and $c_{ij} \in \mathbb{R}$.

The matrix O ("observed") contains the variables that are modeled in the SEM, and the matrix C ("covariates") refers to variables that are not part of the model. A SEM Tree describes a recursive partition of the multivariate covariate space, such that the corresponding observations in each partition are described by a SEM with a unique parameter vector. In order to describe the recursive partition of the covariate space, a binary tree structure that uses labeling functions to encode the partitions and the resulting parameter estimates is employed:

Definition 46 (SEM Tree)**.** A *SEM Tree* is a tuple $\Upsilon = (M, r, N, E, \mathcal{L}_N, \mathcal{L}_E, L_\theta)$ such that

- M is a SEM with k free parameters, referred to as the *template model.*
- N is a set of nodes and $E \subseteq N \times N$ a set of directed edges, such that (r, N, E) is a binary tree, i.e., each node has two or zero outgoing edges and exactly one incoming edge with exception of the *root* $r \in N$, which has no incoming edges. The path from the root to each node is unique.
- $\mathcal{L}_N : N \to \mathbb{N}$, $\mathcal{L}_E : E \to 2^{\mathbb{R}}$ and $\mathcal{L}_\theta : N \to \mathbb{R}^k$ are labeling functions for nodes and edges.
- If $(a, b), (a, c) \in E, b \neq c$, then $\mathcal{L}_E((a, b)) \cap \mathcal{L}_E((a, c)) = \emptyset$; in other words, for each node the labels of the outgoing edges are disjoint.

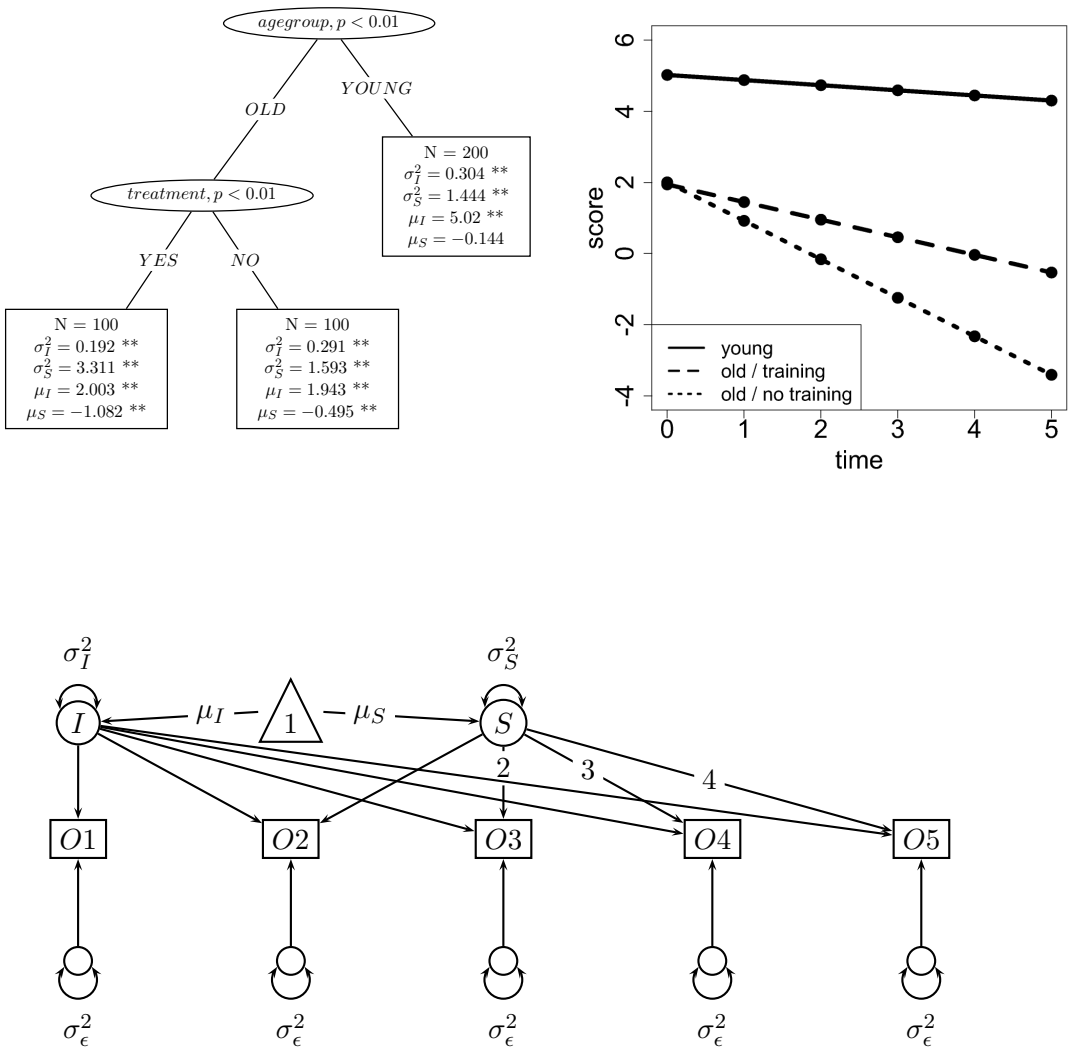


Figure 3.3.2: Top left: Illustration of a SEM Tree built with the template model below on a simulated data set. Top right: Expected trajectories with respect to the group structure retrieved by the SEM Tree. Bottom: Linear growth curve model that serves as a template SEM for the tree above with intercept $I \sim \mathcal{N}(\mu_I, \sigma_I^2)$, slope $S \sim \mathcal{N}(\mu_S, \sigma_S^2)$, and residual error terms with a variance of σ_E^2.

Nodes with no outgoing edges are called *leaves,* or otherwise *inner nodes.* We denote the set of leaves of the tree with $\mathcal{L}(\Upsilon)$, and the set of all inner nodes, i.e., nodes that have two successors, with $I(\Upsilon)$. The number of leaves and the number of inner nodes of a tree will be noted by the cardinalities of the respective sets, $|L(\Upsilon)|$ and $|I(\Upsilon)|$. A sub-tree Υ' is a subset of nodes of Υ which again is a tree, and is denoted as $\Upsilon' \preceq \Upsilon$. The depth of a node is the length of its path from the root. The depth, or sometimes also the height, of a tree is the depth of its deepest node. Therefore, a tree with only the root node has a depth and height of zero.

Essentially, SEM Trees are a template SEM and a labeled binary tree. The tree encodes decisions with respect to covariate values, that is, for a given observation with corresponding covariates, the tree can be traversed in order to find a leaf node which contains a parameter vector for the template model M that describes a likelihood for the corresponding observation. The labeling functions play the following role: $\mathcal{L}_N$ labels nodes with the index of a covariate, that is, the covariate that decides whether we continue to traverse the left or the right sub-tree. The edge labeling function $\mathcal{L}_E$ provides information which sets of covariate values are associated with the left or the right sub-tree of an inner node. Finally, a second node labeling function $\mathcal{L}_\theta$ returns a parameter vector for the template model for each node. In this way, SEM Trees assign different parameter vectors to hypercubes in the covariate space via the following traversal function:

Definition 47. The traversal function $\phi_\Upsilon : \mathbb{R}^c \to \mathbb{R}^k$ of a SEM Tree Υ with root r is defined as $\mathcal{L}_\theta(r)$ if Υ has only one node. Otherwise, let r_{left} and r_{right} be the children of r and Υ_{left} and Υ_{right} the corresponding sub-trees. The traversal function is

$$\phi_\Upsilon(x) = \begin{cases} \phi_{\Upsilon_{left}}(x) & x_{\mathcal{L}_N(r)} \in \mathcal{L}_E((r, r_{left})) \\ \phi_{\Upsilon_{right}}(x) & x_{\mathcal{L}_N(r)} \in \mathcal{L}_E((r, r_{right})) \end{cases}$$

If Υ is obvious, we write ϕ instead of ϕ_Υ.

The traversal function can be used to determine the likelihood of an observation $x \in O$ by finding the node corresponding to the decisions that are encoded by the nodes and edge labels, on the corresponding covariate values $c \in C$ as follows:

Corollary 48. *Let Υ be a SEM Tree with template model M, Σ be the model-implied covariance matrix of M, and μ the model-implied expectations vector of M. The negative two log-likelihood of an observation $x \in O$ with corresponding covariates $c \in C$ is*

$$-2LL(x|\Upsilon) = -2LL(x|\Sigma(\phi(c)), \mu(\phi(c)))$$

Let Υ, Σ, μ be as above. The negative two log-likelihood of a data set D given a SEM Tree is

$$-2LL(D|\Upsilon) = \sum_{(x,c)\in D} -2LL(x|\Sigma(\phi(c)), \mu(\phi(c)))$$

With each split in a node n of a SEM Tree, the current data set is partitioned according to the value of the covariate that is determined by the node-labeling function $\mathcal{L}_N(n)$. In accordance with the definition of the traversal function, we can refer to the resulting split data set as a resulting *left data set* and *right data set:*

Definition 49. Let $D = (O, C)$ be a data set and r a node of a SEM Tree, and let r_{left} and r_{right} be the children of r. The observations are partitioned according to the decision rule encoded by node r into

$$O_{left} = \left\{ O_i | C_{i,\mathcal{L}_N(r)} = \mathcal{L}_E\left((r, r_{left})\right) \right\} \text{ and } O_{right} = \left\{ O_i | C_{i,\mathcal{L}_N(r)} = \mathcal{L}_E\left((r, r_{right})\right) \right\}$$

and, analogously, the covariates are partitioned into

$$C_{left} = \left\{ C_i | C_{i,\mathcal{L}_N(r)} = \mathcal{L}_E\left((r, r_{left})\right) \right\} \text{ and } C_{right} = \left\{ C_i | C_{i,\mathcal{L}_N(r)} = \mathcal{L}_E\left((r, r_{right})\right) \right\}$$

The data set D is partitioned by the decision rule encoded by node r into a *left data set* $D_{left} = (O_{left}, C_{left})$ and a *right data set* $D_{right} = (O_{right}, C_{right})$.

3.4 Algorithmic Aspects of SEM Trees

Above, I have formally defined a SEM Tree and tools that allow us to traverse the tree and obtain the likelihood of observations given the tree. In this section, algorithmic aspects of building a SEM Tree are discussed.

3.4.1 Construction of SEM Trees

In the following, I present a greedy algorithm that, given a data set D and a template model M, builds a tree that minimizes the negative two log-likelihood of observing the data given the tree locally, that is, at each node. After laying the formal basis for evaluating split variables, a pseudocode algorithm that allows the construction of a SEM Tree is given.

Building a SEM Tree starts with the choice of a template SEM M and a data set D that is to be analyzed. First, a root node is created and ML parameters are estimated from the complete observed data. Growing the tree proceeds by iteratively testing whether the current set of leaf nodes, initially only the root, can be split further. Splits should only be performed if the split is useful. The utility of a split is typically formalized with statistical or information-theoretic measures indicating that the split candidate contains relevant information about the target variables. If several potential splits are useful, the best candidate is chosen. The evaluation of split candidates is based on the fundamental notion of a multi-group model (Jöreskog, 1971), that is, a compound SEM of multiple independent SEMs modeling samples from multiple distinct groups. SEM Trees use the property that a SEM which is partitioned into two parts according to a dichotomous covariate, one associated with each resulting partition of the data, can be seen as a two-group model. Furthermore, the model that has not yet been split can be seen as a restricted version of this two-group model. I will show that this is an algebraic nesting, that is, the parameter vector of the original model is a constrained form of the parameter vector of the two-group model. Because of this property, the distribution of the ratio of the log-likelihoods between the two models is known under the hypothesis that the split candidate is uninformative. This allows us to formulate a statistical test that rejects the null hypothesis of an uninformative split candidate.

In the following, I will derive the *maximum-χ^2 criterion* to determine whether a node should be further expanded to a sub-tree or not. The criterion is based on the notion that a split should be performed with respect to one of the covariates whenever its result significantly increases the likelihood of observing the data given the tree. Based on the likelihood ratio test, the maximum-χ^2 criterion allows us to ground the decision to accept or reject a split candidate on a significance test. In order to derive this criterion, we need two notions of models, a *pre-split model* representing the model before a candidate split, and *a post-split model* representing the model after the split.

Definition 50. Let Υ be a SEM Tree with template model M that is described by a expectations vector μ and a covariance matrix Σ. Let $n \in N$ be a node with data set D. The pre-split model of node n is a SEM M with ML parameter estimates $\hat{\theta}_{pre}$ based on minimizing $-2LL\left(D|M\left(\hat{\theta}_{pre}\right)\right)$. D is referred to as the data set of the *pre-split model* of n.

There is an important dual representation of the pre-split model. In the first instance, the pre-split model is a representation of the current data set, as described in the previous definition. Its parameter estimates are obtained by minimizing the negative log-likelihood function for observing the data set given the model. Considering a split of the data set into a left and a right data set, the likelihood of this model is equivalent to the likelihood of a compound model that consists of a left model that is associated with the left data set, and a right model associated with the right data set. When the parameter estimates for the left and right models are obtained by minimizing the respective log-likelihood function under the additional constraint that the parameter vectors of both models are equal, the resulting likelihood of the compound model is equal to the likelihood of the pre-split model as defined above. Even if this observation is not surprising, it is an important building block in deriving a hypothesis test for evaluating the significance of choosing a covariate for a node split:

Theorem 51 (Dual Representation of Pre-Split Models)**.** *Let Υ, M, D, n be as above. Let D_{left} and D_{right} be the left and right data sets of D with respect to the covariate encoded by n. Let $\hat{\theta}_{pre,left}$ minimize $-2LL\left(D_{left}|M\left(\hat{\theta}_{pre,left}\right)\right)$* and $\hat{\theta}_{pre,right}$ minimize $-2LL\left(D_{right}|M\left(\hat{\theta}_{pre,right}\right)\right)$ *under the equality constraints $\hat{\theta}_{pre,left} = \hat{\theta}_{pre,right}$, then*

$$-2LL\left(D|M\left(\hat{\theta}_{pre}\right)\right) = -2LL\left(D_{left}|M\left(\hat{\theta}_{pre,left}\right)\right) - 2LL\left(D_{right}|M\left(\hat{\theta}_{pre,right}\right)\right)$$

Proof. Let $(O, C) \in D$ and N be the number of observed data points O in D. Let D_{left} be the left data set of D and D_{right} be the right data. Let N_{left} and N_{right} be the number of observations in the respective partitions, such that $N_{left} + N_{right} = N$. With Equation 3.1.3, we can rewrite this as

$$
\begin{aligned}
-2LL\left(D'|\mu\left(\hat{\theta}_{pre}\right),\Sigma\left(\hat{\theta}_{pre}\right)\right) &= N\cdot p\log\left(2\pi\right)+N\log|\Sigma| \\
&\quad +\sum_{x\in O}\left[\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right)\right] \\
&\quad (N_{left}+N_{right})\cdot p\log\left(2\pi\right)+(N_{left}+N_{right})\log|\Sigma| \\
&\quad \sum_{x\in\left(O_{left}\cup O_{right}\right)}\left[\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right)\right] \\
&= N_{left}\cdot p\log\left(2\pi\right)+N_{left}\log|\Sigma| \\
&\quad +\sum_{x\in O_{left}}\left[\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right)\right] \\
&\quad +N_{right}\cdot p\log\left(2\pi\right)+N_{right}\log|\Sigma| \\
&\quad +\sum_{x\in O_{right}}\left[\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right)\right] \\
&= -2LL\left(D_{left}|M,\hat{\theta}_{pre,left}\right)-2LL\left(D_{right}|M,\hat{\theta}_{pre,right}\right)
\end{aligned}
$$

□

Theorem 52. *Let Υ, M, D, n be as above. Let D_{left} be the left data set and D_{right} the right data set of D with respect to the decision rule encoded in node n. Let $\hat{\theta}_{left}$ minimize $-2LL\left(D_{left}|M\left(\hat{\theta}_{left}\right)\right)$ and $\hat{\theta}_{right}$ minimize $-2LL\left(D_{right}|M\left(\hat{\theta}_{right}\right)\right)$. The likelihood of the post-split model that independently estimates parameters for the left data set and right data set is*

$$
-2LL\left(D_{left}|M\left(\hat{\theta}_{left}\right)\right)-2LL\left(D_{right}|M\left(\hat{\theta}_{right}\right)\right) \tag{3.4.1}
$$

Proof. This proof is analogous to the proof of Theorem 51. Since both models are assumed to be independent, their likelihoods multiply and consequently their log-likelihoods sum up. The only difference to the previous proof is that the equality constraint across the parameters in both models is removed, and therefore, two separate sets of parameter estimates $\hat{\theta}_{left}$ and $\hat{\theta}_{right}$ are obtained.

As reviewed earlier in Section 3.1.5, algebraic nestings of SEMs have the property to be approximately χ^2-distributed under the null hypothesis that the parameter restriction holds true in the population. This property allows the employment of significance testing to reject the null hypothesis and therefore rejection of the restricted model for being a worse representation than the unrestricted model. □

Corollary 53 (Algebraic Nesting of Pre-Split and Post-Split Models)**.** *The pre-split model is nested in the post-split model because the models are structurally equivalent and they differ by equality constraints equal to the number of free parameters in M.*

An illustration of Corollary 53 is given in Figure 3.4.1. The general principle of nested pre-split and post-split models is illustrated by means of a schematic factor model with three freely estimated factor loadings λ_1, λ_2, and λ_3, and the variance of the common factor σ^2_ξ. The pre-split model (upper panel) freely estimates the four parameters. Following Theorem 51, this

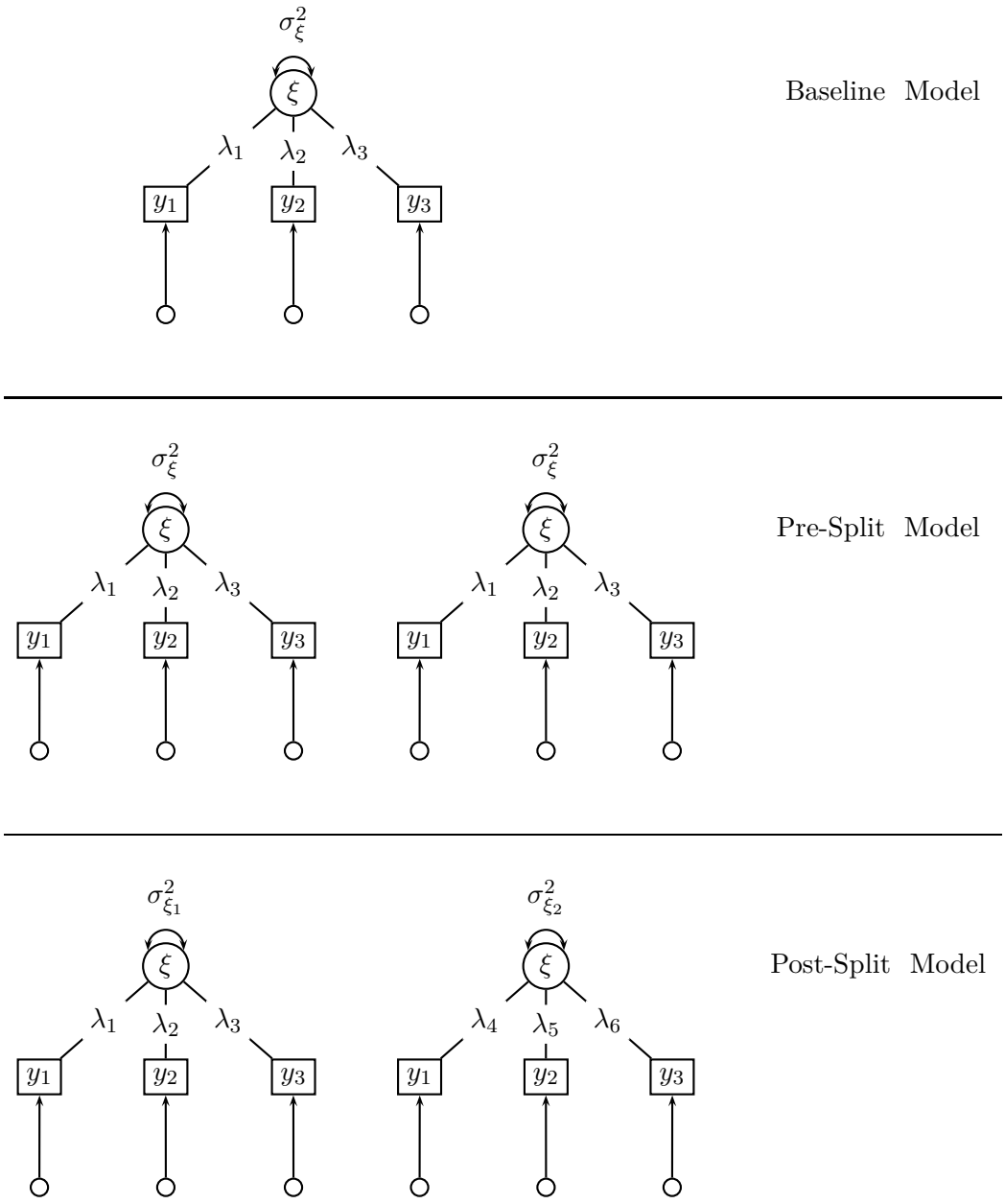


Figure 3.4.1: Evaluating a split candidate involves a comparison of a baseline model (top) that models the data set without partitioning it and a two-group model that represents two models, one model for each of the two partitions after the split (below). This figure illustrates that these models are indeed nested. The baseline model is equivalent to a two-group model in which all parameters are constrained to be equal across the group (middle). This virtual two-group model is in fact nested in the real two-group model (below). This allows the usage of a log-likelihood ratio test to reject the baseline model if the test statistic is significant.

model is equivalent to a two-group SEM that has equality constraints on all corresponding parameters, as depicted in the middle panel. When evaluating a split candidate, all parameters of the two models in the two-group model are freely estimated (post-split model, lower panel). The pre-split model is algebraically nested in the post-split model, and therefore, the asymptotic distribution of the likelihood ratio under the null hypothesis that the restrictions hold in the population, i.e., the split is uninformative, is known:

Theorem 54. *Under the null hypothesis that a split candidate is uninformative, the likelihood ratio LLR between a pre-split and a post-split model is asymptotically χ^2-distributed with degrees of freedom corresponding to the number of free parameters k in the template SEM.*

$$LLR = -2LL\left(D|M\left(\hat{\theta}\right)\right) + 2LL\left(D_{left}|M\left(\hat{\theta}_{left}\right)\right) + 2LL\left(D_{right}|M\left(\hat{\theta}_{right}\right)\right) \sim \chi^2_{df=k}$$

Proof. Following Wilks's (1938) theorem, as reiterated in Section 3.1.5, LLR is asymptotically χ^2-distributed under the null hypothesis that the restriction holds in the population, with degrees of freedom equal to the difference in freely estimated parameters. A pre-split model with k free parameters is compared to a post-split model with $2k$ parameters. The difference in parameters and therefore the degrees of freedom of the χ^2-distribution is k. □

Knowing the distribution of the test statistic LLR under the null hypothesis that the covariate is uninformative and therefore the true parameter values of the pre-split and post-split model are the same, we can formulate a statistical criterion that allows us to reject this null hypothesis for large values of LLR, and accept a split candidate as valid:

Definition 55 (Maximum-χ^2 criterion)**.** Let LLR_i be m log-likelihood ratios obtained from evaluating m candidate splits. Let z be the index that finds LLR_z with the maximum value. For a chosen significance level α, the *maximum-χ^2 criterion* for covariate selection selects the covariate with index z, if

$$LLR_z > crit, \quad P\left(\chi^2_{df} > crit\right) = \alpha$$

Given a set of covariates, all possible splits are evaluated and the model with the best increase in performance, i.e., the largest increase in likelihood of observing the data under the model, is compared against a previously chosen threshold, determined by the type-I error α of the test. If the split is statistically significant, splitting is continued recursively. Note that this model selection procedure is a typical example of the multiple comparison problem, and inferences on the acquired p-values must be taken with caution. Later, we will learn that LLR_z does not follow a central χ^2-distribution, and therefore α is not the type-I error rate for the question whether one of the splits is significant. The proof for that and remedies in form of different selection criteria are discussed in the next section. However, they do not change the general procedure described here but merely change the way the threshold is calculated or the way the statistic is estimated. Algorithm 3.1 gives a recursive pseudocode algorithm that creates a tree for a given template SEM and data set, based on the maximum-χ^2 criterion.

Since the Λ statistics for growing SEM Trees rely on asymptotic assumptions, it is convenient to define a minimum number of observations min_N that is required for a node candidate to be

Algorithm 3.1 Recursive function for creating a SEM Tree. Arguments of the function are a SEM *model* and a data set D

function SEM-TREE-LEARNING(*model*, D)
  $(obs, cov) \leftarrow D$ ▷ observations and covariates
  $best\_lr \leftarrow 0$
  $best\_covariate \leftarrow$ None
  $\theta \leftarrow$ ML estimate of *obs* under *model*
  for *covariate* in *cov* **do**
    obs_1, $obs_2 \leftarrow$ bi-partition of *obs* according to *covariate*
    cov_1, $cov_2 \leftarrow$ bi-partition of *cov* according to *covariate*
    $\theta_1 \leftarrow$ ML estimate of obs_1 under *model*
    $\theta_2 \leftarrow$ ML estimate of obs_2 under *model*
    lr $\leftarrow$ -2LL($obs|model(\theta)$ +2LL($obs_1|model(\theta_1)$)+2LL($obs_2|model(\theta_2)$)
    if $lr < best\_lr$ **then**
      $best\_lr \leftarrow lr$
      $best\_covariate \leftarrow covariate$
    end if
  end for
  critical $\leftarrow (1-\alpha)$-quantile of chi-square-distribution with degrees of freedoms equal to free parameters in *model*
  $cur\_node \leftarrow$ create new node with parameter estimate θ
  if $best\_lr > critical$ **then**
    remove $best\_covariate$ from each cov_1 and cov_2
    $left\_child \leftarrow$ SEM-TREE-LEARNING(*model*, (obs_1, cov_1))
    $right\_child \leftarrow$ SEM-TREE-LEARNING(*model*, (obs_2, cov_2))
    add $left\_child$ as child to cur_node
    add $right\_child$ as child to cur_node
  end if
  return cur_node
end function

considered legitimate. Nodes with a smaller number of observations will be skipped in node evaluation.

The SEM Tree algorithm passes parameter estimates of a node as starting values for submodels when evaluating splits. If the model does not converge, the original starting values can be used. If this still fails, either the potential split is disregarded or a new set of starting values has to be generated by a different estimation method, e.g., by a least-squares estimate.

In the parameter estimation process, the numerical optimizer might converge on negative values for estimated variances. This behavior is known as a Heywood case (Thurstone, 1947). There can be numerous causes for these cases, among these non-convergence, under-identification, model misspecification, outliers, and sample fluctuations. In practice, there are several conditions that require a stop to recursive branching. For once, the optimizer may not converge to a solution. In that case, the respective split has to be dismissed because the temporary solution cannot be trusted to be near a minimum of the error surface. Also, there is the possibility that the optimizer converges to a solution that falls into an inadmissible estimation area. In that case, too, the respective split is dismissed and a warning is passed to the user since the split could indeed have been meaningful if it could have been estimated correctly. A solution for these cases is a provision of new or additional sets of starting values that withdraw the optimizer from the critical region.

3.4.2 Handling Non-Dichotomous Covariates

SEM Trees, as described up to here, operate on dichotomous covariates. This ensures binary trees as an output representation. In psychological data sets, covariates are typically not always dichotomous but they are categorial, like occupation or type of education, ordinal, like the number of children or age in years, or continuous, like reaction time on a cognitive task. SEM Trees employ the approach usually taken in many conventional decision tree approaches (Quinlan, 1992, 1996; Zeileis et al., 2008) to handle these kinds of covariates. The original data set is mapped onto a proxy data set that replaces categorial, ordinal, and continuous covariates with sets of dichotomous covariates, the binary *split candidates*. The advantage of this procedure is that the same algorithm for determining splits can be used independently of the type of variable by only applying the respective mapping function. Furthermore, splits into multiple categories are still possible. For example, a tree that splits a covariate into three subgroups can be represented as a binary tree that first splits into two child nodes, representing category $\{1, 2\}$ and category $\{3\}$. After that, a second split on the next level is performed, splitting category $\{1, 2\}$ into $\{1\}$ and $\{2\}$.

The mapping function depends on the type of the original variable. For categorial data, the variable is mapped onto a set of variables representing all possible subsets of values. Ordinal variables are transformed into a set of dichotomous variables which represent a "smaller or equal" relation on all possible split points. In the worst case, a continuous variable in a data set of size N leads to representation of $N - 1$ dichotomous variables. Continuous data are similarly transformed like ordinal variables with a slight adjustment of the split point between each two observed values. In the following, the mapping procedure is reiterated for the different kinds of covariate types.

Formally, for an ordinal variable $C_{ordinal}$ with the n distinct and ordered observed values $v_1, v_2, \ldots, v_n$, the candidate set $\tilde{C}$ is constructed using a "smaller or equal" relation on all

values, resulting in $n-1$ split candidates:

$$\tilde{C}_i := \begin{cases} 0 & C_{ordinal} \leq v_i \\ 1 & otherwise \end{cases}, \quad i \in [1, 2, \ldots, n-1]$$

For a continuous variable $C_{continuous}$, a similar selection scheme is chosen. In order to reduce the bias of the threshold selection towards observed variables, the mean between each two successive observations is selected as threshold:

$$\tilde{C}_i := \begin{cases} 0 & C_{continuous} < \frac{1}{2}(v_i + v_{i+1}) \\ 1 & otherwise \end{cases}, \quad i \in [1, 2, \ldots, n-1]$$

For a categorial variable $C_{categorial}$ with values $c_1, \ldots, c_n$, we define the implied covariate $\tilde{C}_{A \cup B}$ for a partition $A \cup B = \{c_1, \ldots, c_n\}$ as

$$\tilde{C}_{A \cup B} := \begin{cases} 0 & C \in A \\ 1 & C_{categorial} \in B \end{cases}$$

For each categorial variable, the number of implied covariates is $2^{n-1} - 1$.

In the following, we will sometimes refer to sub-models as *left sub-model* and *right sub-model*, analogous to the notion of left and right data set. By definition, for a continuous or ordinal covariate, the left model represents the subset which has lower or equal values on a specified threshold and the right model corresponds to the subset which has larger values. For categorial variables, the right sub-model represents the subset of the sample with covariate values matching the specified set of values. The left sub-model represents the subset whose covariate values do not appear in the selected set. Graphical representations of SEM Trees adhere to this principle.

3.4.3 Time Complexity

In this section, I will analyze the time complexity of the SEM Tree algorithm. There are two essential variables whose influence on the time complexity is of interest: The number of observed samples N and the number of split candidates, that is, dichotomous covariates. We denoted the number of these candidate covariates with M. During the generation of a SEM Tree, a large number of models are fitted to data. Therefore, we have to estimate how often the optimizer is called and to what extent the run time of the optimizer depends on M and N. Generally, the optimizer for parameter estimation can be regarded as a black box that estimates the model parameters. For a realistic approximation of the runtime, we have to consider that the optimizer will likely use a numerical procedure to find the maximum likelihood estimate. The log-likelihood formulation for data without missing values (cf. Equation 3.1.8) can be evaluated in constant time. Later on, the FIML (*Full Information Maximum Likelihood*) fit function will replace the ML fit function because it is able to handle missing values in the observed variables. A deeper treatment of missing values in data sets follows below. The evaluation of the FIML log-likelihood is feasible in $O(N)$ because it sums up the likelihoods of each individual observation. The number of iterations of the optimizer generally does not depend on N; on the contrary, if data is normally-distributed, a larger N might provide a more stable estimate of the covariance matrix and expectations vector and even reduce the number of steps of the optimization process.

For a time complexity analysis, we have to determine how often the optimizer is called in the process of generating a SEM Tree. During split candidate evaluation in each node, the pre-split model is fitted once. For each dichotomous covariate, the post-split model has to be evaluated once, that is, M calls to the optimizer are executed for the evaluation of the post-split models. We expect that each selected split variable separates the observations in two halves during SEM Tree generation. Under this assumption, the depth of the tree will be bounded by $O(\log(N))$. A second bound to the depth of the tree is given by $O(M)$. Taken together, we obtain the following time complexity for SEM Trees. When employing the likelihood function F_{ML} for data without missing values:

$$T_{ML}(N, M) = O\left(MN + M \cdot 2^M\right)$$

When using the likelihood function for data sets including missing data:

$$T_{FIML}(N, M) = O\left(N^2M + NM \cdot 2^M\right)$$

In typical research situations, missing data will be present and the number of binary covariates will typically exceed the logarithm of the number of observations. For example, in psychological data sets $N = 1000$ already constitutes a large sample size. In this example, as soon as more than $\log_2(1000) \approx 10$ covariates are included, the run time will be determined linearly by the number of dichotomous covariates and quadratically by the number of observations.

3.4.4 Global Parameter Restrictions

A particular strength of SEM is the possibility of algebraic restrictions on models to incorporate specific hypotheses about the data. For example, in a LGCM, we could require measurement errors to have the same variance over different occasions of measurement. The likelihood ratio test offers statistical means to significantly reject restrictions if the sampled data contradict them. Restrictions on SEM Trees help to test hypotheses, reflecting the substantive questions of the researcher. It is plausible that a researcher would want to include such restrictions in SEM Tree analyses. In a typical situation, a certain variance, covariance, or regression is assumed to be equal in all groups but should still be freely estimated. This restriction cannot be maintained only at the local level of split evaluations since this would imply that all currently grown leaves had to be evaluated as a large multi-group model with additional equality constraints representing the parameter restrictions, in order to determine each local split candidate. This again would mean that the order in which branches are traversed during split evaluation would influence the evaluation of split candidates. A true *global restriction* is only achieved if the respective parameters are estimated from the observed data before the tree is grown and are subsequently treated as constants. Inevitably, all resulting leaf models will have an equal estimate on the globally restricted parameter.

3.5 Generalizability and Evaluation of SEM Trees

3.5.1 Attribute Selection Error and Overfitting

The following observation is crucial for the process of selecting between competing models: Increasing the degrees of freedom of a model by introducing additional parameters will increase

the predictive performance of the model on the given data set that was used to estimate the parameters. However, with every additional degree of freedom, the model is prone to represent non-systematic sample fluctuations. Eventually, this leads to a worse performance on an independent sample from the same distribution. This notorious problem is referred to as *overfitting* and many remedies are offered by the literature. The most common approach is to use cross-validation (Stone, 1974) to obtain estimates of the expected likelihood of new observations under the model or to use formal penalizations of model complexity (Burnham & Anderson, 2002), which is commonly referred to as regularization (cf. Bishop, 2006).

I have previously introduced a criterion for choosing a candidate split based on comparison of the log-likelihood ratio of a model before and after a split, and accepting this split only if this value exceeds a threshold based on a known distribution of this statistic under the null hypothesis of an uninformative split. The maximum-χ^2 criterion chooses the split candidate with the maximum increase in likelihood and bases candidate acceptance on the known distribution of the individual test statistics under the null hypothesis. However, this criterion is plagued by a popular fallacy. In fact, by choosing the maximum of a set of statistics, the resulting distribution is no longer the same as the distribution of the individual statistics. In the following, I will analyze this problem and offer remedies.

Maximum-χ^2 Distribution

Suppose a set of m dichotomous covariates is examined for a particular node. A total number of m likelihood ratio tests is performed, comparing the pre-split model to the m post-split models according to the respective covariate splits. Each likelihood ratio test statistic is χ^2-distributed with equal degrees of freedom. Previously, we have introduced the maximum-χ^2 criterion for candidate selection, which selects the covariate whose test statistic has the maximum value and compares this against the critical value of a χ^2-distribution. Formally, we base an inference on the value of a maximally selected random variable

$$\Lambda_{max} = \max(\Lambda_1, \Lambda_2, \ldots, \Lambda_m)$$

with the m individual test statistics $\Lambda_i \sim \chi^2$.

In the following, I will analyze how this maximally selected random variable is distributed and how we can find corrections for our previous selection criterion for potential split candidates.

Bonferroni-Correction

In the hypothesis testing framework of the maximum-χ^2 criterion, the α level is used to determine a critical value x, such that the probability that a random draw from the distribution is larger than x is equal to α, formally $Pr(\Lambda_i > x) = \alpha$. This is only true if we examine a single test statistic. To find an adjusted α level that truly reflects the probability of not rejecting a false null hypothesis for the maximally selected random variable Λ_{max}, we reformulate the problem as: What is the probability that one or more of the independent random variables Λ_i exceeds the critical value associated with α under the null hypothesis. This is equivalent to finding $\alpha' = Pr(\Lambda_{max} > x)$. The following derivation is generally known as the *Bonferroni correction* (see, e.g. Jensen & Cohen, 2000).

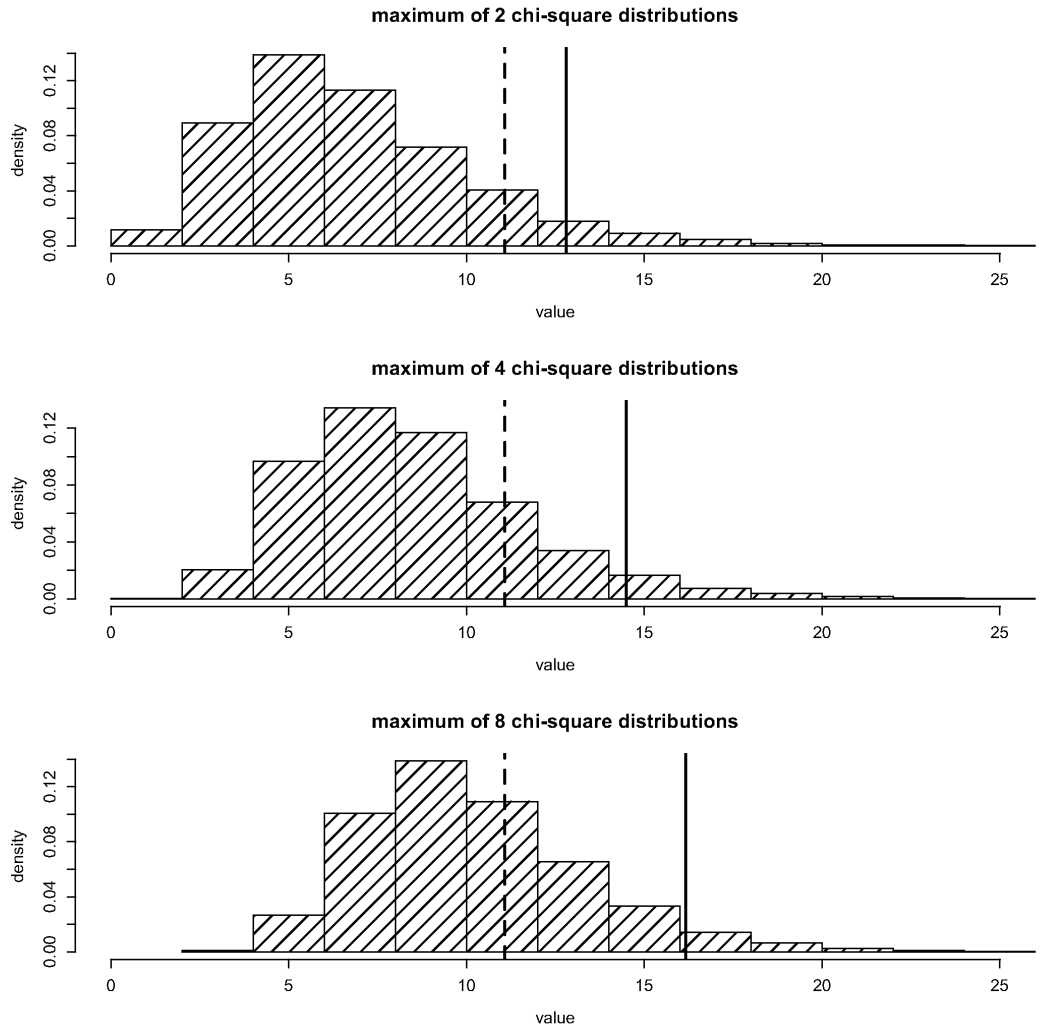


Figure 3.5.1: This figure illustrates the distributions of a maximally selected test statistic $\Lambda_{max} = \max(\Lambda_1, \ldots, \Lambda_m)$ based on different numbers m of underlying Λ_i, which are independently and identically χ^2-distributed with 5 degrees of freedom. This corresponds to the test statistic that is obtained when evaluating candidate splits on a SEM with 5 free parameters. The upper distribution arises as a maximum of 2 variables, the middle distribution as a maximum of 4 variables, and the bottom distribution as a maximum of 8 variables. The dashed line indicates the critical value for a type-I error of $\alpha = 0.05$, based on a single χ^2-distribution. The solid line indicates the true critical value. With increasing m, the dashed critical value constitutes an increasingly liberal choice that leads to an inflated type-I error rate if falsely employed to determine the critical value.

Theorem 56. *During the evaluation of m split candidates in a SEM Tree, let* $\Lambda_i, i \in [1 \dots m]$ *be the log-likelihood ratio statistics and let* Λ_{max} *be the maximally selected statistic* $\Lambda_{max} = \max(\Lambda_1, \dots, \Lambda_m)$ *representing the best split candidate. If the* Λ_i *are independent, the effective type-I error rate is* $\alpha_{max} = 1 - (1 - \alpha)^m$.

Proof.

$$\begin{aligned}
\alpha_{max} &= Pr(\Lambda_{max(m)} > x) \\
&= 1 - Pr\left(\Lambda_{max(m)} \leqslant x\right) \\
&= 1 - [Pr(\Lambda_1 \leqslant x \wedge \Lambda_2 \leqslant x \wedge \dots \wedge \Lambda_m \leqslant x)] \\
&= 1 - [Pr(\Lambda_1 \leqslant x)\, Pr(\Lambda_2 \leqslant x) \dots Pr(\Lambda_m \leqslant x)] \\
&= 1 - [Pr(\Lambda_i \leqslant x)]^m \\
&= 1 - [1 - Pr(\Lambda_i > x)]^m \\
&= 1 - (1 - \alpha)^m
\end{aligned}$$

□

This result shows that the probability of making a type-I error, that is, falsely choosing a non-informative covariate as split candidate, increases drastically with increasing numbers of tested covariates. For example, significance testing of fourteen independent and identically distributed test statistics with $\alpha = 5\%$ will yield a significant result in one of them in more than 50% of all cases

$$P(error) = 1 - (1 - 0.05)^{14} \approx 0.512$$

If we generate a SEM Tree with the maximum-χ^2 criterion and 14 uninformative covariates, an uninformative covariate is chosen as an informative split in every second run of the tree. This is, of course, an intolerable situation. Figure 3.5.1 further illustrates the multiple testing problem of this approach. The graph shows observations of maximally-selected test statistics under the null hypothesis of uninformative splits, with 2, 4 and 8 split candidates. With an increasing number of covariates under examination, the bias of Λ_{max} increases and the more likely it becomes that values of Λ_{max} are observed that are larger than a critical value which is falsely based on an individual χ^2-distribution. Under the assumption of independence of the covariates, we can derive a corrected α level to determine the critical value:

Corollary 57. *Let* $\Lambda_{max} = \max(\Lambda_1, \dots, \Lambda_m)$ *be as above. Let* α *be a chosen significance level for the maximally selected* Λ_{max}. *The critical value x for the statistic is corrected for the multiple tests under the assumption of independence of the m statistics by*

$$Pr(\Lambda_{max} > x) = 1 - \sqrt[m]{1 - \alpha}$$

This allows us to define a Bonferroni-corrected variant of the maximum-χ^2 criterion for split candidate evaluation.

Definition 58 (Bonferroni-Corrected Maximum-χ^2 Criterion)**.** Let Λ_i be m log-likelihood ratios obtained from evaluating m candidate splits. Let z be the index that finds Λ_z with the maximum

value. For a chosen significance level α, the *Bonferroni-corrected maximum-χ^2 criterion* for covariate selection selects the covariate with index z, if

$$\lambda_z > crit, \quad P\left(\chi^2_{df} > crit\right) = 1 - \sqrt[m]{1-\alpha}$$

In the literature, this correction is most often given as the following approximation via the Bonferroni inequality, which is found in most statistics textbooks, e.g., in Bortz and Schuster (2010), to find the effective α level by dividing the initially chosen α level by the number of tests

$$1 - \sqrt[m]{1-\alpha} \approx \frac{\alpha}{m}$$

This criterion reflects a correction for the problem of the multiple comparisons. However, it does not take into account that the m variables were generated from $k < m$ non-dichotomous covariates according to the rules described in Section 3.4.2. Covariates with a higher number of induced dichotomous covariates therefore have a higher chance to be selected under the null hypothesis than those with a lower number. This can be corrected by a nested correction that accounts for the total number of original covariates and the number of induced dichotomous covariates for each original covariate:

Theorem 59. *Let α be a chosen type-I error rate. The nested correction of the type-I error rate correcting for m original covariates with $k_i, i \in 1..m$ induced dichotomous covariates each is given as*

$$\alpha'_i = \left(1 - (1-\alpha)^{1/(m \cdot k_i)}\right) \approx \frac{\alpha}{m \cdot k_i}$$

Proof. We correct each variable by the number of original covariates m and the number of induced covariates of the respective original covariate k_i. Two corrections of the type that were introduced in Corollary 57, lead to the following correction

$$\begin{aligned}
\alpha'_i &= 1 - \sqrt[m]{1 - \left(1 - \sqrt[k_i]{1-\alpha}\right)} \\
&= 1 - \sqrt[m]{\sqrt[k_i]{1-\alpha}} \\
&= 1 - (1-\alpha)^{1/(m \cdot k_i)} \\
&\approx \frac{\alpha}{m \cdot k_i}
\end{aligned}$$

□

The above correction advocates individual significance levels for covariates. This renders the maximum selection of covariates unfeasible. Previously, we have selected the maximum test statistic among the Λ_i and have compared it against a critical value derived from a corrected α level. Now, we have to individually correct the test statistics Λ_i first with their individual α_i and then select a maximum if at least one of them is significant after the correction. Otherwise, there is no significant split. The α level and a critical value x are related by $Pr(\Lambda > x) = \alpha$.

If we observe m realizations of the test statistics $\lambda_1, \ldots, \lambda_m$, we can find the probability of observing a value like this or larger as the p-value by $Pr(\Lambda_i > \lambda_i) = p_i$. Instead of correcting the α threshold for each Λ_i to obtain individual critical values, we equivalently correct the p-value according to the same logic as before. This enables a comparison of differently distributed statistics on the same scale. Thus, selecting the minimum p_i after correction replaces selecting the maximum λ_i.

Definition 60 (Nested Bonferroni-Corrected Maximum-χ^2 Criterion)**.** Let λ_i be realizations of the test statistics Λ_i and let p_i be the corresponding p-values $Pr(\Lambda_i > \lambda_i) = p_i$. Let the corrected p-values for λ_i be $p'_i = \frac{p_i}{m \cdot k_i}$. Let z be the index of the test statistic with minimum p'_i. The *nested Bonferroni-corrected maximum-χ^2 criterion* is

$$p'_z < \alpha \cdot m \cdot k_i$$

The nested or non-nested Bonferroni-corrected maximum χ^2-criterion is a conservative correction. It is fundamentally based on the independence of covariates. Whenever covariates are correlated, the Bonferroni-correction overcorrects. However, the way we constructed dichotomous variables from the original multi-categorial, ordinal, or continuous covariates suggests that there is indeed a correlational structure between them. Summarizing, using no correction renders the significance test for candidate evaluation useless. Applying the Bonferroni-correction corrects for the multiple testing problem under the assumption of independence of the covariates. The nested Bonferroni-procedure is constructed in a way to guarantee unbiased covariate selection under the null hypothesis if the same independence assumption holds. In reality, this independence assumption might be violated to varying degrees. Therefore, the Bonferroni correction is discouraged for SEM Trees. In the following section, a further method for variable selection is discussed that offers remedy.

Cross-Validation

A remedy for the problem of multiple comparisons described above is the evaluation of hypotheses against a disjoint evaluation set. This procedure was recognized early on as a possible method for the evaluation of a statistic when an unbiased estimate is required. Mosier (1951) and many followers separated their samples in two parts that were treated, from then on, as two independent and identically distributed samples of the same population. In practice, the original data set is simply split into two sets, usually referred to as a *training set* and a *validation set*. In model selection, a set of competing models is generated using the training set and subsequently evaluated using the independent validation set. This latter set is also known as the "hold out sample", "calibration sample", or "validation sample". Foremost for small and medium size data sets, the arbitrary split of the original data set into a training and a validation set and the resulting variance of the statistic based on the random choice poses a practical problem. An extension to compensate for this, which is attributed to Stone (1974), is a set of techniques to be categorized as cross-validation (CV). In cross-validation, the population sample S is divided in k disjoint sets S_i also called *folds*. Cross-validation is iterated k times. In the i-th iteration, a selected model is estimated on the training set $S \backslash S_i$ and evaluated on the independent test set S_i. The k scores that are produced by cross-validations are usually averaged to a combined score, which forms an estimate of the expected performance of the model. A

cross-validation using k folds is called a *k-fold cross-validation*. This method is widely applied in machine learning and statistics. Generally, a ten-fold CV is considered a good choice for the bias variance trade-off of the estimator (Kohavi, 1995), that is, the variance of the estimate and the systematic error of the estimate are both low in comparison to other choices of k. A further cross-validation strategy often found in the literature is *leave-one-out* CV, often referred to as LOO-CV. This strategy is the limit case of k-fold CV and sets the fold size to 1 and thereby k to the size of the sample. For SEM Trees, this strategy seems the most impractical as it would maximally increase the number of calls to the optimizer. In line with Kohavi (1995), a reasonable choice seems to be five-fold or ten-fold CV for SEM Trees. Larger numbers of folds increase the number of samples that are used in the estimation process because $N \cdot (k-1)/k$ samples are used to estimate parameters. Increasing the number of folds leads to more stable parameter estimates and reduces the variance of the estimate. At the same time the computation time is increased by the choice of the number of folds. The run-time of the algorithm is linearly scaled by the number of folds. Although this does not change the asymptotic run-time analysis, it could be a crucial aspect for consideration in practical applications.

When using CV for the evaluation of covariate splits, we obtain estimates of the expected log-likelihood ratio of seeing a new data point under the split versus the un-split model. We obtain a new test statistic for each split candidate in the CV scheme:

Theorem 61. *Let $D_1, D_2, \ldots, D_k$ be a k-partition of D with $D_i \cap D_j = \emptyset, i \neq j$, $\forall i : D_i \neq \emptyset$ and $D_1 \cup D_2 \cup \ldots \cup D_k = D$. Furthermore, let $D_{i,left}$ and $D_{i,right}$ be the left and right data set of D_i according to a split candidate $c \in C$. Let $\hat{\theta}_D$ be the parameter estimate obtained by minimizing $-2LL\left(D_i|M_D\right)$. The k-fold cross-validation estimate of the expected log-likelihood ratio of the pre- and post-split model is*

$$\begin{aligned}\Lambda_{CV} &= \sum_{i=1}^{k} -2LL\left(D_i|M\left(\hat{\theta}_{D\setminus D_i}\right)\right) \\ &\quad +2LL\left(D_{i,right}|M\left(\hat{\theta}_{D\setminus D_{i,right}}\right)\right) \\ &\quad +2LL\left(D_{i,left}|M\left(\hat{\theta}_{D\setminus D_{i,left}}\right)\right)\end{aligned}$$

Proof. The proof follows from the definition of cross-validation and the theorems about the likelihoods and nestings of pre-split and post-split models. □

Using this formulation, a simple decision rule can be adopted: Whenever the statistic is larger than zero, the expected log-likelihood ratio of the pre-split and the post-split model is larger than zero. In other words, on average across all possible data sets of this size sampled from the population, observing the data is more likely under the candidate post-split model than under the pre-split model.

Definition 62 (Cross-Validation Selection Criterion)**.** Let $\Lambda_{CV,max}$ be the maximally selected test statistic from a set of cross-validated log-likelihood ratios of pre-split and post-split models. The Cross-Validation Criterion for split candidate evaluation is: $\Lambda_{CV,max} > 0$

In order to stabilize the CV evaluation, *stratified cross-validation* (Breiman et al., 1984) is used. Stratified CV constructs the folds in a way that the folds are stratified to a target

variable. In particular, the distribution of the values of the split covariate should approximate the distribution in the data set. In SEM Trees, it must be ensured during the evaluation of each split candidate that the distribution of the covariate is reflected in each fold while still randomly assigning cases to folds. Kohavi (1995) reports that stratification can reduce bias and minimize variance of the CV estimator.

A further method should be mentioned in the context of multiple comparison problems (Jensen & Cohen, 2000). The very distribution of interest is the distribution of the chosen test statistic under the null hypothesis which assumes that the value of the test statistic is merely due to sample fluctuations and that indeed both models under investigation are close enough representations of the data. If the exact distribution is not known, e.g., if the distribution relies on asymptotic theory and sample sizes are small or the distribution is a-priori unknown, there are methods to construct an empirical distribution of the test statistics under the null hypothesis. By randomly permuting the binary class labels of the covariate under investigation, we can obtain a likelihood ratio under the null hypothesis. By repeating this process we obtain an empirical distribution of the likelihood ratios under the null hypothesis. This requires fitting a large number of models with shuffled data sets. In most practical settings, this will require too much computation time.

In this section, I have discusses criteria for determining slit candidate selection. In the following section, I will present a re-interpretation of these criteria from an information-theoretic perspective.

3.5.2 Likelihood Ratio Tests as Information Criterion

The stopping criteria for SEM Trees have been developed based on a statistic whose distribution is known under the null hypothesis that a split candidate variable provides no information about differences with respect to the parameters in a template SEM. Interestingly, this criterion can also be interpreted as an information-theoretic criterion, in particular, it leads to the selection of covariates that maximize *information gain*, that is, the reduction in entropy of the predicted observations when considering a split of the tree. Therefore, SEM Trees essentially maximize the same criterion as decision trees that employ the maximization of information gain, like ID3 (Quinlan, 1986). The difference between the two is that ID3 predicts categorial outcomes, whereas SEM Trees predict continuous observations based on a model. ID3 maximizes the information gain when using a split candidate to partition a categorial outcome variable. The following section formally derives this relation.

We have previously used the notion of entropy $H(X)$ of a random variable X. Usually, the entropy is estimated from a finite sample drawn from this random variable. In the following, we denote estimates of the entropy of discrete samples $x_1, \ldots, x_N$ of X by $\hat{H}(x_1, \ldots, x_N)$, whereas the entropy of X is denoted by $H(X)$.

The information gain of X knowing Y is defined as the estimated reduction of the entropy of X when knowing Y, formalized as $Gain(x_1, \ldots x_N, y_1, \ldots, y_N) = \hat{H}(x_1, \ldots x_N) - \hat{H}(x_1, \ldots x_N | y_1, \ldots, y_N)$ (Cover & Thomas, 1991). This information gain, as used in ID3, is the reduction in entropy of the target variable after splitting the target variable according to the discrete split candidate variable. Let the target variable X be a discrete random variable. Let the split candidate variable Y be a discrete random variable, whose elementary outcomes can be retrieved by $Values(Y)$. Let N be the number of observations of the target variable.

The information gain about a variable X, knowing the state y of variable Y is defined as the decrease of the entropy estimate $\hat{H}(x_1, \ldots, x_n)$ of X for known values of Y

$$Gain\left(x_1, \ldots, x_n, Y\right) = \hat{H}\left(X\right) - \sum_{y \in Values(Y)} \frac{N_y}{N} \hat{H}\left(x_1, \ldots, x_n | Y = y\right)$$

SEM Trees operationalize the choice of a split candidate by the maximization of a log-likelihood ratio statistic. Interestingly, we can express the likelihood ratio as a variant of the information gain criterion. This links the log-likelihood ratio criterion back to the same fundamental rule of maximizing information gain. Indeed, by maximizing the likelihood ratio criterion, we are maximizing the information gain in the model-predicted distribution of the observations with respect to split candidates Y.

Lemma 63. *Let M be a SEM. Let X be a random variable and $x_1, \ldots, x_N$ a finite set of samples of X. The negative two likelihood is a linear function of the entropy estimate of the model-predicted distribution of M*

$$-2LL\left(x_1, \ldots, x_N | M\right) = 2N \cdot \hat{H}\left(x_1, \ldots, x_N | M\right)$$

Proof. The likelihood is an estimator of the entropy of a continuous multivariate random variable X. Let $x_1, \ldots, x_N$ be N samples of X, and let M be a multivariate SEM. We recall that the negative two log-likelihoods of the observations is the sum over the logarithm of the likelihood function

$$-2LL\left(x_1, \ldots, x_N | M\right) = -2 \sum_{i=1}^{N} \log\left(L\left(x_i | M\right)\right)$$

Samples x_i are drawn according to the distribution of X. For $N \to \infty$, the likelihood converges to the entropy of the distribution that is predicted by the model M (cf. Cover & Thomas, 1991)

$$\begin{aligned} \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \log\left(L\left(x_i | M\right)\right) &= \int X|M \log\left(X|M\right) dx \\ &= -H\left(X|M\right) \end{aligned}$$

Equivalently[1], we can write the fundamental relation between the likelihood function and the entropy of the model-implied distribution as

$$-2LL\left(x_1, \ldots, x_n | M\right) = 2N \cdot \hat{H}\left(x_1, \ldots, x_N | M\right)$$

□

[1] Note that it in maximum likelihood settings, the logarithm is often to base e, whereas in information-theoretic settings, logarithms are often base 2 representing the unit of bits. To account for this discrepancy, an additional multiplicative correction factor has to be added that however does not change the central relation between both measures.

Theorem 64. *Let M be a SEM and D be a data set. Let LLR_Y be the log-likelihood ratio statistic obtained by comparing a pre-split and post-split model with respect to the split candidate, represented by the random variable Y. The likelihood ratio statistic can be expressed as a linear function of the information gain about the model-predicted distribution $L(D|M)$ when knowing the states of Y.*

$$LLR_Y(D|M) = 2N \cdot Gain(L\left(D|M\right), Y)$$

Proof. We begin by considering the information gain for a binary variable Y, representing a split candidate in a binary SEM Tree.

$$\begin{aligned}
Gain(x_1, \ldots, x_n, Y) &= \hat{H}\left(x_1, \ldots, x_n\right) - \sum_{y \in Values(Y)} \frac{N_y}{N} \hat{H}\left(x_1, \ldots, x_n | Y = y\right) \\
&= \hat{H}\left(x_1, \ldots, x_n\right) - \frac{N_{left}}{N} \hat{H}\left(x_1, \ldots, x_n | Y = left\right) \\
&\quad - \frac{N_{right}}{N} \hat{H}\left(x_1, \ldots, x_n | Y = right\right) \\
2N \cdot Gain(x_1, \ldots, x_n, Y) &= 2\hat{H}\left(x_1, \ldots, x_n\right) \\
&\quad - 2N_{left} \hat{H}\left(x_1, \ldots, x_n | Y = left\right) - 2N_{right} \hat{H}\left(x_1, \ldots, x_n | Y = right\right)
\end{aligned}$$

Using Lemma 63, we can rewrite the information gain based on the following observation that $\hat{H}\left(x_1, \ldots, x_n | Y = left\right)$ describes the partition of the observations of a data set D into the left data set and $\hat{H}\left(x_1, \ldots, x_n | Y = right\right)$ describes the right data set. Therefore, we can conclude that

$$\begin{aligned}
2N \cdot Gain(x_1, \ldots, x_n, Y) &= -2LL\left(D|M\left(\theta\right)\right) \\
&\quad + 2LL\left(D_{left}|M\left(\hat{\theta}_{left}\right)\right) + 2LL\left(D_{right}|M\left(\hat{\theta}_{right}\right)\right) \\
&= LLR_Y\left(D|M\right)
\end{aligned}$$

□

Corollary 65. *When choosing a split candidate that maximizes the log-likelihood ratio of a pre-split and a post-split model, in the process of SEM Tree generation, the split variable that maximizes the information gain between the model-predicted distribution and the split variable is chosen.*

The expected value of the information gain is the *mutual information* measure, a general measure of statistical dependence between two random variables (Cover & Thomas, 1991). Mutual information measures how similar a joint distribution of two random variables is to the product of their marginal distributions.

The mutual information is a measure of the degree of statistical dependence. If the target variable and the split candidate variable are statistically independent, that is, the product of their marginal distributions is equal to their joint distribution, the mutual information is zero.

In analogy to the previous reasoning, we can conclude that by maximizing an estimator of the expected likelihood ratio when using cross-validation for candidate selection, we are choosing split candidates that maximize the mutual information between the model-predicted distribution and the split variable.

Corollary 66. *When choosing a split candidate that maximizes the expected log-likelihood ratio of a pre-split and a post-split model, in the process of SEM Tree generation, the split variable that maximizes the mutual information $I(X,Y)$ between the model-predicted distribution X and the split variable Y is chosen, with*

$$I(X,Y) = \int_X \sum_{y \in Values(Y)} Pr(x,y) \log\left(\frac{Pr(x,y)}{Pr(x)\,Pr(y)}\right) dx$$

These elaborations show that the evaluation of split candidates in SEM Trees is reasonable from both a statistical and an information-theoretic point of view and, finally, that the evaluation of split candidates by the log-likelihood ratio and the classic information gain criteria are rooted in the same fundamental concepts.

3.5.3 Pruning

Decision tree algorithms are sometimes built in a two-step procedure: (1) Given a training sample, a tree is constructed. Since some decision tree algorithms are notorious for their tendency to overfit data (Esposito, Malerba, Semeraro, & Kay, 2002), (2) a pruning stage is established to cut back the tree to a smaller tree with less leaves, in the hope of achieving a higher generalization performance. An overview of different pruning algorithms was provided by Esposito et al. (2002). In the following, I adapt *Reduced Error Pruning* by Quinlan (1987) to SEM Trees. Therefore, we replace the classification error rate of a tree as a performance measure by the likelihood function. Generally, the pruning algorithm cuts off leaves of the tree as long as its performance on an independent set increases, which serves as an estimate of its generalizability. Quinlan (1987) uses the number of misclassified items as a performance measure. SEM Trees use the negative two log-likelihood as a performance measure. This allows a straightforward adoption of the method because both measures, negative log-likelihood and error rate, indicate improvements by decreasing numbers and worsening by growing numbers. In the following, a variant of reduced error pruning adapted to SEM Trees is described.

Pruning algorithms are based on the availability of two data sets. This requirement is typically met by separating a single observed data set D into a non-empty training set and a non-empty pruning set, such that $D = D_p \cup D_T, D_p \cap D_T = \emptyset$. The tree is built with the training set either to a large size that is likely to overfit the data. If possible, the tree is grown to the full size. If there are too many covariates and observations, the tree can heuristically be grown to a large size that is justifiable in terms of computation time. In a second stage, the separate pruning set is used to determine which leaves of the full tree were overfit to the training set and represent random fluctuations rather than systematic differences. The heuristic is simple. As long as pruning the tree increases its performance on the independent pruning set, the generalizability of the tree is expected to increase. In contrast, when a sub-tree is pruned to generalize to the pruning set, we expect its performance on the pruning set to decrease since we have pruned important information. This pruning strategy is implemented in an iterative

procedure that keeps a set of pruning candidates and removes sub-trees from the full tree as long as the performance of the pruned tree on the pruning set increases. The set of pruned trees $\{T'|T' \preceq T\}$ in each step is generated by transforming a different, single inner node to a leaf node for each of the sub-trees. The sub-tree with the largest reduction of performance is selected for pruning and removed. This procedure is repeated until there is no feasible pruning left that increases performance on the pruning set.

Quinlan (1987) adds another constraint to the set of pruning candidates. Each pruning candidate T' is considered for pruning only if it contains no sub-tree $T'' \prec T'$ with a better performance, that is, if it fulfills $-2LL(T') < -2LL(T'')$. By construction, SEM Trees are built in a way that $T_1 \preceq T_2 \Rightarrow -2LL(T_1) \leq -2LL(T_2)$ since splitting nodes is only continued if it increases the log-likelihood ratio, or equivalently significantly increases the likelihood of the model before the split in comparison to the model after the split. Adhering to this rule, this reduces the set of pruning candidates in each iteration to inner nodes that only have leaves as children. If a tree T is an appropriate representation of the population, we expect its negative two log-likelihood to be smaller than in any of the pruning candidates T'. However, if the negative two log-likelihood in any of the T' is smaller than T, that is, if $-LL(D_P|T') < -LL(D_P|T)$, the un-pruned tree seems to have overfitted to D_T and pruning is continued.

The main problem with pruning is a significant reduction of the data set that can be used to build the original tree. For a data set with small sample sizes, pruning is not feasible and it seems more appropriate to rely on split candidate evaluation criteria that guarantee good generalizability. When using CV as the candidate evaluation criterion, the danger of overfitting the training set is low in principle. Nonetheless, pruning can be an interesting technique when additional data becomes available at a later stage of data analysis. For example, the same psychological test may have been replicated in a different experiment. In that case, the additional data set can be used to detect whether the previous tree overfitted the initial sample. Moreover, stopping rules for decision trees may stop growing the tree too early with a subsequent split after the current stopping point bearing information still possible. Pruning is a method to confront this situation. By growing the tree without a stopping criterion and then pruning it back, the problem of too early stopping can be mitigated.

3.5.4 Tree Stability and Forests

The tree growing algorithm proceeds in a greedy fashion, i.e., locally optimal choices are taken at each step of the growing process. The researcher has to keep in mind that the resulting tree is not necessarily the single best tree. Particularly, a tree can be susceptible to marginal changes of the data set, resulting in different choices for node splits and eventually structurally different trees. Building a set of trees from bootstrap samples sheds light on the stability of a tree (Strobl, Malley, & Tutz, 2009). Resampling methods generate a set of variants of an original data set by randomly sampling observations from the data set (Efron, Tibshirani, & Tibshirani, 1993) and therefore belong to the class of Monte Carlo methods. Calculating statistics on the resampled data sets provides insights on distributional properties of test statistics. Instead of obtaining only a point estimate of a statistic on the original data set, the estimate of the distribution provides a basis for confidence intervals and statistical tests. One such method is *bootstrapping*. Bootstrapping was first described by Efron (1979). A bootstrap sample of a data set of size N is obtained by drawing N observations from the original data set uniformly and

with replacement. Therefore, we obtain a sample of the same size as the original sample. For large sample sizes, a bootstrap sample of size B represents a substitute for B independent draws from the true population distribution. The probability for an observation not to be chosen in a bootstrap sample is $\left(1 - \frac{1}{N}\right)^N \approx \frac{1}{e} \approx 0.368$. Thus, we expect to choose about two thirds of the data set for the bootstrap sample while one third, sometimes referred to as the *out-of-bag* sample (OOB), forms a separate set that is independent from the bootstrap sample. Thus, a common method to evaluate classification algorithms is to use the bootstrap sample as a training set and the OOB as an independent test set. By drawing a large number of bootstrap samples and repeatedly training and testing the algorithm, a distribution of the algorithm's accuracy is obtained. In the same manner, bootstrap samples can be used to generate a set of trees. If the random variability exceeds the systematic variability in the data set, this will lead to trees differing in their composition. Nevertheless, such a set of distinct trees helps the researcher to gain insights into which variables describe differences in the data set by simply looking at the selected variables. Thinking this through rigorously, one ends up with Structural Equation Forests, that is, an ensemble of SEM Trees that were built from resampling the original data set. Indeed, Breiman (2001) introduced *Random Forests* as ensembles of decision trees, and interesting measures of variable importance exist that could be applied to SEM Forests straightforwardly. A popular method to determine variable importance is the permutation accuracy importance (cf. Strobl, Boulesteix, Zeileis, & Hothorn, 2007). This measure is based on the notion that the more important a split variable in a tree is for the prediction of a tree, the more the tree's performance is affected when the association of the predictor and the target variable is broken. Concretely, a tree is trained on a bootstrap sample and its accuracy in the classification task is calculated on the OOB sample. Then, one by one, each variable that serves as split candidate is randomly permuted and the drop in the tree's accuracy is calculated. If the variable contains only little information, the permutation will affect the accuracy only slightly. In contrast, an important variable that is perturbed will strongly affect the tree's performance. The resulting decrease in accuracy or likelihood for SEM Trees is then aggregated for each variable across all trees. Finally, the variables can be ranked by their average drop in accuracy. The higher the drop is, the higher the permutation accuracy importance. This method is computationally quite intensive. For example, Lunetta, Hayward, Segal, and Van Eerdewegh (2004) examined genetic markers predicting a categorial outcome with Random Forests and report that between 100 and 1000 trees are typically required in a Random Forest.

3.5.5 Traditional Fit Indices and SEM Trees

Traditionally, the evaluation of goodness-of-fit in SEM is a highly debated matter and many researchers have contributed an abundance of fit indices with numerous modifications. In Section 3.1.4, the most commonly found fit indices were presented. In relation to SEM Trees, two questions arise naturally: How should fit indices be used for SEM Trees? Do fit indices indicate whether SEM Trees can offer additional information?

The first questions is easily answered. Since SEM Trees can be conceived as a multi-group SEM, fit indices for multi-group models can be used straightforwardly. Also, traditional fit indices for each node model can be inspected to judge goodness-of-fit for each node of the tree.

By sketching an extreme case, an illustration will be given that common fit indices can fail to detect that there might be more to discover in sample data. Consider a simple SEM with

two observed variables X_1 and X_2. Suppose there are two groups in the populations that we have sampled from. For one group, observations are centered at the origin and observations are negatively correlated

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & -0.2 \\ -0.2 & 1 \end{bmatrix}$$

whereas the other group is distinguished by a shift in the mean and a strong positive correlation of observations

$$\mu_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Let our sample be a mixture of both groups in the population. Assuming equal group sizes, the resulting distribution is

$$\mu_{1+2} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \Sigma_{1+2} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

Contours of these distributions are illustrated in Figure 3.5.2. Suppose that we model the sample with a saturated model. Since the resulting distribution is perfectly normal, the model-implied covariance matrix will be nearly identical to the sample covariance matrix. Therefore a measure of the residual variance like SRMR will tend to go to zero. Also, since the model will be superior to an independence model in all cases because Σ_{1+2} has non-zero off-diagonal elements, incremental fit indices will tend to favor a two-variable saturated SEM. Absolute fit indices are based on differences of the proposed model to the saturated model and will therefore yield perfect index values for a saturated model by definition. Nevertheless, the model is in fact a mixture of two distributions. Despite the perfect fit indices, a SEM Tree can potentially discover underlying distributional mixtures. Figure 3.5.2 shows an exemplary SEM Tree based on a saturated model that expects three covariance parameters and two expectation parameters.

3.5.6 Validation of SEM Trees

Traditionally, the goodness-of-fit in SEM is a measure based on the observed data sample. The most common measures, e.g., GFI, RMSEA, and others mentioned in Section 3.1.4, are based solely on the observed sample. The use of independent sets for model selection and model evaluation in data analysis is emphasized by many authors (Bishop, 2006; Browne & Cudeck, 1992; Kriegeskorte et al., 2009). I adopt this advice to test the generalization performance of a SEM Tree by partitioning the available data set into two disjoint sets, a training set and a test set. The training set is usually chosen to be larger than the test set to provide enough data for the learning phase, respectively, the tree building phase. A robust estimator of the generalization performance of a SEM Tree is the evaluation of the independent test set on the decision tree. Therefore, the joint probability of observing the test set under the SEM Tree is evaluated against the template SEM fitted to the whole data set, which is equivalent to the SEM Tree pruned back to the root node only. The SEM Tree can be thought of as a flat multi-group model, in which the tree represents a covariate-specific mapping of the observed data to the respective group models. In particular, the model associated with the root node

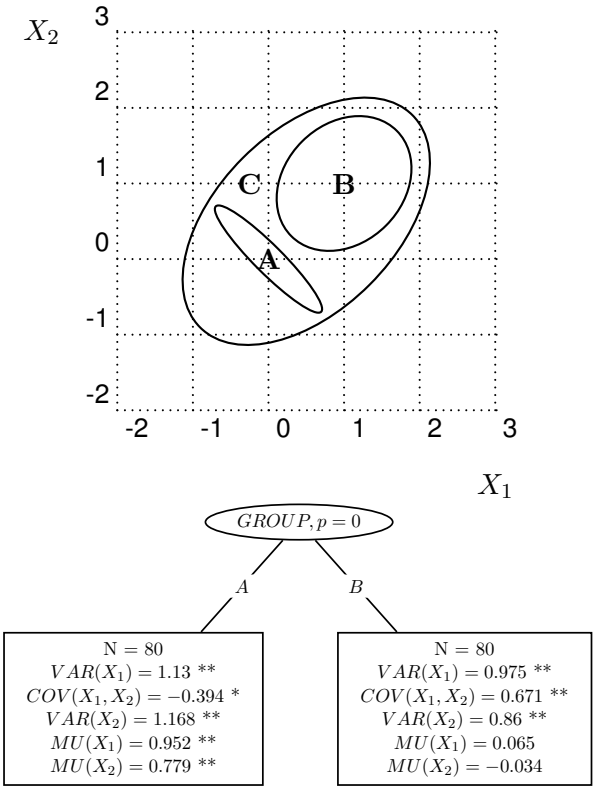


Figure 3.5.2: Utility of SEM Trees despite perfect model fit of the template model. Top: Contour plot of two bivariate normal distributions. The distribution A is negatively correlated, B is positively correlated. C is the respective contour for the sum of the two bivariate distributions, which is again normally distributed. Bottom: A SEM Tree based on a saturated model recovers the structure of some simulated data. The parameter estimates MU indicate the expectations of the respective variable, VAR the variance of the respective variable, and COV the covariance between both.

is an algebraically nested model of the model implied by the SEM Tree. The likelihood ratio between the root and tree model is thus a test statistic which is χ^2-distributed with $(\lambda - 1)df$ degrees of freedom, with λ being the number of leaf nodes and df being the degrees of freedom, i.e., the number of freely estimated parameters of the template model. Significant values of the test statistics reject the null hypothesis that both a single model for the whole data set and the tree-implied model represent the true model. In that case, we are inclined to believe that indeed the more complex tree-implied model is a more appropriate representation of the population.

3.6 Dealing with Missing Values

In applied research, it almost never happens that a data set is collected without missing data. No matter how well the process of data collection is planned, not all data will be available at the end. Particularly in lifespan research, people are sometimes tracked over years and one of the

most common reasons for missingness is the drop-out of participants through relocation, illness, or death. However, a variety of reasons for missingness is capable of surprising the researcher at the end of data collection, starting from illegibility of test results to the unwillingness of certain participants to answer certain questions in a questionnaire. The latter reason states a relationship between the missingness and a variable under investigation, whereas the former example gives reason to believe that the two are independent. According to Rubin (1976), mechanisms of missingness are seen as probabilistic models and can be divided in three categories according to assumptions about their correlational structure with variables under investigation.

Let Y be an observed variable that contains missing values and X be a fully observed variable. Let R be a dichotomous random variable which indicates the occurrence of missingness in Y. If the distribution of missingness depends on observed data but not on missing data, the mechanism of missingness is termed *missing at random* (MAR). If R is independent of any observed variable, it is *missing completely at random* (MCAR). The most adverse case is *not missing at random* (NMAR), whereby the pattern of missingness depends on the unobserved missing values themselves.

Under the assumption of MAR processes, SEM Trees can handle missing values in the observed variables as well as in the covariates. Furthermore, SEM Trees can serve as a tool for model-based imputation of observed variables.

3.6.1 Missing Values in Observed Variables

McArdle (1994) makes a strong point that "structural equation models do not require all variables to be measured on all individuals under all conditions" (p. 409). A common approach to estimate model parameters under missing data is the *Full Information Maximum Likelihood* (FIML) approach. The maximum likelihood fit function as presented so far is not able to handle data sets with missing values since a mean vector and covariance matrix cannot be computed. However, there is a reformulation of the same function that is able to obtain ML parameter estimates under missing values (Finkbeiner, 1979).

The FIML fit function is defined without resorting to an aggregated mean vector and a covariance matrix. This allows the calculation of a likelihood for single observations that arise for unique patterns of missing values in a larger data set

$$-2LL_{FIML}\left(x_1,\ldots,x_N|\mu,\Sigma\right) = N\cdot p\cdot\ln\left(2\pi\right)+\sum_{i=1}^{N}\left[\ln\left|\Sigma_i\right|+\left(x_i-\mu_i\right)^T\Sigma_i^{-1}\left(x_i-\mu_i\right)\right]$$

where Σ_i is the model-implied covariance matrix with rows and columns deleted according to the pattern of missingness in the i-th observation x_i, and likewise μ_i is the model-implied mean vector with elements deleted according to the respective pattern of missingness.

The implementation of a model with missing data can be integrated into the existing framework in an elegant way. The model is decomposed into a set of sub-models according to the patterns of missingness in the data. A pattern of missingness is simply a vector of Boolean values representing the missingness of the according variable. All data rows with the same pattern of missingness are tied together to form subsets of the original data set.

3.6.2 Missing Values in Covariates

Missing values in covariates can be dealt with by introducing only slight modifications to the likelihood ratio statistic for candidate split evaluation and to the traversal function. First, we modify some previous definitions so that we can incorporate missing values:

Definition 67. The symbol Ø represents a missing value[2]. The set $\mathbb{R}_{\emptyset}$ extends the real numbers by a missing value Ø, $\mathbb{R}_{\emptyset} = \mathbb{R} \cup \{\emptyset\}$. A missing-value data set $D = (O, C)$ has elements $o_{ij} \in \mathbb{R}_{\emptyset}$ and $c_{ij} \in \mathbb{R}_{\emptyset}$.

For a given observation of a data set $(o, c) \in D$, the traversal function traverses the tree according to the decision rules for the values of c that are encoded by the tree. Whenever the traversal function encounters a node that requires a decision on a value c_i that is missing, we cannot decide whether to continue with the left or right sub-tree. In that case, the parameter estimate of the current inner node is returned as the best model for the respective observation o. The modified traversal function that can handle missing covariate values is given in the following definition:

Definition 68. The missing-value traversal function $\phi_{\Upsilon} : {\mathbb{R}_{\emptyset}}^{c} \rightarrow \mathbb{R}^{k}$ of a SEM Tree Υ is defined as $\mathcal{L}_{\theta}(r)$ if Υ has only one node. Otherwise, let r_{left} and r_{right} be the children of r and Υ_{left} and Υ_{right} be the corresponding sub-trees.

$$\phi_{\Upsilon}(x) = \begin{cases} \phi_{\Upsilon_{left}}(x) & x_{\mathcal{L}_N(r)} \in \mathcal{L}_E((r, r_{left})) \\ \phi_{\Upsilon_{right}}(x) & x_{\mathcal{L}_N(r)} \in \mathcal{L}_E((r, r_{right})) \\ \mathcal{L}_{\theta}(r) & x_{\mathcal{L}_N(r)} = \emptyset \end{cases}$$

This modified traversal function subsumes the previously introduced traversal function and can generally replace the previous definition. In order to deal with missing values, the evaluation of split candidates has to be modified. It makes sense that, during the evaluation of each candidate, the likelihood ratio of only the non-missing values is considered for the split. This avoids comparison of a larger number of observations in the pre-split model against a decreased number of observations in the post-split model. This necessitates a modification of the likelihood ratio test statistic for split candidate evaluation. Previously, we were able to calculate the negative two log-likelihood of the pre-split model once for each evaluation of a set of split candidates. With missing values, the situation changes slightly. The pre-split model has to be evaluated once for every split candidate, since the number of observations changes depending on the pattern of missing values in the covariate:

Theorem 69. *Let Υ be a SEM Tree with template model M that is described by a expectations vector μ and a covariance matrix Σ. Let $n \in N$ be a node with data set $(o', c') = D' \subseteq D$. Let D_{left} be the left data set of the split candidate and D_{right} the right data set. Let $\hat{\theta}$ be found by minimizing $-2LL\left(D_{left} \cup D_{right} | M, \hat{\theta}\right)$, $\hat{\theta}_{left}$ be found by minimizing $-2LL\left(D_{left} | M, \hat{\theta}_{left}\right)$, and $\hat{\theta}_{right}$ be found by minimizing $-2LL\left(D_{right} | M, \hat{\theta}_{right}\right)$. Under the null hypothesis that a split candidate is uninformative, the likelihood ratio Λ between a pre-split and a post-split model under covariates with missing values is*

[2]Note the distinction between the missing value symbol Ø and the symbol for the empty set $\emptyset$.

$$\Lambda = -2LL\left(D_{left} \cup D_{right}|M, \hat{\theta}\right) + 2LL\left(D_{left}|M, \hat{\theta}_{left}\right) + 2LL\left(D_{right}|M, \hat{\theta}_{right}\right)$$

Proof. The proof follows from the definition of cross-validation and Theorems 51–54 concerning the pre-split and post-split models. □

Note that the same applies to the CV criterion if applied for model selection. For the sake of brevity this criterion is not re-iterated here.

Alternatively, a surrogate approach can be employed to deal with missing values (cf. Hastie, Tibshirani, & Friedman, 2001). The approach is based on finding another covariate that most closely describes the same split. This can be thought of as finding another covariate that best predicts the outcome of the missing-value covariate. If the value for the surrogate is also missing, the process is continued. If the value for all covariates is missing, the variable can be attributed to the larger class.

3.6.3 Variable Imputation With SEM Trees

At the stage of data analysis, one might want to infer missing variables of the observations $o \in O$. For example, this could be a necessary pre-processing step for further analyses that require a data set without missing values. Variable imputation based on a SEM Tree improves classic model-based imputation as SEM Trees provide a model that partitions the data set into groups with significant differences in the data set. Assuming that the tree recovers the group structure of the population, or at least parts of it, a more accurate retrieval of missing variables is to be expected.

The following description follows the single imputation procedure as described by Rubin (1976). In a maximum likelihood setting, we may choose to draw missing values according to a multivariate distribution conditioned on the observed values. The sampling distribution for missing values is accordingly given by the conditional multivariate normal. For a given pattern of missingness, let a data point x be partitioned in

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and let the mean vector μ and the covariance matrix Σ be partitioned as follows

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

with x_1, μ_1, Σ_{11} corresponding to the current pattern of missingness and x_2, μ_2, Σ_{22} corresponding to the non-missing values. The missing values of the vector x_1 can then be inferred by the maximum likelihood solution of the conditional multivariate normal, which is the mean value of the conditional normal:

$$\hat{x_1} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\left(x_2 - \mu_2\right)$$

A caveat of the imputation of the maximum likely value is an artificial reduction of the variance in the imputed values which leads to a reduction of confidence intervals for parameter estimates and an overestimation of the precision of subsequent analyses. A variant of the above

method is the introduction of systematic variation by drawing new values from the conditional distribution with

$$\begin{aligned} \hat{\mu} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ \hat{\Sigma} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned}$$

$$\hat{x_1} \sim \mathcal{N}\left(\hat{\mu}, \hat{\Sigma}\right) \tag{3.6.1}$$

As a refinement to this general approach, model-based variable imputation offers the reconstruction of missing data sets according to hypotheses about the data. However, Schafer and Graham (2002) state that there is no need for a scientific theory underlying an imputation model. An example of variable imputation is given in Section 5.

3.7 The SEM Tree Package

SEM Trees were implemented in the R Project (Ihaka & Gentleman, 1996) as the "semtree" package, which is available at `http://www.brandmaier.de/semtree` (Brandmaier, 2011). It is based on OpenMx (Boker et al., 2011), which can be freely downloaded at `http://openmx.psyc.virginia.edu` (OpenMx Project, 2011). The package allows the construction of SEM Trees based on SEMs defined in OpenMx either as path models or as matrix representation. Furthermore, it supports the rendering of trees directly to clean and publication-ready LaTeX code. For non-LaTeX users the output can be rendered to a postscript file or to any other supported graphic file format based on the LaTeX source code. The suggested Bonferroni-corrections, CV, and pruning are available for the construction of trees. By this integration into the R framework and OpenMx, SEM Trees seamlessly integrate in the analysis workflow of researchers.

3.8 Summary

SEM Trees combine the benefits of exploratory and confirmatory approaches and provide a powerful tool for the theory-guided exploration of empirical data. The approach offers exploratory analyses for the refinement of a large range of hypothesized models. I introduced the methodology and discussed four approaches to select split candidates in the tree generation process which are based on the likelihood ratio of pre-split and post-split models. Two approaches provide multiple-comparison corrections for the first criterion based on selection of a maximum of individual log-likelihood ratios of pre-split and post-split models. The fourth criterion is based on a cross-validation procedure to select split candidates. We have discovered that in this way SEM Trees choose covariates with maximal information gain with respect to the model-predicted distribution. When the CV criterion is employed to build SEM Trees, covariates that maximize the mutual information are chosen. The suggested criteria also constitute stopping rules that determine when tree growing is terminated. Two criteria are based on Bonferroni-type corrections that assume independence of covariates. There are a variety of further approaches that aim at correcting the family-wise error rate of multiple tests; for example, the Bonferroni-Holm

correction (Holm, 1979) is another well-known method that could be applied to trees. Some go as far as to argue that a Bonferroni correction "creates more problems than it solves" (p.1236 Perneger, 1998); for example, it increases the type-II error rate, and the strength of the finding depends on the number of total tests. These problems are addressed by using the suggested cross-validation procedure. A promising approach was presented by Leek and Storey (2008), who estimate a dependence kernel from the data that can be used to render multiple correlated tests independent in their framework. A synthesis of this framework with SEM Trees could provide an additional stopping criterion that is faster to calculate than cross-validation but may promise a less conservative variable choice than provided by the Bonferroni correction.

If groups are truly different in the population, respective covariate splits are invariant to monotonous transformations of the original covariates since monotonous transformations do not change the order of the observed variables on which the binarization of the covariate splits is based. This follows immediately from the definition of the induced covariates (cf. Subsection 3.4.2) since the process is essentially based on sorting the attributes and the same conclusions pertinent to the invariance of PDC apply (cf. Section 2.5). For example, in lifespan research, researcher can be interested in age as a predictor and test interaction terms like age, age^2 and/or age^3 in their linear models, for example, to examine whether a particular measured variable depends polynomially on age (e.g. McArdle & Prescott, 1992; Kotter-Grühn, Wiest, Zurek, & Scheibe, 2009; Wrzus et al., 2011). A typical construct could be a cognitive ability that is less pronounced in young and older age, but peaks during young adulthood. In that case, a polynomial of degree 2 could be used to model this influence of age on the outcome. Also, age is often centered or expressed as standardized age, often termed $z-age$, as a predictor. Testing such polynomial transformations and normalizations of split candidates is implicit in the search procedure of SEM Trees, which can potentially reduce the number of covariates that need to be included as candidates.

In this thesis, parameter estimation is performed using a maximum-likelihood estimation procedure. However, the suggested approach can be extended to other estimation methods. Any other discrepancy function can also be used for parameter estimation, such as *Generalized Least Squares* (GLS) or the *Asymptotic Distribution Free* method (Browne, 1984). As long as this function provides an asymptotic χ^2-fit statistic, the split selection can be based on the likelihood ratio test. If the distribution is different or unknown, Cross-Validation estimates can still be employed. An important advantage of the ML approach is the information theoretic duality of the split candidate selection criteria maximizing information gain and mutual information.

SEM Trees inherit the statistical properties that the template SEM and the chosen estimator impose; for example, when using a maximum likelihood estimator, multivariate normality is an essential requirement to the data set. Violations of the normality assumption, e.g., excessive skewness or kurtosis, may affect model fitness and significance values. SEM Trees tighten this restriction by assuming multivariate normality for the conditional distributions of the model variables given the induced covariates. In future work, I plan to consider different estimators, which are, in some situations, more robust to violations of the normality assumption, like GLS or *Weighted Least Squares* (cf. Browne, 1974). SEM Trees provide a good basis for such extensions as the choice of the objective function for the parameter estimation can be replaced without changes to the construction method of the tree.

A second caveat for multi-group models is that a sufficient number of participants have to be assigned to each group to assure asymptotic behavior. Multiple suggestions for an appropriate

sample size have been made, ranging from minimum sample sizes of 10 to 200 (cf. Tanaka, 1987). If groups are formed by only a few individuals, standard errors of parameter estimates have to be interpreted with care. A possible way to decrease the influence of low sample size is to apply Bayesian estimation (cf. Lee, 2007), which I regard as an interesting extension to SEM Trees.

Variable and cut-point selection strategies in decision trees are often subject to bias (Kim & Loh, 2001; Dobra & Gehrke, 2001). Typically, a larger number of categories of a variable leads to a higher probability that this variable is selected under the null hypothesis. (Loh, 2002) suggested a bias correction of the χ^2-statistic based on bootstrap-estimates. Zeileis et al. (2008) and Hothorn, Hornik, and Zeileis (2006) present an elaborate framework based on permutation statistics that allow unbiased selection of variables in decision trees. Strobl, Wickelmaier, and Zeileis (2011) employ a two-stage process to correct for selection bias. In the first stage, the permutation framework is used to determine an unbiased split candidate. In a second stage, a cut point for binary trees is selected by choosing the binary split with the maximal likelihood after the split. SEM Trees correct for multiple testing by employing CV. However, this procedure is potentially biased to select variables with a larger number of splits under the null hypothesis. Depending on the degree of the violation of additional independence assumptions, the nested Bonferroni-correction tends to a bias for variables with less categories. Employing an independence kernel, as suggested by (Leek & Storey, 2008) can provide a correction that removes the bias of the Bonferroni correction. Alternatively, a nested CV approach can be employed that selects cut points of variables in an inner loop and variables in an outer loop. Comparing these approaches with respect to bias, stability, sensitivity, and computational demands remains for future work.

Another promising future line of research concerns integration of time-varying covariates. SEM Trees are well suited for this extension as the integration of time-varying covariates is common in SEMs, but research needs to be done to find optimal ways of integrating these covariates into the split process.

As mentioned above, an implementation of SEM Trees with a range of options summarized below is available for the R language: `http://www.brandmaier.de/semtree`

3.8.1 Relation to Other Methods

SEM Trees are in line with recent research that increasingly use advances in computing power. Ever-growing large-scale data sets, prominently from neuroimaging, have led to exploratory search techniques in the construction of SEMs. For example, Gates and colleagues (Gates, Molenaar, Hillary, & Slobounov, 2010; Gates, Molenaar, Hillary, Ram, & Rovine, 2010) proposed a large-scale model selection procedure of SEMs for the connectivity among brain regions, and Kenny et al. (2009) presented a parallel architecture that explores a gigantic model space for finding connectivity models in fMRI data.

In many situations, it is reasonable to state that, instead of a single model for all participants, a multi-group model is more appropriate to represent the structure of the data. In clinical trials, subgroup analysis is an often performed technique to test whether a treatment affects sample subgroups differently. In this perspective, SEM Trees can be regarded as a structured approach to recursive subgroup partitioning with respect to a set of covariates. Multi-group modeling in SEM has been around since its introduction for factor analysis by Jöreskog (1971). SEM Trees

can be conceived of as a recursive multi-group approach that exhaustively searches all available covariate-specific splits of a data set and constructs a hierarchically partitioned multi-group SEM with a greedy approach.

In general, SEM Trees are similar to unsupervised clustering techniques in addressing the question whether latent variables and their relations among each other and to observed variables form covariate-specific clusters.

SEM Trees also relate to longitudinal recursive partitioning as introduced by Segal (1992). He extended the decision tree paradigm to multivariate outcomes, which allows the modeling of longitudinal data. As he pointed out, visual analysis is often performed to detect substantial differences in longitudinal analysis. The identification of different underlying groups may be possible for small sample sizes. However, this approach lacks a formal definition of difference and, as relying only on the researcher's experience, is error prone. Even worse, with large sample sizes the identification of subgroups with structural differences quickly becomes unfeasible. Therefore, a robust and systematic way of performing the analysis of substructures is of substantial worth. Segal's (1992) approach minimizes the distance between the observed covariance matrix and a hypothesized covariance matrix. SEM Trees are in the same line of research since essentially they estimate differences of model-implied and data-implied covariance matrices. However, SEM Trees offer a larger variety of underlying models by harnessing the full potential of SEM.

The previous approaches were based on the detection of a data set's heterogeneity that can be observed. Mixture models address a similar problem like SEM Trees with the difference that they ask how unobserved heterogeneity in the data can be discovered and dealt with. Assuming that the collected observations stem from multiple groups, a multi-group model can be formulated that distributes each observation to a unique group of a fixed number of g hypothesized groups. Each group is distributed according to

$$-2LL\left(x|\theta\right) = -2\sum_{i=1}^{g} \pi_i \cdot LL\left(x|\theta\right)$$

subject to a choice of π_i satisfying $\sum_i^g \pi_i = 1$. The likelihood of each model in the mixture is calculated in the standard way. The number of parameters of the resulting mixture model is the number of free parameters in the template model times the number of groups plus the number of mixing parameters. Lee and Song (2003) give a derivation of how to estimate this type of model. Mixture models are an elegant method to account for unobserved heterogeneity. However, there are important distinctions to SEM Trees. SEM Trees partition the data set with respect to covariates that explain maximal parameter differences in a SEM, i.e., they produce a partition with respect to observed heterogeneity. In this respect, they maximize heterogeneity with respect to model parameters between groups and maximize homogeneity within groups. Most importantly, the splitting in subgroups is performed in a recursive fashion, i.e., the number of resulting groups is determined without requiring a method to select that number. Furthermore, a clear partition of the observed data is achieved with SEM Trees. Easily interpretable rules are generated that distribute already observed, but also newly observed data into the hierarchical group structure and the respective model parameters. Last, the group structure is directly interpretable, since covariates bear a meaning or description about the observations. Mixture models are a practical method to account for heterogeneity, but SEM Trees additionally find decision rules with respect to covariates that describe the heterogeneity.

Parallel to the development of SEM Trees, the pathmox package (Sanchez & Aluja, 2010) for R was released. Pathmox proposes recursive partitioning for path-analytic models. However, to my knowledge, there exist no publications about this approach. The estimation approach of SEM Trees has an advantage because it finds covariates that maximize mutual information. Furthermore, SEM Trees are equipped with tools that allow for the handling of missing values. Lastly, SEM Trees subsume extensions to specific models; most importantly for the behavioral sciences, SEM Trees subsume trees for factor models. The following chapter introduces extensions to the elementary SEM Tree paradigm.

4
Extensions of SEM Trees

In the previous chapter, fundamental aspects of SEM Trees were discussed. In this chapter, some important extensions to the concept of SEM Trees are presented. Three choices of template models can profit from a special treatment. An important topic when using factor models is the question whether the construct that is measured in both groups is in fact the same. This concept, usually referred to as *measurement invariance*, is integrated in Factor Model Trees. With a similar template model, Principal Component Trees that describe differences in principal subspaces of data can be built. Finally, a Gaussian approximation to the permutation distribution enables Permutation Distribution Trees that combine the central ideas of Chapter 2 and Chapter 3.

4.1 Factor Model Trees with Measurement Invariance

A widely used class of SEMs constitute factor models that define relations between hypothesized latent factors and observed scores, partitioning the observed variance into common variance, the common factors, and unique variance, the measurement error. In psychological research, hypotheses about factor-analytic structures are often tested across multiple groups. For example, in aging research, a common approach is the separation of the participants into models according to their age group, e.g., a model for the younger and a model for the older participants. Essentially, multi-group factor models are replications of a template factor model for each group, in which free parameters are unique within each group. In order to test hypotheses, parameter constraints can be set within groups but also across groups. This allows testing whether parts of the model are indeed equal across groups or significantly differ from each other. When using factor models, an obvious question arising is: "Do two groups differ in their average value of the latent construct?". For example, if the latent construct is intelligence, the research question could ask, whether there are differences in the average intelligence between groups. Questions like this can only be answered validly if the researcher can ascertain that indeed the conceptually identical latent constructs are measured across all groups. This

requirement is referred to as measurement invariance (MI) or *measurement equivalence* and has been debated in psychology for more than a century (Horn & McArdle, 1992; Meredith, 1964, 1993). Measurement invariance is traditionally tested through a sequence of hypothesis tests. In the literature, different concepts of invariance have been agreed upon. Typically, four nested concepts are described: (1) *Configural invariance*, sometimes termed *configuration invariance* or *pattern invariance,* requires the invariance of the pattern of zero and non-zero factor loadings across groups. Loosely, the structural connections of the variables in the models have to match. (2) *Metric invariance*, *weak invariance,* or *factor pattern invariance* requires the invariance of the values of factor loadings across groups, that is, all regression weights from latent variables to observed variables have to be equal across groups. (3) *Strong factorial invariance* assumes intercepts of the factors and all factor loadings to be the same across groups. (4) *Strict invariance* additionally restricts the residual error variances to be equal across groups, to allow the interpretation of standardized coefficients across groups.

Theorem 51 showed that the pre-split model can be seen as a combined model consisting of two models under the constraint that the parameters in both models are equal: One model for the left data set and one model for the right data set. As we have seen in Theorem 53, the pre-split model is nested in the post-split model. We extend this notion and will find that the post-split model with invariance constraints is a model that is nested in the post-split model and the pre-split model is nested in the invariance model. The nesting structure of these models is illustrated in Figure 4.1.1. The following definition formally describes the relation of the models by restriction on their respective parameter vectors:

Definition 70. Let M_{pre} represent a model with a parameterized expectations vector and covariance matrix. For an arbitrarily chosen split candidate, let M_{post} be the two-group model representing the left and right sub-model that originate from partitioning M_{pre} according to the split candidate. Furthermore, let M_{inv} be a restricted version of M_{post} that demands a chosen measurement invariance. Let a parameter vector $\hat{\theta}$ be partitioned into parameters in the factor loadings $\hat{\theta}_l$, parameters in the expectation structure $\hat{\theta}_m$, parameters in the residual terms $\hat{\theta}_\epsilon$, and other parameters $\hat{\theta}_x$, such that $\hat{\theta} = \left\{\hat{\theta}_l, \hat{\theta}_m, \hat{\theta}_\epsilon, \hat{\theta}_x\right\}$. As before, all three models are compound models of a left and a right model, and are therefore associated with a left parameter set $\hat{\theta}_{left}$ and a right parameter set $\hat{\theta}_{right}$ that can both be partitioned into the four parameter vector components. The model with measurement invariance M_{inv} can be obtained from M_{post} as follows:

(1) Configural invariance: M_{inv} and M_{post} are the same.

(2) Metric invariance: The invariance model is obtained by adding the equality constraint $\hat{\theta}_{left,l} = \hat{\theta}_{right,l}$ to the pre-split model.

(3) Strong invariance: The invariance model is obtained by adding the equality constraints $\hat{\theta}_{left,l} = \hat{\theta}_{right,l}$ and $\hat{\theta}_{left,m} = \hat{\theta}_{right,m}$

(4) Strict invariance: The invariance model is obtained by adding the equality constraints $\hat{\theta}_{left,l} = \hat{\theta}_{right,l}$, $\hat{\theta}_{left,m} = \hat{\theta}_{right,m}$, and $\hat{\theta}_{left,\epsilon} = \hat{\theta}_{right,\epsilon}$

In all cases, the pre-split model M_{pre} is obtained by requiring $\hat{\theta}_{left,l} = \hat{\theta}_{right,l}$, $\hat{\theta}_{left,m} = \hat{\theta}_{right,m}$, $\hat{\theta}_{left,\epsilon} = \hat{\theta}_{right,\epsilon}$, and $\hat{\theta}_{left,x} = \hat{\theta}_{right,x}$ to either the invariance model or the post-split model.

Theorem 71. *Let M_{pre}, M_{inv}, and M_{post} be as above. The nesting structure of the models induces the following relation under configural, metric, strong, or strict invariance:*

$$-2LL\left(x|M_{post}\right) < -2LL\left(x|M_{inv}\right) \leq -2LL\left(x|M_{pre}\right)$$

Proof. Let the partition of $\hat{\theta}$ in left and right model, and in parameters concerning the factor loadings, expectations, residuals, and other parameters be as in the previous definition. Let df_{pre}, df_{inv} and df_{post} be the degrees of freedom of M_{pre}, M_{inv}, and M_{post}. The nesting structure as above leads to the following differences in degrees of freedom and therefore the following nesting structure:

(1) Configural invariance: $df_{post} - df_{inv} = |\hat{\theta}_{left}|$, $df_{inv} - df_{pre} = 0$

$$-2LL\left(x|M_{post}\right) < -2LL\left(x|M_{inv}\right) = -2LL\left(x|M_{pre}\right)$$

(2) Metric invariance: $df_{post} - df_{inv} = |\hat{\theta}_{left,m}| + |\hat{\theta}_{left,\epsilon}| + |\hat{\theta}_{left,x}|$, $df_{inv} - df_{pre} = |\hat{\theta}_{left,l}|$

$$-2LL\left(x|M_{post}\right) < -2LL\left(x|M_{inv}\right) < -2LL\left(x|M_{pre}\right)$$

(3) Strong invariance: $df_{post} - df_{inv} = |\hat{\theta}_{left,\epsilon}| + |\hat{\theta}_{left,x}|$, $df_{inv} - df_{pre} = |\hat{\theta}_{left,l}| + |\hat{\theta}_{left,m}|$

$$-2LL\left(x|M_{post}\right) < -2LL\left(x|M_{inv}\right) < -2LL\left(x|M_{pre}\right)$$

(3) Strict invariance: $df_{post} - df_{inv} = |\hat{\theta}_{left,x}|$, $df_{inv} - df_{pre} = |\hat{\theta}_{left,l}| + |\hat{\theta}_{left,m}| + |\hat{\theta}_{left,\epsilon}|$

$$-2LL\left(x|M_{post}\right) < -2LL\left(x|M_{inv}\right) < -2LL\left(x|M_{pre}\right)$$

□

The following variant of the split selection procedure is explained for cases when using the likelihood ratio criterion but it applies analogously to using the CV criterion. So far, a split was considered if the likelihood ratio was significant or the expected likelihood ratio larger than zero. For example, when using the maximum-χ^2 criterion for building a SEM Tree, a variable split is only maintained if the likelihood ratio $M_{presplit} - M_{postsplit}$ is significantly different from a central χ^2-distribution, suggesting that $M_{presplit}$ is an inferior representation of the data set. Under measurement invariance, a further condition that has to be fulfilled by a split candidate to be a valid choice is added. A split candidate is only valid if $M_{restricted} - M_{postsplit}$ is not significant, i.e., at least there is no significant hint for a violation of measurement invariance. If this difference is significant, the restrictions imposed by the chosen measurement invariance significantly worsened the likelihood of observing the data under the model. The split candidate has to be rejected since MI cannot be guaranteed. In order to evaluate MI in the candidate selection process, the evaluation procedure has to be modified only marginally. After having found the best split candidate, by either the maximum log-likelihood ratio or the CV criterion, the additional invariance condition is checked. If the model passes, a split is performed. If it fails the invariance test, the procedure is repeated for the second-best split candidate that still fulfills the original criterion, and so on. If there are no split candidates that fulfill measurement invariance, splitting is stopped at that leaf node.

This procedure allows the generation of a SEM Tree under a chosen level of MI. Researchers choosing to use MI in their models may continue using SEM Trees with MI. This will be most relevant for those using factor models. Note that there are situations in which building a SEM Tree without factor invariance can be reasonable, even if the template model is a factor model.

In that case, a SEM Tree could be interpreted as a structure that describes different factor profiles within the sample, that is, different subgroups that differ in the configuration of their latent structure. Latent scores across groups can then only be compared with caution. An application of a SEM Tree with a factor model as template model is presented in Chapter 5.

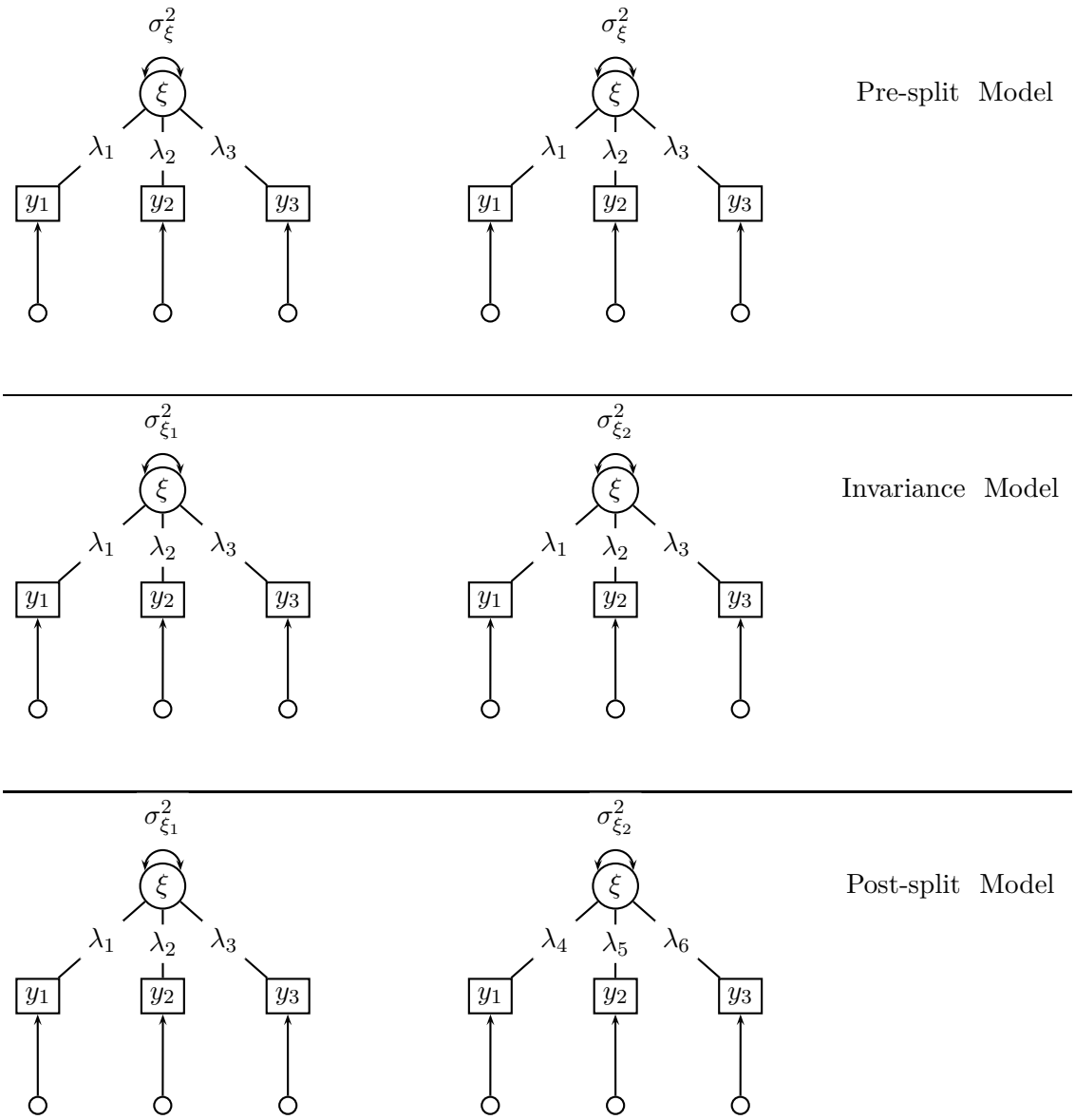


Figure 4.1.1: Schematic nesting of models during the evaluation of a split candidate with measurement invariance. Each model is nested in all models below itself. The upper model depicts a pseudo multi-group model before a split. All parameters are equal across groups since no split was actually performed. The lower model is the two-group model after the split, estimating all free variables uniquely in each group. The middle model is the invariant model that introduces further restrictions to the lower model depending on the invariance assumptions. In this example, equality of factor loadings is illustrated.

4.2 Principal Component Trees

Principal Component Trees (PC Trees) are a special case of SEM Trees using a Principal Component Analysis (PCA) as a template SEM. In the following section, it will become clear that, indeed, only a few modifications are needed to build tree structures with models that describe a PCA. The resulting hierarchical structure describes a partition of the data set with maximal differences with respect to the principal subspaces of the subsets. PCA is briefly reviewed before the extension from SEM Trees to PC Trees is shown.

4.2.1 Principal Component Analysis

PCA is a method that transforms a set of possibly correlated observed variables into a set of uncorrelated variables. The new set of variables is a linear combination of the original variables. The applied mapping is called an orthogonal transformation. The resulting transformed variables are called the *principal components* and are, by convention, sorted in a way that the first variable accounts for the highest variance in the original observations and the last component for the least variance. Therefore, PCA can be used as a method of dimension reduction by removing principal components from a data set that account for small explained variance. The projection onto principal components is found by an eigenvalue decomposition of the covariance matrix of the data set or a singular-value decomposition of the data matrix (cf. Bishop, 2006). In addition, a PCA can also be formulated as SEM. PCA is closely related to the latent factor model because both methods discover sources of common variance across the observed variables. However, in a PCA without reduction of components, the observed variables are assumed to be measured without error and thus, there are no residual error terms in the model. By definition, in PCA, there is no covariance between latent factors, whereas in latent factor analysis, the covariance between latent factors is often of interest.

The eigenvalue decomposition of a matrix X is given by solving

$$X = UVU'$$

where U is a matrix of eigenvectors and V a diagonal matrix of eigenvalues.

PCA is a linear transformation $y = Ax$ with $x \sim N(0, B)$ and thus an implied covariance matrix of $\Sigma = ABA'$ which is equivalent to the formulation of the eigenvalue decomposition above. In PCA representations with a reduced number of principal components, there will be a residual error which denotes the error of representing the original observations by only the reduced number of principal components.

4.2.2 PCA and Factor Analysis

PCA and factor analysis (FA) with factor models show much resemblance. Both methods hypothesize that the observations are linear combinations of a set of latent variables and, based on this assumption, represent reduced rank representations of a data set. The essential conceptual difference is that in FA, the covariance matrix of the residuals is diagonal and of full rank, i.e., the measurement errors are assumed to be independent and unique to each observed variable, whereas in PCA, the covariance matrix of the residuals is not of full rank and not diagonal, i.e., the measurement error structure is correlated (Velicer & Jackson, 1990). Velicer and Jackson

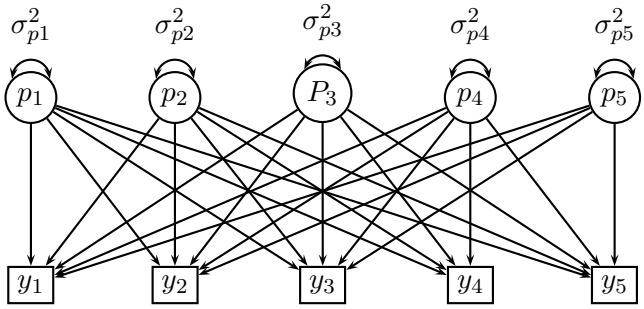


Figure 4.2.1: A SEM representation of a Principal Components Analysis (PCA). Without additional constraints, this model is under-identified, that is, it contains more free parameters than degrees of freedom in the empirical data.

(1990) claims that results of PCA and FA are often very similar This claim is contrasted by simulation results of Widaman (1993), who showed that PCA parameter estimates differ consistently from the true parameters if a factor model holds perfectly. This author concludes further that the choice between PCA and FA depends on the type of research question and no model is as such superior to the other. Whenever researchers are interested in a simple rank reduction of their data sets, PCA is favorable because it does not tempt the researchers to impose any interpretations to the data, particularly, regarding the observed variables as manifestations of latent entities.

4.2.3 PC Trees are SEM Trees

The SEM representation in Figure 4.2.1 of the PCA has p^2 free parameters in the structural matrix, representing the eigenvectors and another p free parameters for the latent variances, representing the corresponding eigenvalues. This amounts to $p^2 + p$ free parameters that face only $(p^2 + p)\,\frac{1}{2}$ degrees of freedom of the sample covariance matrix, that is, the model is under-identified. PCA requires a orthonormal transformation matrix. This imposes another p constraints that the columns of the structural matrix sum up to one, i.e., the basis vectors of the principal subspace have unit length. Furthermore, the basis vectors have to be orthogonal. This imposes linear independence constraints on all pairs of vectors, resulting in additional $(p^2 + p)\,\frac{1}{2}$ constraints. This suffices to identify the model.

It is not trivial to include the above constraints on the parameters into a regular SEM software. Dolan, Bechger, and Molenaar (1999) suggest conceiving a PCA in a SEM setting as a multi-group model in which one group represents the under-identified structural model of the PCA as illustrated in Figure 4.2.1, and a second dummy model, which is fitted at the same time as a multi-group model, contains the orthonormality constraints.

In OpenMx there is full freedom in the modification of the fitting function to incorporate these constraints. Therefore, I suggest using a modified maximum likelihood fit function to

estimate PCA in SEM, which can be regarded as a penalized maximum-likelihood function

$$F_{MLP} = F_{ML} + \lambda \cdot tr\left[\left(\Lambda\Lambda^T - I\right)^T \left(\Lambda\Lambda^T - I\right)\right]$$

whereby F_{ML} is the regular maximum likelihood fit function, I is the identity matrix and λ is a parameter that determines the influence of the penalty on the fit function. The penalty has the mere function of assuring the matrix Λ to be orthonormal. The transpose of an orthonormal matrix is its inverse. Therefore, the PCA solution satisfies

$$\Lambda\Lambda^T = I$$

and consequently, $\Lambda\Lambda^T - I$ determines the element-wise deviation of Λ from fulfilling this constraint. Thus, $\left(\Lambda\Lambda^T - I\right)^T \left(\Lambda\Lambda^T - I\right)$ is the matrix of sums of squares and cross-products and the trace of this product determines the deviations' sums of squares. Hence, the penalty represents the least squares error of the orthonormality constraint. Starting values for Λ and for the diagonal entries of the latent variances can be randomly chosen.

It seems reasonable to add another post-processing step to PC Trees. The principal components are usually expected to be presented in an order according to their influence, e.g., sorted with respect to their eigenvalues, from largest to smallest. Accordingly, the latent factors in a PC Tree are sorted for each model so that the latent variance estimates fulfill $\sigma_i^2 \geqslant \sigma_j^2$ for $i < j$, with σ_k^2 being the variance estimate for the k-th latent factor.

In principle, it is possible that the optimizer does not converge on the minimum of the penalized likelihood function that features a sufficiently small value of the penalty. Therefore, in practice, it is necessary to check whether the penalty of the solution that the optimizer returned is small enough, e.g., smaller than a fixed threshold, e.g., 1×10^{-10}. When the penalty is larger, it is assumed that not only numerical reasons led to the deviation from zero, and the optimization process is restarted with a new set of random starting values.

The presented method of re-expressing PCA as SEM allows us to construct PC Trees. These structures essentially predict differences in the principal subspaces based on partitions of a data set. The extension from SEM Trees to practical PC Trees is minimal and subsumes the modified likelihood function, the sorting of the parameter estimates and the numerical checking whether the penalty is close to zero. Within the SEM Tree library, a wrapper function to the generic SEM Tree function is available that constructs the PCA model and the modified likelihood function. This allows the straightforward use of PC Trees without any additional programming effort.

In principle, PC Trees answer two questions, namely, "Do hierarchical group differences exist in the rotation of the principal subspace?" and "Do the variance contributions of the principal axes differ across group hierarchies for a chosen rotation?". The first question addresses the existence of any significant differences in the model estimates across groups and can therefore discover differences of variance contributions of principal axes and differences of subspace rotations. The second question addresses differences in the captured variance for a chosen rotation and can therefore potentially find subgroups that differ in the amount of variance that is explained by the chosen rotation. The probabilistic formulation of the PCA has the added advantage that it can handle missing values appropriately.

A PCA with all components leaves no degrees of freedom in the fitting process, that is, the resulting tree will be equivalent to a SEM Tree with a freely estimated covariance matrix, which

is commonly referred to as a saturated model. Any imposed structure with the same number of free parameters as degrees of freedom is indiscernible from the freely estimated covariance matrix because the likelihood values will always be equal. As a consequence, the usage of PC Trees only makes sense if a dimensionality reduction is performed, that is, the number of latent components selected is smaller than the number of observed components.

4.3 Permutation Distribution Trees

So far, I have made a clear distinction between settings in which either SEM Trees or PDC are applied. PDC assumed neither a model of the data nor the availability of additional covariates, whereas SEM Trees exploit available covariates to refine formal models of the data. I have already alluded to the fact that SEM Trees resemble clustering but with the important distinction that only meaningful clusters are explored, that is, only clusters that we can interpret in terms of covariates. It could be reasonable, though, that in a setting in which a researcher plans to employ PDC, additional covariates are indeed available. In this situation, it would be beneficial to consolidate SEM Trees and the permutation distribution. In the following, I will investigate how this can be achieved.

4.3.1 Permutation Distribution Covariance Matrix

In the situation introduced above, where PDC is to be applied while exploiting knowledge from additional covariates, the permutation distribution (PD) is implicitly taken as a model of the data. In particular, we consider model-based recursive partitioning with the models representing the codebooks. Above, we have noted that the PD is a multinomial distribution (see Section 2.11). We have used likelihood ratio tests to decide whether two subsets of the data set should be represented by distinct distributions or whether a single distribution is sufficient to model the data. SEM Trees require models of normally-distributed observed variables. In order to consolidate both methods and build SEM Trees with PD models as template models, we need an approximation of the multinomial distributed codewords by multivariate normal distributions. Indeed, there is a known straightforward approximation.

Let $M(n, \theta)$ be a multinomial distribution, that is, a probability measure on $\mathbb{Z}_+^m$ with $\theta := \{\theta_1, \ldots, \theta_m\}$ representing the joint distribution of counts in m cells when distributing n balls independently among these, with θ_i representing the probability of choosing the i-th cell for a ball. For large n, the central limit theorem ensures that the count of each cell X_i is approximately normally distributed with the first two moments of the binomial distribution

$$\begin{aligned} \mathbb{E}(X_i) &= n \cdot \theta_i \\ \mathrm{Var}(X_i) &= n \cdot \theta_i (1 - \theta_i) = n\theta_i - n\theta_i^2 \end{aligned}$$

For m cells, we can construct a covariance matrix between all X_i with the following formula

$$Cov(X_i, X_j) = \begin{cases} n \cdot \theta_i (1 - \theta_i) & i = j \\ -n\theta_i\theta_j & otherwise \end{cases} \tag{4.3.1}$$

With this normal approximation, we have obtained a vector of expectations and a matrix of covariances representing a multinomial distribution. Using these as a model-implied covariance matrix and expectations vector, we can start exploring covariate structures that explain significant differences between permutation distributions, based on likelihood ratio tests of normal approximations of the of codebooks' multinomial distributions.

4.3.2 Path Representation of PD Trees

Assume that we have collected a set of time series. If we chose an embedding dimension of $m = 3$, the resulting covariance matrix is of the size $m! = 3! = 6$. Let a given PD be a multinomial distribution $M(n, \theta)$ with observed frequency vector $p := \{p_1, \ldots, p_6\}$. We can approximate P with a multivariate normal distribution with a multivariate normal distribution X with covariance matrix

$$Cov(X) = n \cdot \begin{bmatrix} p_1 - p_1^2 & -p_1p_2 & -p_1p_3 & -p_1p_4 & -p_1p_5 & -p_1p_6 \\ -p_1p_2 & p_2 - p_2^2 & -p_2p_3 & -p_2p_4 & -p_2p_5 & -p_2p_6 \\ -p_1p_3 & -p_2p_3 & p_3 - p_3^2 & -p_3p_4 & -p_3p_5 & -p_3p_6 \\ -p_1p_4 & -p_2p_4 & -p_3p_4 & p_4 - p_4^2 & -p_4p_5 & -p_4p_6 \\ -p_1p_5 & -p_2p_5 & -p_3p_5 & -p_4p_5 & p_5 - p_5^2 & -p_5p_6 \\ -p_1p_6 & -p_2p_6 & -p_3p_6 & -p_4p_6 & -p_5p_6 & p_6 - p_6^2 \end{bmatrix}$$

and the corresponding vector of expectations

$$\mathbb{E}(X) = n \cdot [p_1, p_2, p_3, p_4, p_5, p_6]$$

If a SEM software allows the specification in general matrix algebra, the implementation of a model like this is straightforward. But can this model also be shown in a path-analytic representation, i.e., is the model within the model class of path-analytically representable models? In the following, a constructive proof is sketched.

In order to build a path-analytic representation of the covariance matrix, we start with a simplified model. We construct path representations of the X_i without covariances, and therefore effectively build a set of independent binomial distributions, approximated by a model of normally distributed observations. For each variable X_i that measures the counts of the i-th cell, we define the observed variable X_i and a set of latent variables A_i, B_i, C_i, and V_i with the following equations in our model

$$\begin{aligned} V_i &= A_i + p_iB_i + p_iC_i \\ X_i &= \sqrt{n} \cdot V_i + n \cdot p_i \\ Var(A_i) &= p_i \\ Cov\,(B_i, C_i) &= -\frac{1}{2} \\ Var\,(B_i) &= Var\,(C_i) = 0 \end{aligned}$$

A graphical representation of this model is provided in Figure 4.3.1. By applying the rules of covariance algebra (see Appendix A), we find that

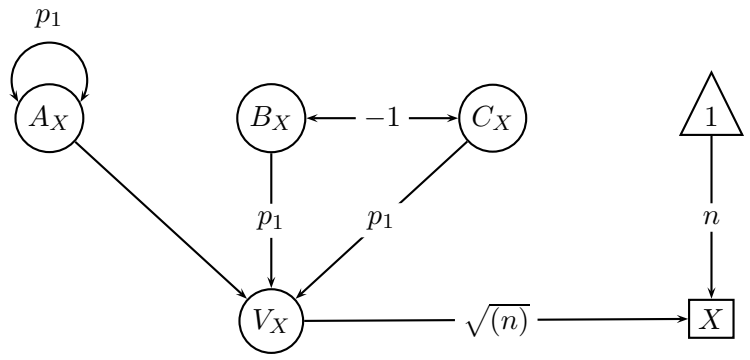


Figure 4.3.1: Path representation of the normal approximation of a binomial distribution. Even if this representation is unusual and can generally be replaced by a clearer matrix representation, it nevertheless shows that the model is within the class of path-analytically representable models.

$$\begin{aligned} Var\left(V_i\right) &= Cov(V_i, V_i) \\ &= Cov(A_i + p_i B_i + p_i C_i, A_i + p_i B_i + p_i C_i) \\ &= Var(A_i) + 2p_i^2 \cdot Cov(B_i, C_i) \\ &= p_i - p_i^2 \end{aligned}$$

and therefore

$$Var\left(X_i\right) = Var(\sqrt{n} \cdot V_i) = n \cdot \left(p_i - p_i^2\right) = n \cdot p_i \left(1 - p_i\right)$$

The expectation of X_i results to

$$\begin{aligned} E\left(V_i\right) &= E\left(A_i\right) + p_i E\left(B_i\right) + p_i E\left(C_i\right) \\ &= 0 \\ E\left(X_i\right) &= \sqrt{n} \cdot E\left(V_i\right) + n \cdot p_i = n \cdot p_i \end{aligned}$$

In order to represent a multinomial distribution, the covariances of the binomial distributions have to be accounted for. The covariances as given in Equation 4.3.1 are extended by additional covariance terms between all C_i and C_j

$$Cov\left(C_i, C_j\right)_{i \neq j} = -1$$

which yields the required covariance between the X_i

$$\begin{aligned} Cov\left(V_i, V_j\right) &= Cov(A_i + p_i B_i + p_i C_i, A_j + p_j B_j + p_j C_j) \\ &= p_i p_j Cov\left(C_i, C_j\right) \\ &= -p_i p_j \end{aligned}$$

$$\begin{aligned} Cov\left(X_i, X_j\right) &= Cov\left(\sqrt{n} \cdot V_i + n \cdot p_i, \sqrt{n} \cdot V_j + n \cdot p_j\right) \\ &= Cov\left(\sqrt{n} V_i, \sqrt{n}, V_j\right) \\ &= n \cdot Cov\left(V_i, V_j\right) \end{aligned}$$

The latent variables in this model are purely derived to match the required covariance matrix and have no further apparent interpretation. Indeed, the resulting model looks unusual but shows that the model belongs to the class of SEM that can be represented in a path notation.

4.3.3 Visualizing PD Trees

Because PD Trees are a form of SEM Trees, there is no need to change the algorithm or presentation of the trees in principle. However, for the most part, it will be useless to present the estimates for the codeword counts. The estimates constitute $m!$ free estimates and will take up much space in the graphical representation while conveying only little information. The essential information are the values of the split criterion and associated p-values, if available, that represent the magnitude of the differences between the permutation distributions. Therefore, I recommend hiding the estimates when depicting PD Trees.

4.4 Summary

In this chapter, I have introduced three extension to the paradigm of SEM Trees. For SEM Trees based on factor models, I extended the split criterion to account for measurement invariance. A related method to factor analysis is PCA. Using a SEM formulation of PCA enables PC Trees that recover covariate-specific partitions of a data set based on differences in the principle component structure. Using the permutation distribution as a model, PD Trees become possible that allow covariate-specific hierarchic structures of differences in the PDs of time series.

All three proposed extensions show the versatility and flexibility of SEM Trees. Any model that can be formulated in SEM can be used in SEM Trees. Factor analytic trees allow the refinement of factor-analytic models while testing the level of measurement invariance required by the investigator. This provides researchers with a new technique to detect heterogeneity in the sample, to find differences in the latent structure, and differences in the factor levels. Similarly, PC Trees allow to find differences with respect to the related technique of principal components of a sample, which is also regularly employed to detect commonality structures in a sample (Widaman, 1993). The third technique responds to the need researchers encounter after having found clustering structures with the PDC method. Given a clustering, the question may arise, what covariates describing different conditions or manipulations explain the observed

complexity differences of the time series. PDC Trees provide a solution to this problem by using a SEM formulation of the PD as template model for SEM Trees. Thereby, they allow finding covariate-specific partitions of a sample with respect to largest differences in the PD.

5 Applications

The following chapter contains demonstrations of both proposed algorithms, PDC and SEM Trees, on simulated data various real data sets. I selected data sets from the literature that were previously investigated by other researchers using different methods, so that the results of the novel methods can be discussed in the light of previous findings. Additionally, I present two data sets to analyze the performance of clustering that have not yet been reported in the literature. The first data set is a 64-channel electroencephalographic (EEG) recording of five participants in two conditions, with their eyes open and their eyes closed. Here, I present results of PDC in comparison to results of re-implementations of other important clustering approaches. The second new data set consists of a tri-axial accelerometer recording of two persons walking on a treadmill at different speeds. I will demonstrate the utility of PDC for segmenting this time series into homogeneous segments corresponding to the different walking speeds.

5.1 Applications of SEM Trees

First, demonstrations of SEM Tree analysis are presented. The first example demonstrates differences in the choice of a stopping criterion for growing a SEM Tree by comparing the reconstruction error of a parameter in an auto-regressive model. Then, I present three different data sets that have been analyzed previously by different methods, and I compare the results of SEM Trees with the earlier findings. Finally, I will re-use one of these data sets and simulate missing data values. A SEM Tree is then used to impute the observed variables.

5.1.1 Simulation

The following simulation compares different tree growing methods in their accuracy of parameter reconstruction. The generating model was an auto-regressive process of order 1 with a single parameter β, the autoregressive coefficient. I generated a data set by choosing β depending on the interaction of two covariates, $X \in \{0, 1\}$ and $Y \in \{0, 1, 2, 3, 4\}$. The distinct population

		RMSE	
Method	Parameter	Condition I	Condition II
Bonferroni	p=0.05	0.053	0.027
	p=0.001	0.042	0.022
Cross-Validation	five folds	0.038	0.024
	ten folds	0.038	0.023
Template Model		0.051	0.090
Pruned Model		0.049	0.040

Table 5.1: Results of different SEM Tree growing methods on a simulated data set. Data were generated by an autoregressive process. The table shows the corresponding root mean squared reconstruction error of the autoregressive coefficient.

groups were determined as *group 1* for $X = 0$ and $Y \in \{0, 1, 2\}$, *group 2* for $X = 1$ and $Y \in \{0, 1\}$, *group 3* for $X = 0$ and $Y \in \{3, 4\}$, and *group 4* for $X = 1$ and $Y \in \{2, 3, 4\}$. The parameter β was chosen based on the group membership of each individual. The simulation was performed under two conditions. In the first condition, the β_i-values for the respective group i were set to $\beta_1 = 0.1$, $\beta_2 = 0.15$, $\beta_3 = 0.2$, and $\beta_4 = 0.25$. In the second condition, the values were chosen as $\beta_1 = 0.1$, $\beta_2 = 0.2$,$\beta_3 = 0.3$, and $\beta_4 = 0.4$. A third covariate Z was generated from an uninformative binomial distribution. In the simulated data set, 500 participants were assumed to be observed of five time points, equally spaced across the study time span. The simulation was repeated 100 times. SEM Trees were constructed with five different methods: Two with nested Bonferroni-corrected p-values compared against α levels 0.05 and 0.001, two with five-fold and ten-fold cross-validation, and a tree that was trained on 80% of the data to a height of 5 and pruned back with 20% of the data. The accuracy of each tree was determined by the root mean squared error (RMSE) of the reconstruction of β. The RMSEs of the single trees were again averaged across all repetitions of the simulation. The simulation results are described in detail in Table 5.1. CV evinced the lowest reconstruction error in the first condition. In the second condition, Bonferroni and CV showed comparable results.

5.1.2 Univariate Regression in SEM Trees

Zeileis et al. (2006) presented an example of model-based recursive partitioning with univariate regression models. In the following, I will examine whether SEM Trees are able replicate their findings. Zeileis et al. (2006) used data collected by Bergstrom (2001), which is freely available in the AER package (Kleiber & Zeileis, 2008). The data set was collected from 1999 to 2000, and comprises bibliometric facts about 180 economic journals, including the numbers of citations, pages, and library subscriptions. Zeileis et al. (2006) fit a tree with a regression model to the data set. They report a template model that is driven by economic knowledge and regresses the price per citation of a journal on the number of library subscriptions, both in log scale. In this model, the slope of the regression line describes the price elasticity of the journal, a measure that determines the extent to which demand decreases by raising the price of the product. As potentially interesting covariates the authors included the age of the journal, the number of citations, the number of characters, the subscription price, and the journal's association with a

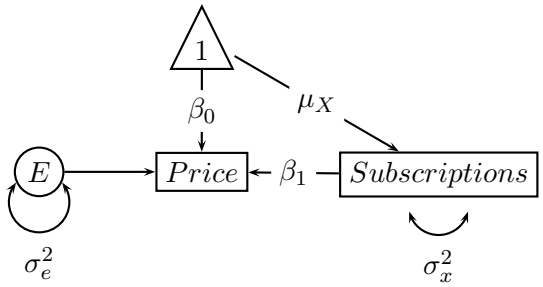


Figure 5.1.1: Univariate regression model of the number of subscriptions of a journal on the price per citation. The regression intercept is estimated as β_0, the regression of subscriptions on price as β_1. The distribution of the predictor subscriptions is estimated to be $\mathcal{N}(\mu_X, \sigma_X^2)$. The variance of the residual error is σ_e^2.

society. There were no missing values in any of the covariates.

I set up a univariate regression model as a SEM with five estimated parameters, so that we obtain the same model as Zeileis et al. (2006) used for their regression tree. The model includes two variables that represent the price per citation and the number of subscriptions of a journal, both logarithmically scaled. After this rescaling, the influence of the subscriptions on the price is assumed to be linear. The resulting regression line is estimated by the parameters β_0 representing the regression intercept and β_1 representing the slope. This slope is a formalization of the price elasticity. The residual variance is designated with parameter σ_e^2. Mean and variance of the regressor, the number of subscriptions, is estimated as μ_x and σ_x^2. A graphical representation of the model is given in Figure 5.1.1. Fitting this model to the complete data set yields a price elasticity of $\beta_1 = -0.533$.

First, I applied the SEM Tree algorithm with the default settings, using a ten-fold CV scheme and a minimal number of elements $min_N = 20$. The resulting SEM Tree is shown in Figure 5.1.2. The tree is equal to the regression tree by Zeileis et al. (2006), suggesting a division of the data set into two partitions, one representing all journals older than 19 years[1] and another one representing the more recent journals. The tree suggests a price elasticity of -0.605 for the older journals and a price elasticity of -0.403 for the younger journals. This suggests that demand for younger journals is less responsive to the journal's price per citation. For comparison, a second run of the algorithm was performed using the Bonferroni-correction with a significance level of $\alpha = 0.01$. The result did not change. For illustration, the resulting partitions of the data with their corresponding regression lines are plotted in Figure 5.1.3.

The SEM Tree algorithm was evaluated in comparison to Zeileis et al. (2006) who used the *model-based partitioning* algorithm (MOB) (Zeileis et al., 2008) from the party package (Hothorn, Hornik, Strobl, Zeileis, & Hothorn, 2011) to create a regression tree. I ran the SEM Tree algorithm with the default setting on 250 bootstrap samples of the journal pricing data set. In each replication, a SEM Tree was constructed from the bootstrap samples. The RMSE of

[1]Note that journal age is relative to the year 2000, when the data were collected.

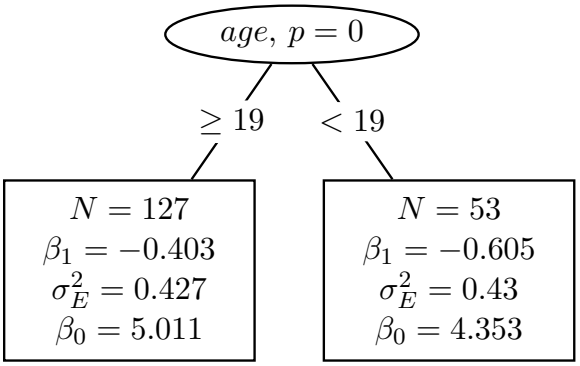


Figure 5.1.2: SEM Tree of journal pricing data set (Bergstrom, 2001) with Bonferroni correction and $\alpha = 0.001$. The same tree is obtained when using cross-validation. The estimated parameters are regression intercept β_0, slope β_1, and the residual error σ_E^2.

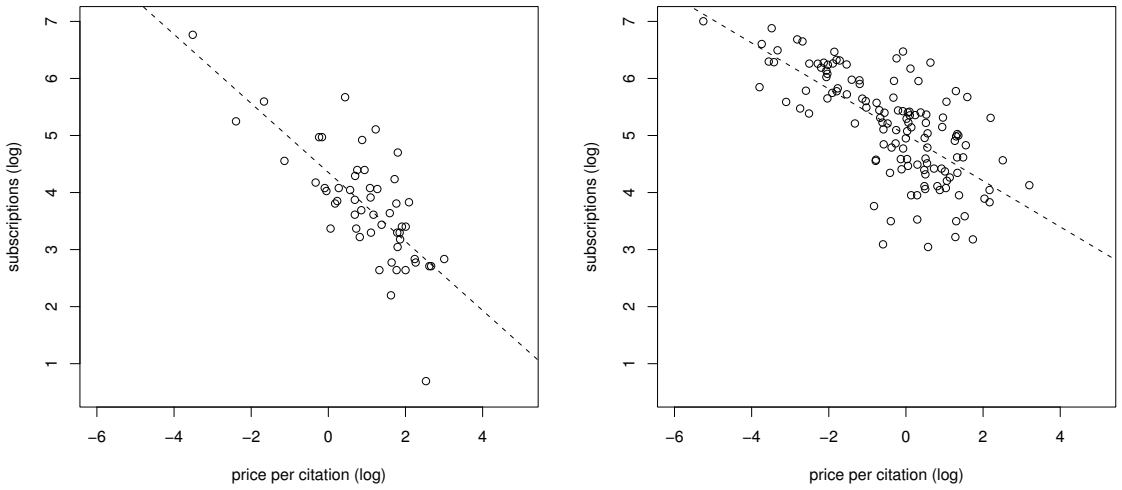


Figure 5.1.3: Subsets of journal pricing data set (Bergstrom, 2001) which were found with the SEM Tree algorithm. Left: Price elasticity for journals older than 19 years, Right: Price elasticity for journals younger than 19 years.

the tree was evaluated on the remaining OOB sample, i.e., the sample consisting of the journals not picked by the bootstrap method, to estimate the generalization performance of the model. Zeileis et al. (2006) reported results from 250 bootstrap samples. The median number of splits per tree was two, which parallels the results of the MOB algorithm. The median RMSE of Rpart was reported to be 0.804, whereas their MOB regression tree yielded a median error of 0.730. In the simulation with the Bonferroni-corrected split evaluation, the median RMSE on the OOB sample over 250 repetitions with the SEM Tree was at 0.702. Using SEM Trees with CV estimates is superior to all other methods, with a median RMSE of only 0.650. These results suggest that SEM Trees perform

5.1.3 Developmental Latent Growth Curve Model: Wechsler Intelligence Score for Children

The data for the following illustration of an application of SEM Trees was originally obtained by Osborne and Suddick (1972) between 1961 and 1965. A well-known and previously analyzed data set (McArdle & Epstein, 1987; McArdle, 1988) was chosen deliberately, in order to judge the performance of SEM Trees in the light of previous analyses. This data set was obtained by measuring 204 children on eleven different items from the *Wechsler Intelligence Scale for Children* (WISC; Wechsler, 1949). The children were followed longitudinally and were repeatedly measured at four time points, at the ages of six, seven, nine, and eleven years. Summary statistics and models were reported by McArdle and colleagues (McArdle & Epstein, 1987; McArdle, 1988). McArdle and Epstein (1987) described aspects of LGCMs for modeling the development of the underlying latent construct "intelligence." Intelligence was measured on four unidimensional subscales labeled "verbal" and four unidimensional subscales labeled "performance." The raw scores were rescaled to represent a "percent-correct" scaling, thus ranging between 0 and 100. Covariates included the dichotomous variables sex, race[2], age, and the years of education of mother and father. Only the covariate "father's education" had missing values. For the following analyses, I excluded the covariate "race" to avoid raising controversial discussions.

I set up an equivalent factor model for an analysis with SEM Trees based on the data as described by Ferrer, Hamagami, and McArdle (2004) and McArdle and Epstein (1987). The template model for the SEM Tree was a linear LGCM with a latent intercept and a latent slope, modeling a composite average score of the verbal and performance subscales on non-equidistant points in time. Centering the model on the first occasion of measurement (age 6), the respective factor loadings for the linear slope component are 0, 1, 3, and 5. The model was defined by six free parameters, mean μ_S and variance σ^2_S of the slope, mean μ_I and variance σ^2_I of the intercept, σ^2_e for the measurement error, and σ^2_{IS} for the correlation between intercept and slope. A SEM Tree was generated from the full data set of $N = 204$ observations with a minimum of $min_N = 20$ observations per leaf and a α level of 0.01.

The resulting tree of the CV method was in fact a pruned sub tree of the Bonferonni-constructed SEM-Tree. In Figure 5.1.4, the SEM-Tree with the p-values from the Bonferonni-corrected construction method are presented. According to the CV method, the left sub-tree was associated with a significant improvement of model fit. Both construction methods found the variable mother's education as primary split and found a subsequent split according to father's education in the right sub tree. The right sub tree with the split point *sex* was not

[2]This was the covariate label in the original data set.

chosen by CV and would not have been chosen with the Bonferonni-corrected method if the α level had been set to 0.01. This suggests that this split indeed should be ignored.

The educational level of children's mothers was previously found by McArdle and Epstein (1987) to be an informative predictor of individual differences in this data set. They investigated a binary variable in their model that represents whether the children's mothers graduated from high schools or not. Using this covariate as a causal influence on the latent variable, the authors found that the variable "high school graduation of mother" reflects differences in cognitive growth. They claimed that this could have been easily overlooked by other path analysis or MANOVA models. In their approach, a manual exploration of covariates led to the conclusion that "mother's education" predicts differences in the cognitive development of children. Again, one has to be aware that such manual exploration and model modification can be misleading. Among the set of investigated covariates, the SEM Tree confirmed that mother's education indeed predicts the largest differences with respect to the model parameters.

Both covariates, "mother's education" and "father's education", were originally coded with six categories, the first three representing various levels of high school graduation and graduation from higher schools, and three levels encoding for different levels of pre-high school education. Remarkably, the SEM Tree found a dichotomization of both covariates, mother's education and father's education, according to whether they had graduated from high schools or not. With SEM Trees, this plausible result was confirmed in a purely data-driven fashion that ensures validity and generalizabilty.

To examine the stability of the algorithm on this data set, I drew 100 bootstrap samples from the original data set. In each replication, I drew $N = 204$ samples from the original data with replacement to construct a training set. The remaining samples formed an independent test set. SEM Trees were generated from the training sets as described above and were evaluated on the test sets. In all 100 out of 100 replications, the SEM Tree algorithm found a model description with a higher likelihood of observing the test set under the SEM Tree than under a standard LGCM. In 33 of the bootstrapped trees, mother's education was chosen as first split variable. In all 33 trees, father's education was chosen as a single further split with respect to "father's education", thus replicating the tree structure obtained from the whole sample. In the remaining 67 bootstrapped trees, "father's education" was chosen as first split variable. In 2 out of the 67 trees, "mother's education" was chosen as subsequent split. The variables sex and age were never chosen. These results indicate that both parents' education variables predict key differences in the model parameters. Particularly, the variable "father's education" is a strong predictor (chosen in 98 of 100 replications). Presumably, the instability in the bootstrap results from a high correlation of both variables. From the tree analysis, we can conclude that a variable representing parents' educational background is highly predictive for parameter differences in the model, while the covariates "sex" and "age" are uninformative.

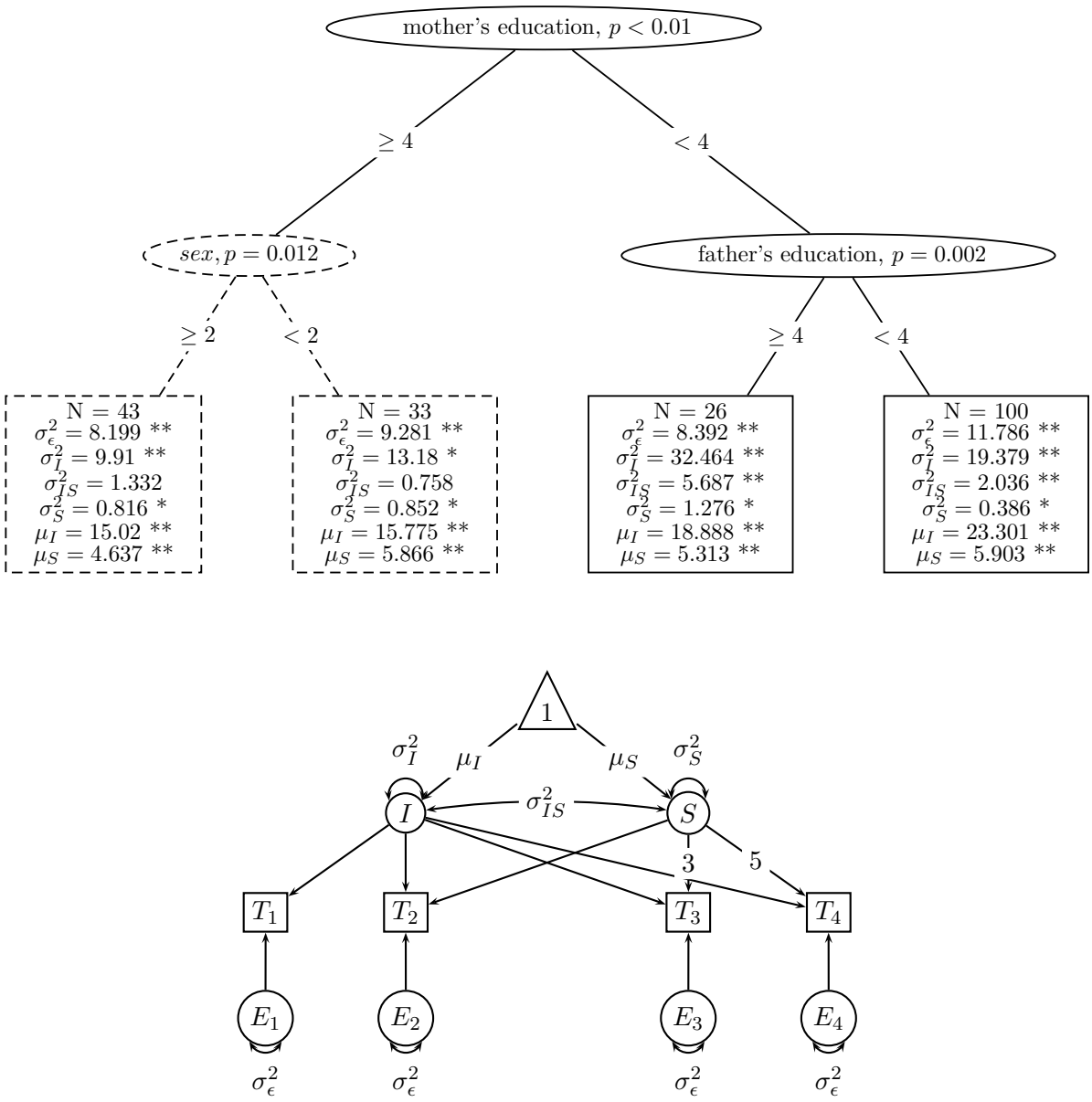


Figure 5.1.4: Upper panel: SEM Tree on the longitudinal WISC data set Osborne and Suddick (1972) . A linear LGCM served as the template model. The model hints that the sample is maximally heterogeneous with respect to mother's education, which explains interesting differences in the learning curves of both subsamples. The split points of the two covariates "father's education" and "mother's education" correspond to whether parents graduated from high school or not. The dashed line indicates the sub-tree that was not found when CV was employed as construction method. Lower panel: Graphical representation of a linear LGCM which served as the template model for the SEM Tree above. Model parameters represent the intercept $I \sim \mathcal{N}(\mu_I, \sigma^2_I)$, the slope $S \sim \mathcal{N}(\mu_S, \sigma^2_S)$, the covariance between both σ^2_{IS}, and the residual error term σ^2_ϵ.

5.1.4 Factor Model SEM Tree for the WAIS-R Data Set

The *Wechsler Adult Intelligence Scale Revised* (WAIS-R) is a set of cognitive tests, similar to the WISC test presented in the previous example, but targeting adult participants. I demonstrate a SEM Tree analysis with a factor model tree with measurement invariance. Again, the analysis is based on a data set that was previously analyzed by others (Horn & McArdle, 1992; McArdle & Prescott, 1992), which allows us to compare our results with preceding analyses. The analyzed sample was collected between 1976 and 1980 by the Psychological Corporation and includes the performance scores of 1880 individuals on 11 WAIS-R subscales. A rich set of demographic covariates is available for this data set. The number of induced dichotomous covariates for split candidate evaluation was 70. The investigated covariates are listed in the following; the former number in brackets indicates the number of categories of each variable and the latter the number of induced covariates: *agegroup* (9/8), geographical information about the *place of residence* (4/7), *urban/rural place of residence* (2/1), *place of birth in the United States* (2/1), *marital status* (4/7), *handedness* (2/1), *education* (6/5), *occupation* (6/31), *sex* (2/1), and *birth order* (9/8).

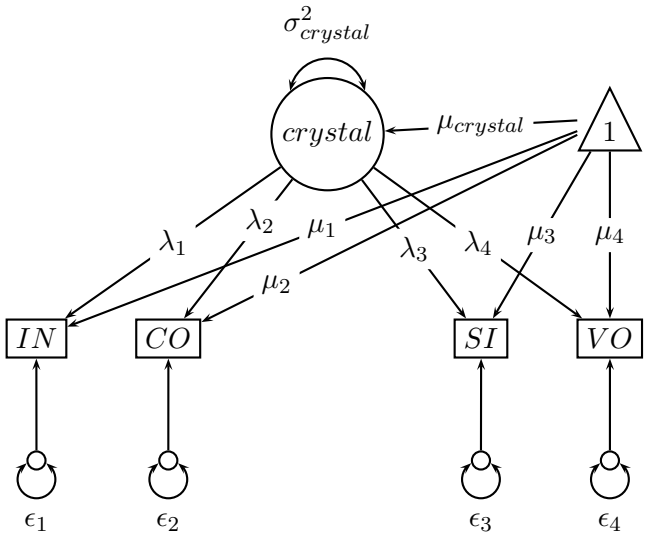


Figure 5.1.5: A latent factor model with a single latent factor representing crystallized intelligence by four observed test scores from the WAIS-R test: Information (IN), Comprehension (CO), Similarity (SI), and Vocabulary (VO). The latent construct is modeled as $crystal \sim \mathcal{N}\left(\mu_{crystal}, \sigma^2_{crystal}\right)$. The factor loadings are modeled as λ_1, λ_2, λ_3, and λ_4. Observed expectations are modeled as μ_1, μ_2, μ_3, and μ_4. The corresponding residual error terms are ϵ_1, ϵ_2, ϵ_3, and ϵ_4.

Following McArdle and Prescott (1992), I included all individuals aged 18 or older, obtaining a sample size of 1,680. I set up a single factor model that hypothesizes a single latent factor representing crystallized intelligence, measured on four verbal performance scores. The factor

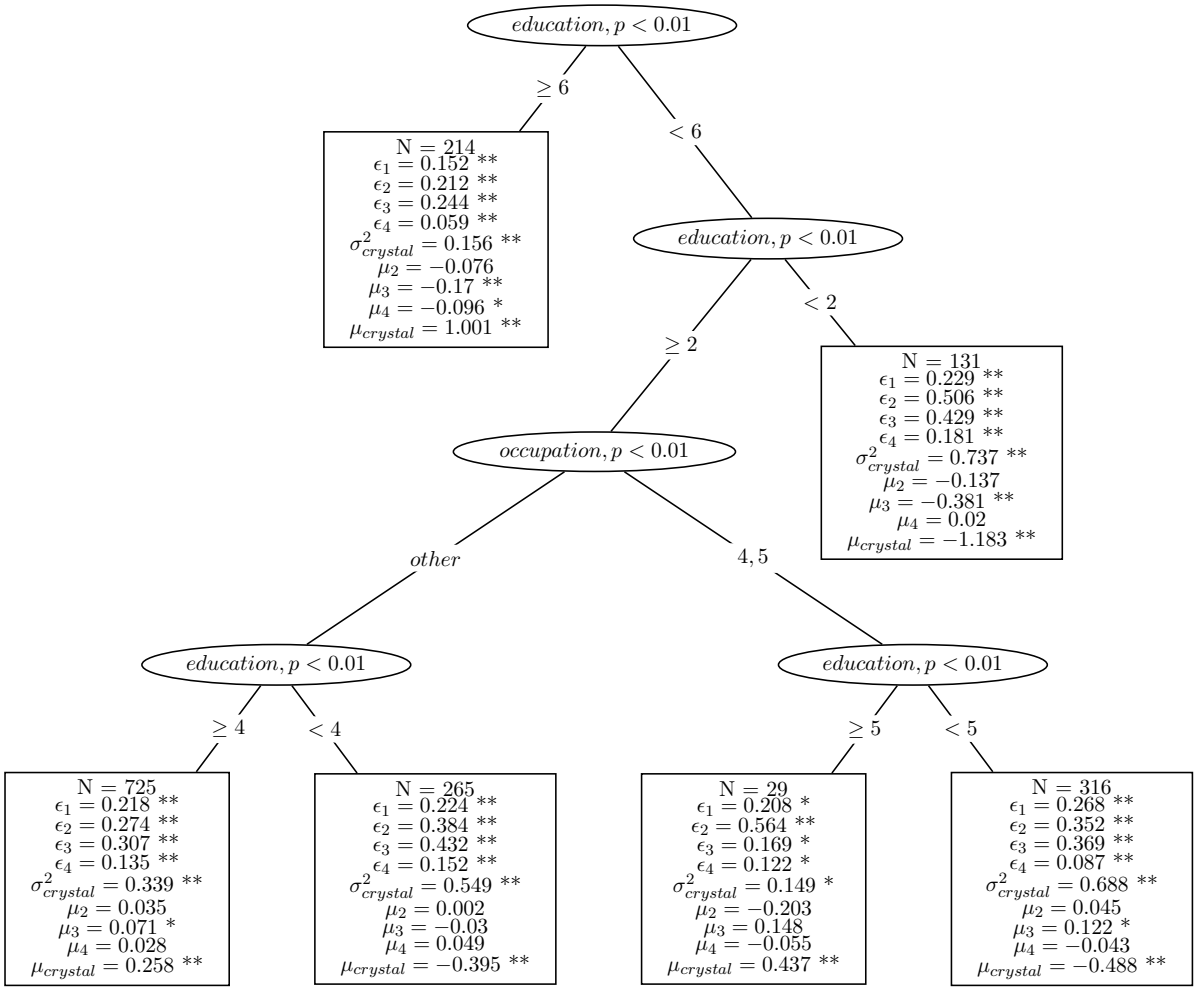


Figure 5.1.6: SEM Tree with a factor model representing a single factor of verbal IQ, which was observed on four subscales (Information, Comprehension, Similarities, and Vocabulary) from the WAIS-R data set for 1680 individuals. The resulting splits indicate that the demographic covariates *education* and *occupation* predict substantial differences in the hypothesized model.

is represented by a latent mean $\mu_{crystal}$ and a latent variance $\sigma^2_{crystal}$. The factor loading of the Information score (IN) was fixed to $\lambda_1 = 1$ and the corresponding expectation of the observed score to $\mu_1 = 0$. Free parameters in the model include the remaining factor loadings λ_2 (Comprehension) , λ_3 (Similarities), and λ_4 (Vocabulary), and the residual errors ϵ_1, ϵ_2, ϵ_3, and ϵ_4 of the four scores. The resulting model is sketched in Figure 5.1.5. The factor loadings were chosen to be tested for weak measurement invariance, i.e., only candidate splits that fulfill the invariance of the factor loadings' values, were considered for further evaluation. The first four levels of the resulting SEM Tree under the factorial invariance assumption is shown in Figure 5.1.6. The tree structure shows that two variables seem to primarily influence partitions with respect to a model of a crystallized intelligence factor. The first two splits concern the levels of the education variable. They split off the two extreme groups from the data set,

those with an education of 0–7 years, encoded in the data set as "education = 1" and those with an extremely long education of 16 or more years, encoded as "education = 6". For both extreme groups, there is no subsequent split that allows for an improvement in model fit. For the remaining 1,335 individuals with an education between 7 and 16 years, occupation explains the largest difference with respect to model parameters. The resulting two partitions are semi-skilled and skilled workers, encoded as "occupation = 4, 5" and all other occupations. For both subgroups, again, education describes the largest difference depending on whether participants had an education of more or less than 12 years (left sub-tree) or 13–15 years (right sub-tree). It is very unlikely that the above structure would have been retrieved by a manual approach. Even if the variables were selected a-priori, it is likely that the partition would have been selected according to a heuristic criterion. With SEM-Trees the selection of the variable split point is founded on an information-theoretic criterion that provides evidence that the induced difference generalizes to the population. Thereby, researchers bias is minimized and the initial model is refined with additional predictive information provided by the selected covariates.

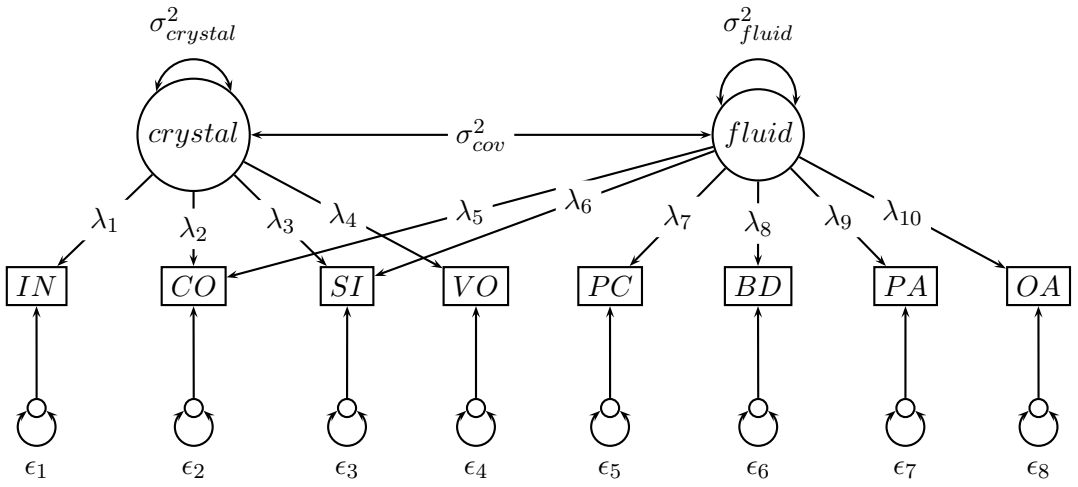


Figure 5.1.7: Latent factor model with two hypothesized latent factors representing fluid intelligence and crystallized intelligence. The two factors load on eight cognitive tests: Intelligence (IN), Comprehension (CO), Similarity (SI), Vocabulary (VO), Picture Completion (PC), Block Design (BD), Picture Arrangement (PA), and Object Assembly (OA). Two latent factors model variance of crystallized intelligence with $\sigma^2_{crystal}$ and fluid intelligence with σ^2_{fluid}. The covariance between both constructs is σ^2_{cov}. The relation between the factors and observed variables is modeled by ten factor loadings λ_1 to λ_{10}. The residual errors for the eight observed variables are ϵ_1 to ϵ_8.

McArdle and Prescott (1992) examined a set of more complex models that varied in the number of latent factors, in the structure of the factor loadings, and in the number of covariates that were regressed on the latent factors. They concluded that the best fitting model within the

scope of models investigated in their article included two latent factors, representing crystallized intelligence and fluid intelligence. Crystallized intelligence are skills based on earlier learning, whereas fluid intelligence refers to the ability to adapt to new situations (Cattell, 1963). Furthermore, the model included the variables "age" and "high school graduation," a dichotomous covariate that was manually constructed from the categorial variable "education," as regressors on both latent variables. The variables were found in a manual exploratory approach by plotting correlations of different covariates and target variables, and then comparing likelihoods and fit indices for the competing models. As mentioned earlier, this process can help to find better representations of the sample but contains an immanent danger of overfitting the data. In the following, I demonstrate the analysis of this data set with a SEM Tree regarding the factor model proposed by McArdle and Prescott (1992). However, instead of modeling the covariate interactions as linear influences on the latent variables,I will analyze which covariates or covariate interaction are discovered by a SEM Tree. The previously applied model of cognitive capabilities with a single factor is therefore extended to a two-factor model with factor loadings as suggested by McArdle and Prescott (1992). A graphical representation of the model is shown in Figure 5.1.7, and the resulting SEM Tree is depicted in Figure 5.1.8. Most noteworthy, the tree retrieves covariates that include the variables "age" and "education", as previously found by McArdle and Prescott (1992). Age is the first covariate selected for a split. Subsequent splits again include the covariate "age": The selected split points of the variable "age" were 24 years ($age = 3$), 44 years ($age = 5$), and 64 years ($age = 7$). This implied split of the covariate in equidistant groups potentially reflects the proposition of McArdle and Prescott (1992) that age indeed has a linear influence on model differences. The covariate "education" is also contained in the tree and indicates a separation of the extreme group ($education \geq 6$) that represents persons with 16 or more years of education. The covariate "occupation" induces a tri-partition into (1) a group that subsumes technical and professional staff, managers, officials, and skilled workers ($occupation = 1, 2, 3$), (2) a group of semi-skilled workers ($occupation = 4$), and (3) a group of unskilled workers and unemployed persons ($occupation = 5, 6$). The occupation variable was declared categorial and, thus, potential splits were searched for all possible subsets. The tree recovered splits according to the underlying ordered scale ranging from higher achieving jobs to less achieving ones. Furthermore, in the right sub-tree, for all participants younger than 20 years ($age \leq 3$), the variable describing the region of their birth yielded the most prominent split, discriminating between the South of the U.S. ($region\,direction = 3$) and the rest. This could indicate geographical cohort differences in the educational systems. Most importantly, the results indicate that exactly the same covariates are important as were found in the previous manual exploratory approach. The selection rules of the tree avoid overfitting the sample. In addition, the tree retrieved two more covariates that promise to explain differences with respect to the model parameters.

Given the large number of covariates and possible splits, this would never have been found by a manual search of split points. Furthermore, a manual multi-group analysis might not have included all covariates due to potential selection bias of the investigator. SEM Trees succeeded in finding covariates that predict differences in the model parameters that were previously identified by hand, and extends these results to further covariates with predictive power.

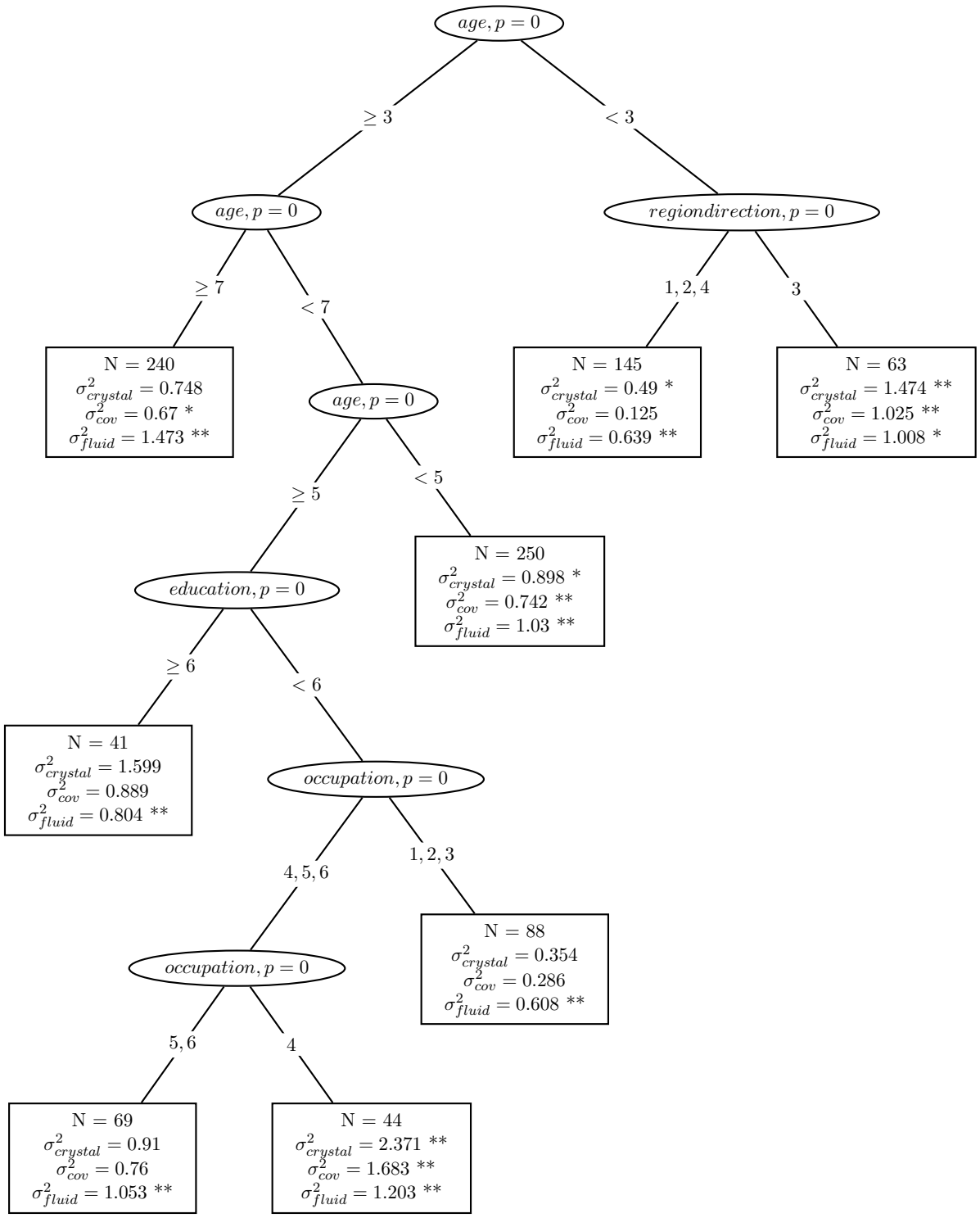


Figure 5.1.8: Two-factor SEM Tree for the WAIS-R data set McArdle and Prescott (1992). The template model contains two factors: Crystallized and fluid intelligence. The tree retrieves model differences in subgroups determined by age and education. This replicates previous findings. Additionally, the variables "occupation" and "geographic region" predict differences in the model parameters.

5.1.5 Variable Imputation With SEM Trees

The following example demonstrates the capability of SEM Trees for model-based imputation. Again, I consider the WISC data set (see Subsection 5.1.3). Based on the linear LGCM (see Figure 5.1.4), I compare two variants of model-based imputation. The WISC data set provides four observed variables, which are composite scores of a set of intelligence tests. The data set contains test scores for 204 persons. Originally, the test scores had no missing values. To demonstrate the potential of SEM Trees to impute data, I simulated missing values by corrupting observations under MCAR assumptions. This was done by uniformly choosing 30% of the data points and removing them from the data set, effectively replacing their values with a marker for missing values. A SEM Tree was generated from this incomplete data set. In a second step, the SEM Tree was used on the same data set to infer the missing values. In this example, I estimated the missing values by a maximum likelihood estimate, i.e., we replaced each missing value by the expectation of the conditional distribution as described in Formula 3.6.1 in Section 3.6.3. In other applications, it could be interesting to draw values according to the conditional distribution. However, in this context, we were interested in how well imputed values match the true values in expectation.

In order to evaluate the goodness of the imputed values, I calculated the RMSE of reconstruction. For a total of n missing values, let y_i be the true value for the i-th missing value in a data set and let $\hat{y}_i$ be its imputed estimate. The RMSE of reconstruction for that data set is given as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

In order to report a normed coefficient, I report the coefficient of variation of the RMSE (CVRMSE) as

$$CVRMSE = \frac{RMSE}{\bar{y}}$$

with $\bar{y}$ being the empirical mean across the missing values. Another common measure to quantify the relation of predicted values and true values is the coefficient of determination, which is the squared correlation of both variables, indicating the common variance of both variables under the assumption of a linear relation between them.

As a comparison, I ran the same imputation mechanism with the template model only, i.e., data was imputed using a single SEM for the complete data set only. The average reconstruction error of the SEM Tree was 6.19 in comparison to a higher error of 7.45 units with the template SEM only. By using SEM Trees, the coefficient of determination between true values and imputed values increased from 0.83 to 0.88.

The analysis was repeated 100 times with a bootstrap sample of the original data set. Averaged across the 100 trials, SEM Trees could increase the accuracy of the predicted values by 21%, decreasing the absolute RMSE from 8.10 to 6.39 units. The average coefficient of determination was increased from 0.63 to 0.77. A two-sided paired t-test ($t = -12.5411, df = 2, p = 0.0063$) confirmed the significant improvement of RMSE when using SEM Trees rather than the template LGCM only.

The presented example shows that SEM Trees have the ability to increase model-based imputation approaches that rely on SEMs. By exploiting the structure of the covariate space, more fine-grained representations of the multivariate outcomes of interest can be retrieved. This allows a better recovery of missing values in the observed values of a SEM than using the SEM alone.

5.2 Applications of PDC

Turning to the method presented in Chapter 2, I will present some demonstrations of PDC on various data sets. Before showing results, I briefly discuss the approach to evaluation of PDC. I introduce a quality measure based on a known ground-truth for a data set and briefly review other reference clustering approaches that are compared to the new algorithm.

For the illustration of time series analyses with PDC, the time series are plotted together with dendrograms. These show the hierarchical clustering resulting from the application of PDC with the average linkage function. By default, PDC is used as a parameterless method by using the MinE heuristic for the determination of the embedding dimension and, if not stated otherwise, the AIC criterion for determining potential splits of the hierarchical clustering in separate clusters (see Section 2.11). Figure 5.2.1 shows the entropy criterion for some data sets presented in this chapter for varying embedding dimensions. According to the MinE criterion, the embedding dimension for each data set that minimizes the average normed entropy of all time series in the data set is chosen. The respective minima are depicted as diamonds in the plots.

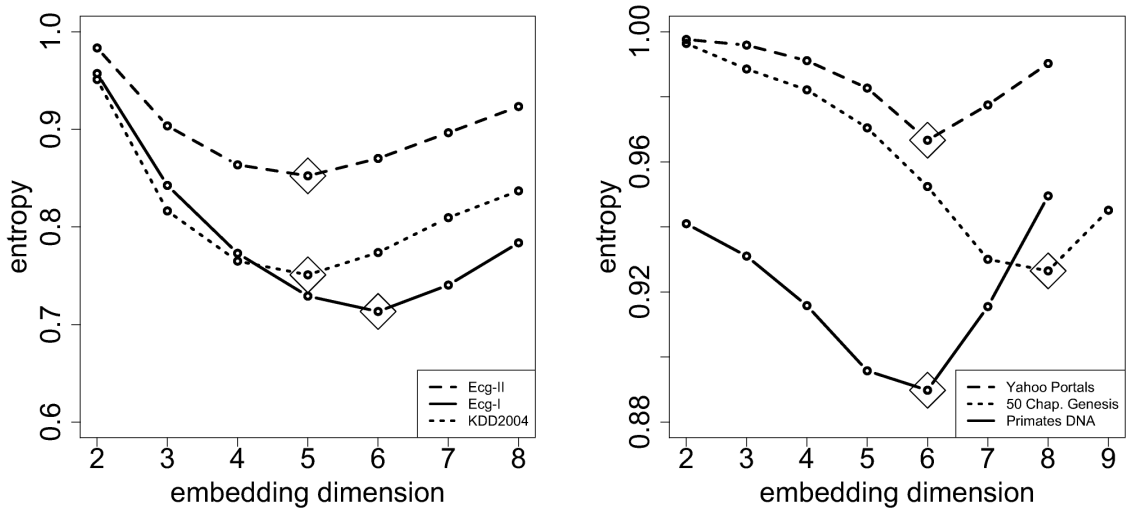


Figure 5.2.1: MinE criterion for varying embedding dimensions for six different data sets. The minimum of the average entropies that determine the automatic choice of the embedding dimension are marked with diamonds on the respective curves. Left: Entropy curves for the data sets Ecg-I, Ecg-II, and KDD2004 (see Subsection 5.2.4 and 5.2.5). Right: Entropy curves for the data sets Yahoo Portals, 50 Chapters of Genesis (see Subsection 5.2.6), and Primates DNA (see Subsection 5.2.7).

5.2.1 Comparing Clustering Approaches

In the following, an overview of important competitors of PDC is given. Furthermore, a measure to evaluate the quality of clustering, given a known ground-truth, is defined.

Baseline Candidates for Clustering

Keogh and Kasetty (2003) criticized that most clustering approaches are rarely evaluated against enough competitors and on a sufficiently large number of data sets. As an extreme case, they presented performances of 11 distinct clustering algorithms proposed in the literature on two freely available[3] data sets, "cylinder-bell-funnel" and "control charts". They showed that none of these algorithms can beat the simplest straw man, the Euclidean distance employed as dissimilarity measure between time series. Therefore, they generally suggested using Euclidean distance and dynamic time warping (DTW; Berndt & Clifford, 1994) as baseline methods to which new clustering algorithms should be compared.

The Euclidean distance between two time series X and Y of the same length T is defined as

$$D_{EUC}(X,Y) = \sum_{i=1}^{T} (x_i - y_i)^2$$

[3]Both datasets are available in the *UCR Time Series Data Mining Archive* (Keogh & Folias, 2002).

DTW is a dynamic programming approach to the derivation of a cost-measure between time series that naïvely is the pair-wise absolute distance between time points but allows local stretching and compression of each time series.

In order to evaluate the results of PDC, I also ran a different information-theoretic clustering variant, an approximation of the universal metric based on an approximation of Kolmogorov complexity via string compression (M. Li et al., 2004), which Keogh, Lonardi, and Ratanamahatana (2004) confirmed as a versatile measure for clustering. The Kolmogorov complexity of an object, as introduced by M. Li and Vitányi (2008), is defined as the length of the shortest program on an universal computer that generates this object. This complexity measure is incomputable but can be approximated by common string compression methods (e.g. M. Li et al., 2004). I approximate Kolmogorov complexity by discretizing the continuous signals into 256 bins and thus producing an ASCII-string for each channel of the time series. These strings are compressed with the *zlib* algorithm (Gailly & Adler, 2004), a variation of Ziv and Lempel's (1977) LZ77. Formally, the compression distance is given as:

$$D_{ZIP}(X,Y) = \frac{Z(x \circ y) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}$$

with "$\circ$" being string-concatenation and $Z(x)$ being the length of the compressed string x. Keogh, Lonardi, and Ratanamahatana (2004) reported clustering results with a similar approach. They first performed a piecewise constant approximation, which was then discretized in a finite alphabet such that each letter occurs with equal probability over the time series. They formulated their compression-based dissimilarity measure (CDM) as follows:

$$D_{CDM}(X,Y) = \frac{Z(x \circ y)}{Z(x) + Z(y)}$$

Evaluating the Clustering Quality

Whenever a known ground truth is available, clustering can be evaluated in a supervised learning setting (Keogh & Kasetty, 2003). Given a training set with true labels and a clustering result, the clusters can be seen as a classification of the training set in classes corresponding to clusters. A new observation is classified as belonging to the cluster which is most similar to the observation. Similarity is usually defined in the Euclidean metric. The percentage of clusters that are grouped with a nearest neighbor that matches their class leads to the final accuracy estimate. A high accuracy indicates a successful clustering. This procedure is known as leave-one-out one-nearest neighbor (LOO-1NN; Keogh & Kasetty, 2003). An efficient implementation of the LOO-1NN scheme, given a distance matrix, is provided in Algorithm 5.1. Note that this method is an evaluation of the distance metric, regardless of the actual clustering algorithm used to generate a clustering structure.

In some of the following cases, there is no ground truth about the full clustering structure available. In these cases, I employ ad-hoc measures to evaluate, whether the clustering reflects our structural assumptions about the data.

Algorithm 5.1 One-nearest-neighbour scheme with leave-one-out cross-validation, based on a dissimilarity or distance matrix of the observations.

```
function LOO-1NN(M,L) ▷ Let M be a matrix of divergences between time series and let
L be a vector of classes of the time series
    c ← 0
    for i in 1 to rows(M) do
        M_{i,i} ← ∞
        m ← j that minimizes M_{i,j}
        if L_i == L_m then
            c = c + 1
        end if
    end for
    return c/M · 100
end function
```

5.2.2 Time Series Segmentation on Accelerometer Data

This example illustrates time series segmentation, i.e. the detection of continuous subsequences of a time series that significantly differ from the remaining segments. PDC was used to segment an accelerometer recording of two persons walking at different speeds on a treadmill. Each person wore an *Apple iPod Touch* in their front left trouser pocket, which recorded the three-dimensional sensor stream of the built-in accelerometers. Segmentation was carried out by clustering windows of the original time series of size 1000. The windows overlapped by one half of their length. An illustration of both participants' results for is given in Figure 5.2.3. For each person, only a single dimension of the time series is plotted. Both walked on the treadmill with the treadmill speed fixed at the levels 3 km/h, 4 km/h, 5 km/h, 6 km/h, and 7 km/h. Between changes of speed, the treadmill was brought to a full stop. A successful clustering should therefore indicate homogeneous segments for each segment with the same treadmill speed, and also retrieve the full stops of the treadmill in-between. The number of clusters for each participant was determined with the AIC and again with the BIC (cf. Section 2.11). The embedding dimensions were automatically chosen according to the MinE heuristic. The average entropy curves for both participants are shown in Figure 5.2.2. The figure contains average entropy for each channel of the tri-axial accelerometer, and an average score. As previously defined, the embedding dimension was chosen based on the average curve. Taking a closer look at the individual channels, the MinE criterion suggests that indeed individual choices for each dimension can make sense. Also, the average entropy curves for both participants show that channel Z, representing the vertical axis, has the lowest representation throughout all embedding dimensions. Participants wore the iPhone such that the antero-posterior axis, the main axis of movement, was covered by both X and Y channels of the accelerometer. It seems reasonable that more discriminative information is contained in the X and Y axis. This information from the MinE heuristic helps to choose informative channels from a multi-channel recording.

Regarding the clustering result as shown in Figure 5.2.3, PDC can indeed retrieve a useful segmentation of the time series. Full stops of the treadmill are successfully identified and different walking speed segments are found. In this example, the BIC-based segmentation

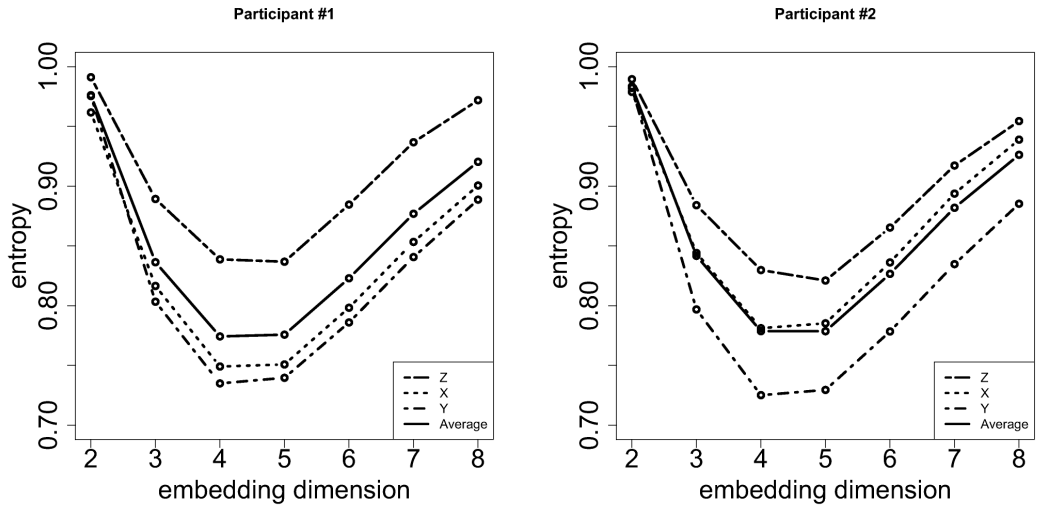


Figure 5.2.2: Embedding dimension heuristics for the two walking speed data sets of participant 1 (left) and participant 2 (right). For each participant, the average entropy per embedding dimension is plotted per channel of the accelerometer sensor, and the average over the three sensors is plotted. The minimum of the average curve is depicted with a diamond. According to the MinE criterion, the embedding dimension that minimizes the average entropy is chosen.

tends to find less clusters than expected, whereas the AIC tends to over-segment. That is, for both participants, the AIC sometimes divides segments of homogeneous walking speed into two segments, whereas BIC-based segmentation does not always divide clusters of different walking speeds. However, the ground truth is unclear. We cannot exclude that, for instance, the respective walker's stability changed over time, and thus a complexity-based division of an apparently speed-homogeneous segment actually makes sense. Therefore, it is reasonable to restrict the analysis to a descriptive summary here, which gives a convincing picture that time series segmentation with PDC is a helpful tool in the process of data analysis.

5.2.3 Clustering Electroencephalographic Data

Permutation entropy was previously applied to biomedical signals, e.g., it was used as a feature in a classification task to predict absence seizures (X. Li, Ouyang, & Richards, 2007) or the fetal behavioral states from bio-magnetic recordings (Frank, Pompe, Schneider, & Hoyer, 2006). In this section, I apply PDC to resting state EEG recordings from five participants under two conditions: Eyes open and eyes closed. Data were provided by the COGITO study (Schmiedek, Lövdén, & Lindenberger, 2009). Generally, power in the 8–12 Hz frequency band, which is also called the *α-power*, is suggested as a measure of resting-state arousal (Barry, Clarke, Johnstone, Magee, & Rushby, 2007). In the following example, I examine whether PDC can find partitions of the underlying dynamics of the two conditions in a high-dimensional data set.

Usually, preprocessing is a crucial and labor-intensive process prior to the analysis of EEG signals. However, in this demonstration, the available data was not artifact-corrected. This is

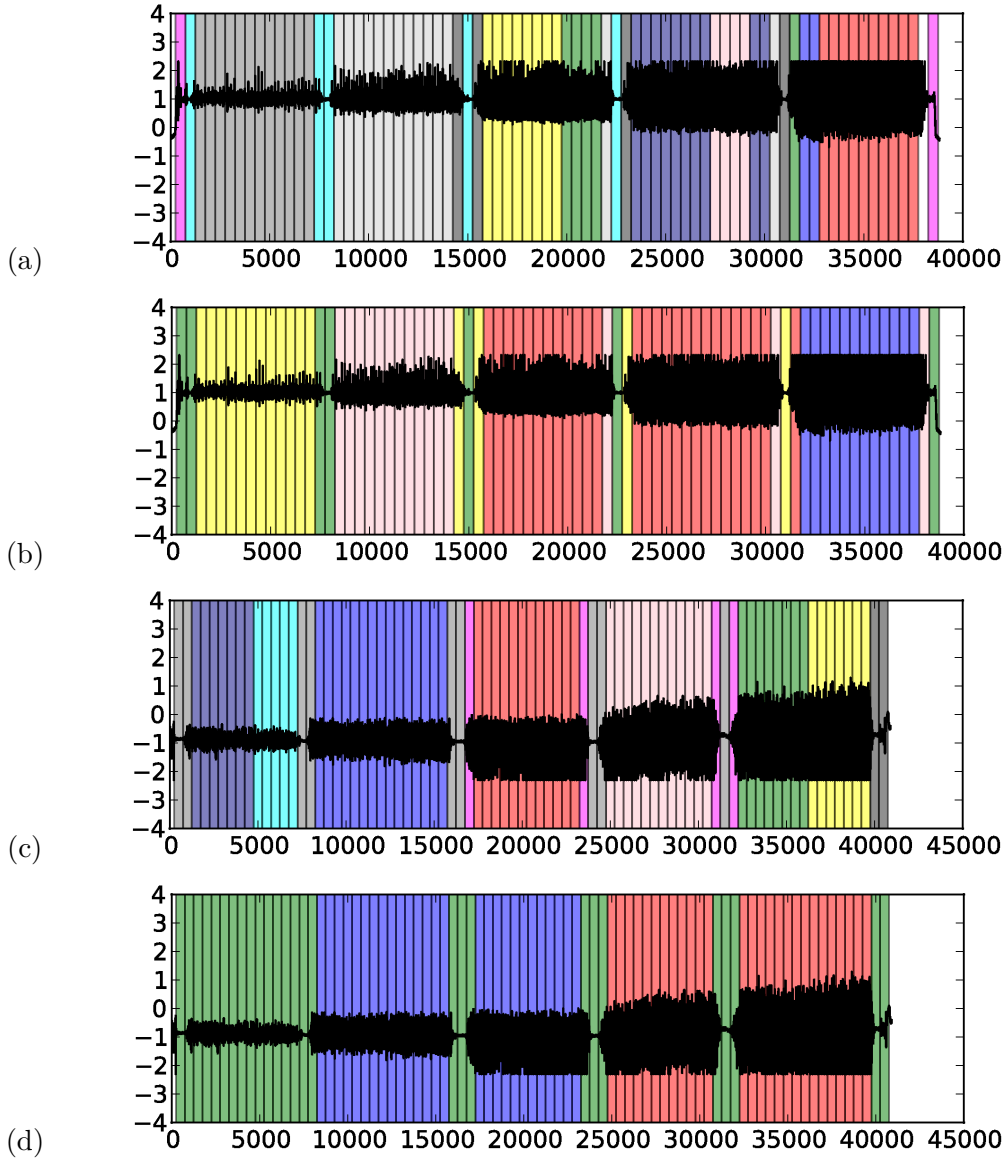


Figure 5.2.3: Illustration of time series segmentation with PDC on accelerometer data. Time series from two different persons walking on a treadmill at different speed with full stops of the treadmill between changes of the walking speed are shown: (a) Participant 1 with a segmentation based on the AIC; (b) Participant 1 with a segmentation based on the BIC; (c) Participant 2 with a segmentation based on the AIC; (d) Participant 2 with a segmentation based on the AIC. The walking speeds from left to right were 3, 4, 5, 6, and 7 km/h with full stops in-between. Only one dimension out of the three is plotted (black line), indicating the movement of the participant along the anterior-posterior axis. Colors within a time series indicate sub-sequences of the time series that were clustered together. Note that all four analyses were run separately and therefore cluster colors do not correspond across the time series. The PDC successfully recovers segments with increasing speed of walking.

to show that PDC can operate well on raw data. Data were recorded with an EasyCap EEG electrode cap (Brain Products GmbH) with 64 electrodes. Each electrode recording was centered and normalized to unit variance. For the clustering, I excluded the electrooculogram electrodes which explicitly measure eye movements, and might simplify the task. The continuous segments of each condition of each trial were cut into non-overlapping segments of 1,000 sample points each, and five segments from each participant and each condition were selected.

	Method					
Participant	EUC	DTW	ZIP	PDC3	PDC4 (MinE)	PDC5
1	70%	70%	80%	100%	100%	90%
2	40%	40%	70%	80%	90%	90%
3	10%	40%	60%	70%	70%	80%
4	50%	80%	100%	100%	100%	100%
5	80%	90%	70%	100%	100%	100%
Average Percentage	50%	64%	76%	90%	92%	92%

Table 5.2: Clustering accuracy on EEG data for the Euclidean Distance (EUC), Dynamic Time Warping (DTW), compression-based distance (ZIP), and Permutation Distribution Clustering (PDC) with varying embedding dimensions (PDC3, PDC4, and PDC5). An embedding dimension of 4 was chosen for each individual separately using the MinE criterion. The table lists the 1NN-LOO accuracy and states the number of grouped segments out of a total of 10 in percent correct. The bottom row states the empirical average accuracy of each method.

Clustering results are given in Table 5.2. The clustering was performed with several algorithms based on the Euclidean distance, dynamic time warping, and PDC with varying embedding dimensions. The embedding dimension was additionally evaluated with the MinE criterion. For each participant and each algorithm, the table lists the number of correctly grouped segments out of 10 according to the 1NN-LOO scheme. On average, PDC achieves the best clustering results in comparison to the other methods, achieving a 1NN-LOO accuracy of 92% averaged across five participants. A dendrogram of the clustering resulting from an analysis of participant 1 using PDC with $m = 6$ is shown in Figure 5.2.4. It can be seen that this result is highly more accurate than complexity based clustering, DTW, and the naïve Euclidean distance, which performed mostly at chance.

5.2.4 Clustering Electrocardiographic Data

The utility of PDC for the analysis of biophysical signals will be illustrated by two further examples. The original data sets were made available by Keogh, Lonardi, and Ratanamahatana (2004). Both data sets consist of univariate time series of electrocardiography (ECG) recordings. Clustering results of PDC are illustrated in Figure 5.2.5. For the first data set, referred to as *ECG-I* in the following, the authors randomly extracted ten subsequences each from two ECG databases. The data set consists of 20 time series of length 2,000. As the authors report, the ground truth of the lower level clustering is unclear, but the top level split in a hierarchical clustering should reflect that data originates from two different sources. The same applies to the second data set (*ECG-II*) that contains ECG recordings of four different persons: Two

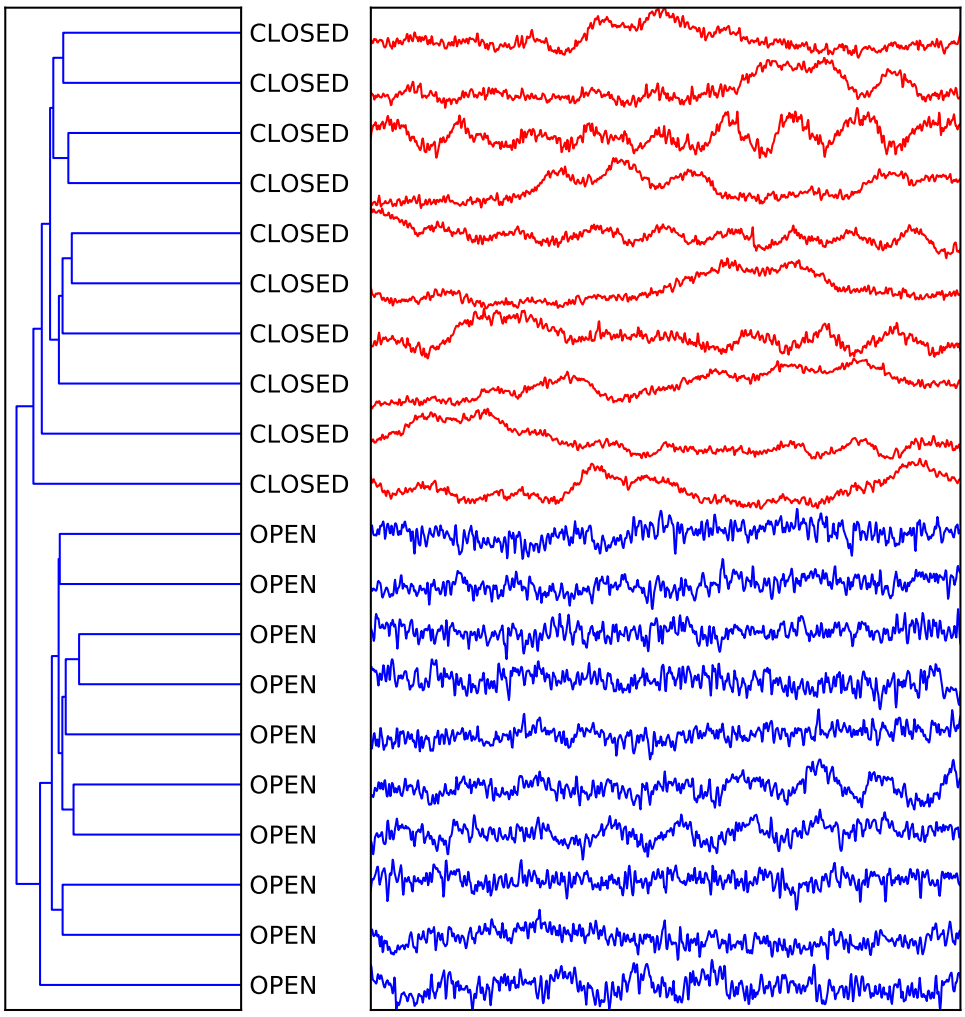


Figure 5.2.4: Dendrogram of a PDC analysis of time segments from EEG recordings of participant 1 with eyes open (blue) and eyes closed (red). The clustering was based on 60-dimensional time series. For clarity, only a single channel corresponding to electrode *Fp1* is plotted. The embedding dimension four was automatically determined with the MinE criterion. Each segment had a length of 1000. Both the AIC and the BIC indicate a split into two clusters as above.

	Method						
Data Set	EUC	DTW	ZIP	PDC3	PDC4	PDC5	PDC6
ECG-I	80%	100%	100%	100%	95%	100%	100% (MinE)
ECG-II	75%	-	100%	95%	100%	100% (MinE)	100%

Table 5.3: Comparison of base line clustering methods on electrocardiographic data. PDC was evaluated for embedding dimensions 3–6. The embedding dimension chosen by the MinE criterion is denoted in the table. Both PDC and the ZIP based clustering algorithm perform perfectly. The Euclidean distance performs worse but still on a surprisingly high level. The calculation of the DTW for ECG-II was terminated after 24 hours.

subjects from the *BIDMC Congestive Heart Failure Database* (first participant in the green cluster, second participant in the blue cluster), one participant from the *Long Term Stress Test Database* (yellow), and one participant from the *MIT-BIH Noise Stress Test Database* (salmon color). The *ECG-II* data set consists of 20 time series of length 10,000. Analogously, a hierarchical clustering should reflect four different sources in the data set, while we have no knowledge about what constitutes the optimal clustering of the subsequences of a single participant.

Indeed, for both ECG-I and ECG-II, the hierarchical clustering obtained from PDC matches the clustering obtained by the original authors with their compression-based method, on the level that distinguishes the participants. A comparison of the base line algorithms' performance is given in Table 5.3. As reported by Keogh, Lonardi, and Ratanamahatana (2004), the ZIP-based clustering performs perfectly. The PDC performs perfectly. For lower dimensions than chosen by the MinE heuristic, the PDC performs below optimum, presumably because the most powerful embedding dimensions has not yet been chosen. DTW performs also perfect for the ECG-I data set. For the ECG-II data set, the calculation of DTW was stopped for its excessive computation time beyond 24 hours.

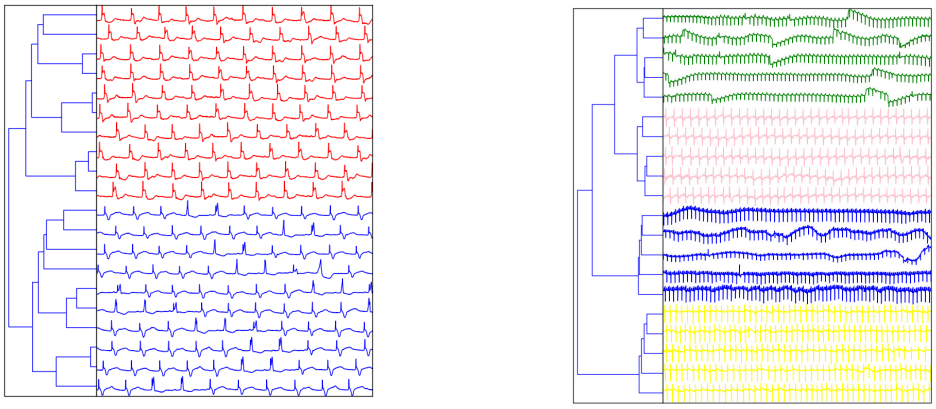


Figure 5.2.5: Clustering result of PDC on two time series data sets. Colors correspond to ground truth. Embedding dimensions were automatically chosen with the MinE criterion. Left: ECG-I data set with two classes. Clustering was executed with an embedding dimension of 6. Right: ECG-II data set. Clustering was performed with an embedding of 5.

5.2.5 Clustering on the KDD2004 Data Set

Keogh, Lonardi, and Ratanamahatana (2004) also presented an important data set for the evaluation of clustering algorithms. They collected 18 pairs of time series from the *University of California Riverside (UCR) Time Series Archive* that greatly differ in their characteristics. The data set is freely available from the authors[4]. The data set comprises various time series that include yearly power demand of different countries, the number of sunspots per year, and ECG data. Therefore, a clustering in which each pair of time series is closest to each other is assumed as the ground truth. The authors ran 51 different clustering approaches on this data set, evaluating them with different linkage metrics and reporting only the best for each algorithm.

I ran PDC on the data set. For this particular task, complete linkage promised the best results. Figure 5.2.6 illustrates the results of PDC on the set of time series using complete linkage and an embedding dimension of $m = 6$. Matching pairs of time series are indicated in the same color. Keogh, Lonardi, and Ratanamahatana (2004) use an ad-hoc measure for evaluating this task. Their Q-measure counts the number of correctly retrieved pairings at the lowest clustering level divided by the total number of pairs. Thus, the measure evaluates the known structure of the lowest clustering level without regarding the unknown higher-level structure. The authors report that more than $3/4$ of the clustering approaches that they have tested yielded the worst possible score of $Q = 0$. The best results were achieved by the simple Euclidean distance with

[4]It can be downloaded at http://www.cs.ucr.edu/~eamonn/SIGKDD2004.

$Q = 0.27$, dynamic time warping with $Q = 0.33$, piecewise linear approximation with $Q = 0.33$, LPC Cepstra, autocorrelation with $Q = 0.16$, and LCSS with $Q = 0.33$. A compression-based clustering was clearly superior on the data set, yielding a $Q = 1.0$. PDC with an embedding dimension of 5 can keep up very well with 15 out of 18 time series in perfect pairs at the lowest levels ($Q = 0.78$). The lower level clustering seems to improve if an embedding dimension higher than determined by the criterion is chosen. With an embedding dimensions of 6, 16 out of 18 time series form pairs on the lowest level. However, an evaluation of the 1NN-LOO accuracy hints that the embedding dimension chosen by the MinE criterion is optimal: Table 5.4 reports the 1NN-LOO accuracy of PDC for different dissimilarity measures and varying embedding dimensions. The Hellinger distance achieves the best result on the embedding dimension suggested by the MinE criterion (also see Figure 5.2.1, left). This result provides empirical evidence that the MinE criterion chooses the optimal embedding dimension, and that the Hellinger distance outperforms the symmetric Kullback-Leibler divergence, Euclidean distance, and the χ^2-divergence. I complete this internal validation with an external validation. As before, I compare results of the baseline clustering algorithms with PDC (see Table 5.5). I compared the clustering results for two different choices of ground truth: (1) Time series are assumed to form 18 pairs, and (2) time series are assumed to form 14 pairs according to whether they came from the same collection. For example, given time series representing annual power demand of the Netherlands and Italy for two different months, the first ground truth assumes two groups, one for each country, while the latter ground truth assumes a single group representing the collection "power demand" in contrast to a substantially different collection, e.g., "EEG". Keogh, Lonardi, and Ratanamahatana (2004) reported a perfect clustering with a clustering based on the ZIP distance extended by a preprocessing of the data. The ZIP distance employed here performs at a relative low level compared to PDC. In comparison to the investigated baseline candidates, PDC performs best. Furthermore, the MinE heuristic automatically retrieved the embedding dimension that yields the best clustering accuracy.

	Embedding Dimension				
Method	3	4	5	6	7
Hellinger	66.67%	77.78%	**83.34%**	80.56%	69.45%
Symmetric KL	66.67%	77.78%	80.56%	77.78%	69.45%
Euclidean	61.11%	75%	77.78%	77.78%	69.45%
χ^2	66.67%	77.78%	61.11%	77.78%	72.23%

Table 5.4: 1NN-LOO Accuracy of combinations of embedding dimension and codebook distance measure for the ground truth "Sources" (see Table 5.5). The MinE heuristic determines an optimal embedding dimension of 5 which corresponds to the best clustering accuracy for the Hellinger distance, in boldface.

5.2.6 Clustering Natural Language Texts

Two further data sets were obtained from Keogh, Lonardi, and Ratanamahatana (2004). They consist of natural language texts in different languages. The authors showed that their compression-based clustering method can identify clusters of natural languages that closely resemble the hierarchical classification of languages from linguistics. In this subsection, I demonstrate that PDC

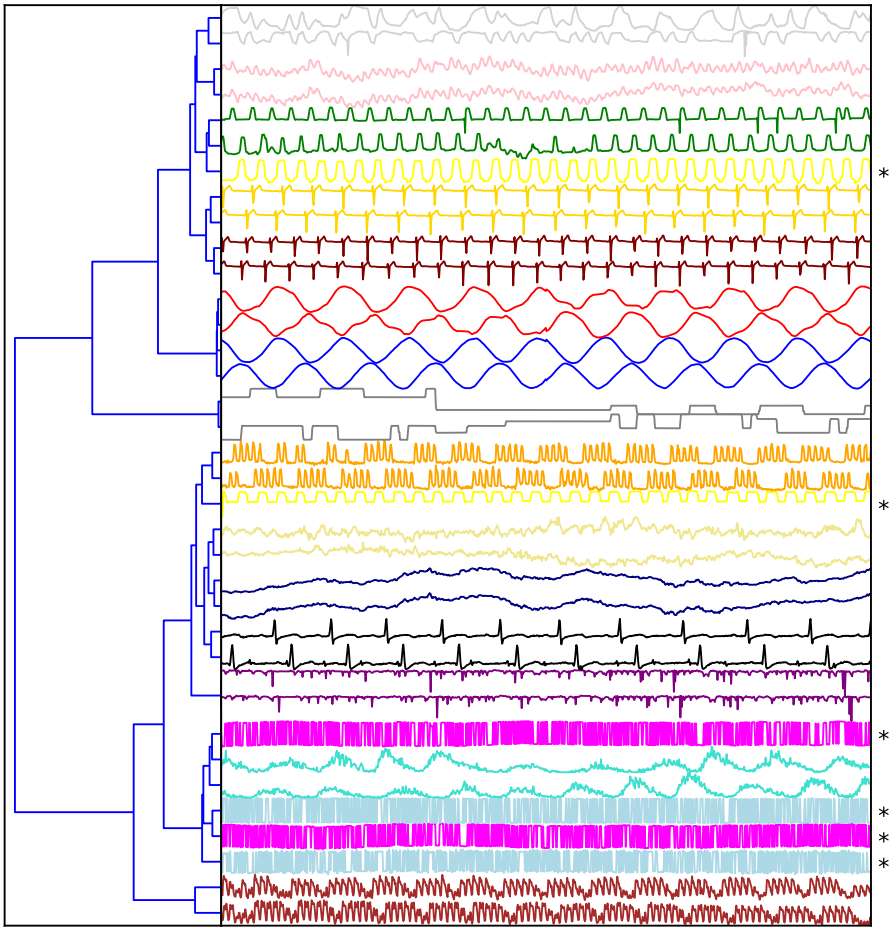


Figure 5.2.6: Eighteen assorted pairs of time series from the UCR Time Series Archive (Keogh & Folias, 2002) and a resulting dendrogram of a PDC analysis. Pairs of time series representing ground truth match in color. PDC retrieves a nearly perfect clustering of pairs on the lowest level of the clustering, i.e., the two elements of each pair are clustered together before being clustered together with any other time series. Three pairs of time series (indicated by stars) do not match their partner on the lowest level of the clustering hierarchy. The embedding dimension of this clustering is 5 chosen by the MinE criterion.

	Method						
Ground Truth	EUC	DTW	ZIP	PDC3	PDC4	PDC5 (MinE)	PDC6
Pairs	27.78%	44.45%	30.56%	47.22%	58.33%	72.22%	69.44%
Sources	41.67%	66.67%	47.23%	66.67%	77.78%	83.34%	80.56%

Table 5.5: Comparison of clustering methods on the KDD2004 data set for two different ground truths: "Pairs" states that the ground truth is given by 18 pairs corresponding to the pairs of time series in the original data set. "Sources" assumes a ground truth described by 14 qualitatively different sources that the time series were taken from; for example, one pair in the data set represents subsequences of the power demand over the year of the Netherlands, another pair represents the power demand of Italy. In "Pairs" they are assumed to be two separate pairs, whereas in "Sources", they are considered a single cluster.

can also achieve important results in this domain without any modification of my algorithm.

In order to be able to apply PDC to natural language texts, a conversion of the text to a numerical time series has to be performed. Any preprocessing step that potentially adds free parameters or assumes other prior knowledge would render this approach futile. Therefore, I simply transformed the string representation of the texts, character by character, into the ordinal number of the underlying character encoding. In this case, texts were available in an ISO-8859-1 representation (ISO, 1987), also known as *Latin 1*, which encodes 191 different characters in 8 bits per character (see Figure 5.2.7). This standard representation of text has interesting properties that help the clustering algorithm to obtain useful information about the language. The blank character encodes word boundaries. The Latin 1 representation of standard alphabet characters ranges between 65–90 (A–Z) and between 97–122 (a–z). The blank character is encoded with the number 32 and is thus represented by a smaller value than all standard Latin characters and those with accents or other special characters. Similarly, punctuation is encoded in the range between the blank and the Latin characters, e.g., the colon is at 44, the full stop at 46, and the exclamation mark at 33. Furthermore, accentuated characters and other special characters always have larger values than the standard alphabet characters.

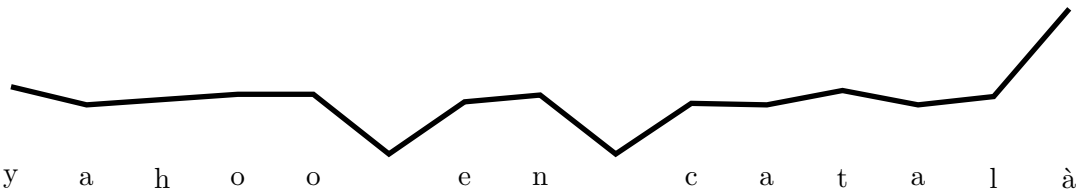


Figure 5.2.7: Ordinal representation of the string "yahoo en català" (Yahoo in Catalan) by replacing each character with its index of the character set Latin 1 (ISO-88590).

The first data set that was analyzed contains the first 50 chapters of Genesis from the Old Testament in the languages Dutch, German, Norwegian, Danish, Spanish, Italian, Latin, French, English, and Maori. The length of the time series was about 130,000. The minimum average entropy of the representations was achieved with an embedding dimension of 8 according to the MinE criterion (see Figure 5.2.1). The resulting clustering has a high face validity and is in alignment with linguistic classifications (cf. Keogh, Lonardi, & Ratanamahatana, 2004). The clustering is illustrated in Figure 5.2.8, showing two main clusters and one expected outlier. The first large cluster, colored in red, represents the Indo-European languages, with two sub-clusters, the Scandinavian cluster with Norse and Danish, and the West Germanic cluster with Dutch and German. The second large cluster (colored in green) represents the Romance languages with Spanish, Italian, Latin and French. In compliance with linguistic classifications, the Maori language, which belongs to the group of Malayo-Polynesian languages, is found to be a clear outlier.

A second data set contained the textual contents from various Yahoo portals in different languages. All hypertext markup tags were removed and all textual elements were pasted together into a single text. The resulting texts had a length of about 1,600 characters each. Based on the MinE heuristic (see Figure 5.2.1), an embedding dimension of 6 was chosen in this case. The clustering is depicted in Figure 5.2.8. Again, the two main clusters, which are united only at the top level of the clustering, represent the Germanic languages (green color) and the Romantic languages (red cluster). These clusters were retrieved as distinct clusters when using a heuristic based on the AIC. Using the BIC suggested a single cluster containing all languages. Within the Germanic cluster, a Scandinavian cluster with Norway, Denmark, and Sweden is found, and within the Romance cluster, the Spanish-speaking countries form a sub-cluster. PDC is able to reveal inherent structure in this data set as well as the approach of Keogh, Lonardi, and Ratanamahatana (2004).

5.2.7 Clustering DNA data

Keogh, Lonardi, and Ratanamahatana (2004) also presented a data set containing strings of DNA of twelve primate species and one non-primate outlier. They could successfully reconstruct a tree representing the ancestral hierarchy that is in line with today's theories of evolution. A similar successful endeavor based on compression distance was reported by M. Li et al. (2004). I applied PDC in the same way as before, without adding any preprocessing or transformation steps. I simply treated the mitochondrial DNA sequences of nuclide acid bases as strings consisting of the letters "a","c","g", and "t" with the relation $a < c < g < t$ implied by the ISO-8859-1 encoding. The time series had an average length of 16,700. According to the MinE heuristic (see Figure 5.2.1), an embedding dimension of 6 was chosen. The resulting hierarchical clustering is presented in Figure 5.2.9 (left). The corresponding ground truth according to today's scientific view on the evolutionary relations between primates can be found in Figure 5.2.9 (right).

The clustering is similar to the one obtained by the original authors with a single difference: the Capuchin and the Malayan Flying Lemur are falsely swapped in the PDC. Otherwise, the clustering closely matches the scientific classification. Both chimpanzee species form the subfamily *Ponginea*. Together with the Human, the Gorilla, the Ponginea, and both the Orang Utan species, they form the family of *Great Apes*. When adding the Gibbon, the cluster repre-

sents the superfamily of the *Hominoids*. By adding the *Old World monkeys*, Barbary Ape and Baboon, they form the Catarrhines. By further adding the Lemurs and the Capuchin, we obtain a cluster corresponding to the order of Primates. The Oyster is not a primate and as such, correctly classified as an outlier in the resulting clustering. The only divergence from biological taxonomy is that both Lemur types are thought to form a single category, the *Prosimians*, which is a division of Primates on the same level as the *Anthropoids*, which subsume the Catarrhines and the Capuchin.

The presented examples demonstrate the versatility of PDC on various data sets represented as strings rather than floating point numbers. PDC was not modified in any respect but simply be used with the default mapping from strings to integer numbers implied by the ISO-8859-1 standard.

The criteria to determine the number of clusters were not evaluated in this context since the relations of the time series are thought of as a hierarchical relation rather than clearly distinct clusters. Without such a ground truth, the determination of the number of clusters does not lead to information gain. However, it became apparent that PDC could successfully recover hierarchical structures inherent in the textual data sets.

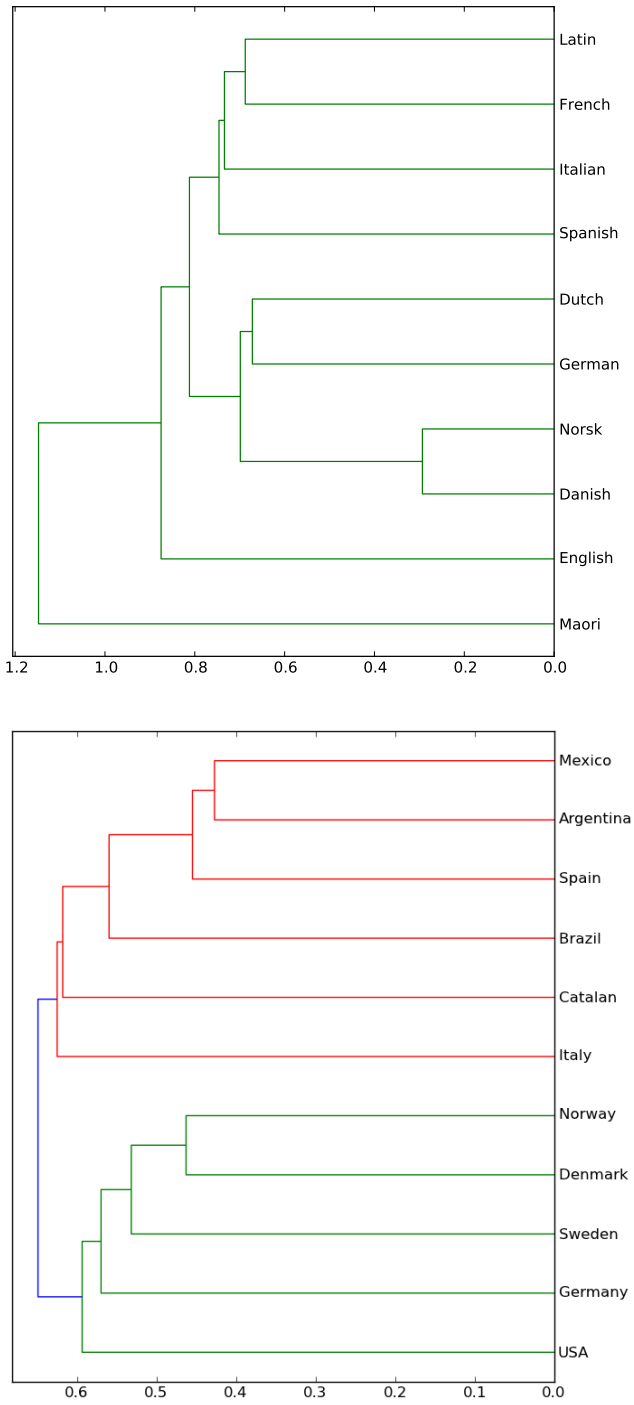


Figure 5.2.8: Permutation Distribution Clustering on text files. Text strings are regarded as ordinal time series, based on the ISO-8859-1 encoding of the characters. Top: The first fifty chapters of Genesis in different languages, resulting in ordinal time series of more than 130,000 observations. Clustering is based on an embedding dimension of 8. Bottom: Texts from various Yahoo portals, resulting in time series with more than 1,615 characters. The MinE criterion chose an embedding dimension of 6. Distinct clusters based on the AIC are depicted in same colors.

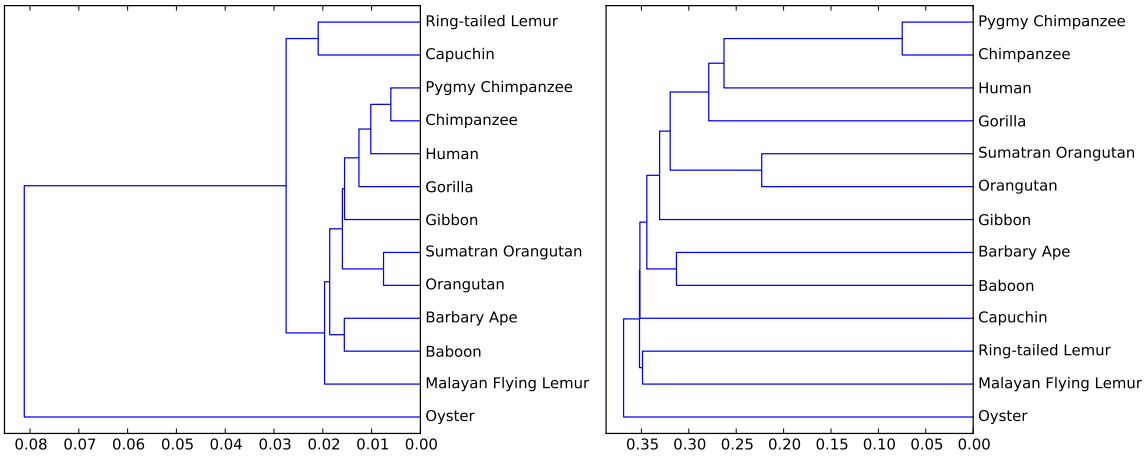


Figure 5.2.9: Clustering of mitochondrial DNA based on a simple character set encoding of the nucleic acid bases: a = 0, c = 1, g = 2, and t = 3. The series had an average length of about 16,700 characters. Left: Clustering resulting from PDC. The primates form a cluster and the only non-primate, the Oyster, is a clear outlier. Within the primates, the subgroups reflect the current classification system of primates, with the exception of the most distant relatives to the Human: the Lemurs, and the Capuchin. The embedding dimension was chosen to be 6 using the MinE criterion. Right: The clustering of the DNA data set obtained from the ZIP distance measure. Keogh, Lonardi, and Ratanamahatana (2004) reported having used a DNA-specific compression algorithm and found the resulting clustering to be in line with today's scientific view on the evolutionary relations of primates. Here, I replicated the reported hierarchical clustering based on the ZIP distance without employing a DNA-specific representation but the encoding as described in Subsection 5.2.1. The clustering is structurally identical to the clustering reported by Keogh, Lonardi, and Ratanamahatana (2004) and is considered as the ground truth for the task.

6 Conclusion

The primary goal of this thesis is the development of methodologies for the exploratory analysis of multivariate and multi-method data sets in psychological research. Two new methods to support researchers in informed theory formation were introduced. They are both built on tree structures to represent differences between observations in empirical data sets.

Permutation Distribution Clustering (PDC) is a method to enrich understanding about time series based on a measure of their mutual complexity. The typical situation for PDC is distinguished by the availability of a collection of univariate or multivariate time series with no or little additional information about the inherent structural differences. This renders choosing an explicit model for the time series a problem. Despite some difficulties in finding evaluative criteria, clustering is an acknowledged technique for gaining knowledge about data (Halkidi et al., 2001). Importantly, the very concept of hierarchical clustering was introduced by S. Johnson (1967) in the journal Psychometrika. In particular, clustering provides an important tool for the analysis of physiological time series, and is therefore relevant for psychophysiological studies that focus on how physiological responses relate to behavior (Andreassi, 2006). For example, Wismüller et al. (2002) and Golay et al. (1998) use clustering to understand brain responses measured with functional magnetic response imaging (fMRI).

PDC has especially interesting attributes. In a unified framework, it combines criteria for selecting representational complexity that determines the number of distinct order patterns a time series is represented by, as well as the number of distinct clusters. Furthermore, it has important inherent invariance properties and is efficient to compute, which makes it a powerful technique for clustering time series. We demonstrated the advantages and versatility of PDC by showing results on relevant data from motion prototypes in accelerometry, electrocardiography, and changes in α-power in electroencephalography. In all settings, PDC succeeded in finding inherent structure in the time series. PDC is a parameter-free technique that makes it possible to reveal commonality structures in sets of time series based on their complexity.

Another research situation is common: Hypotheses about the data exist and can be formalized in a model that determines relations between variables. During the model selection process,

researchers often modify models heuristically. Thereby, they are prone to overfit their models to the observed data, lose generality of their findings, and, in the worst case, are led to wrong conclusions. SEM Trees respond to the need for a formal model selection process for the integration of additional variables into theory-guided models. They provide an exploratory data analysis tool that harnesses the power from SEM and tree-structured methods. The graphical representation of covariate influences on SEM parameters in SEM Trees is clear and appealing.

The creation of SEM Trees is based on statistical tests. Different ways to account for the inherent multiple comparison problem by using Bonferroni-type corrections and cross-validation estimates were presented. We have seen that the split candidate selection criteria are inherently related to the information-theoretically motivated maximization of information gain and mutual information.

SEM Trees facilitate exploratory analyses for a large variety of models. In Section 5, applications of SEM Trees were shown on a factor model of adults' cognitive ability and on a longitudinal model of children's cognitive development. Particular attention was paid to factor-analytic SEM Trees. In order to find and interpret differences between individuals or groups in factor-analytic models, measurement invariance has to be ensured (Meredith, 1993), that is, additional restrictions of freely estimated parameters have to be established across groups. SEM Trees include specific mechanisms to guarantee measurement invariance, and can thus provide information whether the chosen flavor of invariance assumption holds across a data set for a given factor model. Moreover, finding significant differences in this framework without the invariance assumption is also possible, and enables the retrieval of different factor profiles for distinct sub-populations.

The range of models for SEM Trees is, of course, much larger. For example, functional connectivity in the human brain is an important topic of research. Mclntosh and Gonzalez-Lima (1994) already suggested using SEM to model functional connectivity between brain regions based on fMRI data. A method recently proposed by Gates and colleagues (Gates, Molenaar, Hillary, Ram, & Rovine, 2010; Gates, Molenaar, Hillary, & Slobounov, 2010) uses SEM, temporal embedded fMRI data and a greedy method that constructs connectivity networks in a SEM context. An application of SEM Trees or an combination of this modeling approach with SEM Trees is certainly highly important to gain knowledge about the relation between brain responses and behavior. For another example, developmental psychologists model interindividual differences in intraindividual change processes with latent difference score models (McArdle & Hamagami, 2001). SEM Trees can add a layer to the investigation of change by examining covariate-specific changes of change processes. A third method of SEM analysis that has recently gained popularity are Latent Differential Equations (Boker et al., 2004). These models allow the modeling of oscillatory or coupled oscillatory processes by approximating differential equations in a SEM framework. Such process types are highly interesting in social and cognitive psychology. For example, Chow, Ram, Boker, Fujita, and Clore (2005) modeled emotion regulation with a damped oscillator model. Within the same framework, Boker and Laurenceau (2007) modeled intimacy and autonomy in married couples as a coupled oscillatory process. SEM Trees can be applied to these models straightforwardly. With the advent of each new type of SEM and new data set that is analyzed with a traditional SEM technique, a new opportunity for an analysis with SEM Trees arises. Building on the synthesis of well-established tree-structured methods and the power of the SEM approach, SEM Trees provide a versatile, exploratory tool for detecting non-linear interactions of covariates on latent and observed parameters in SEM.

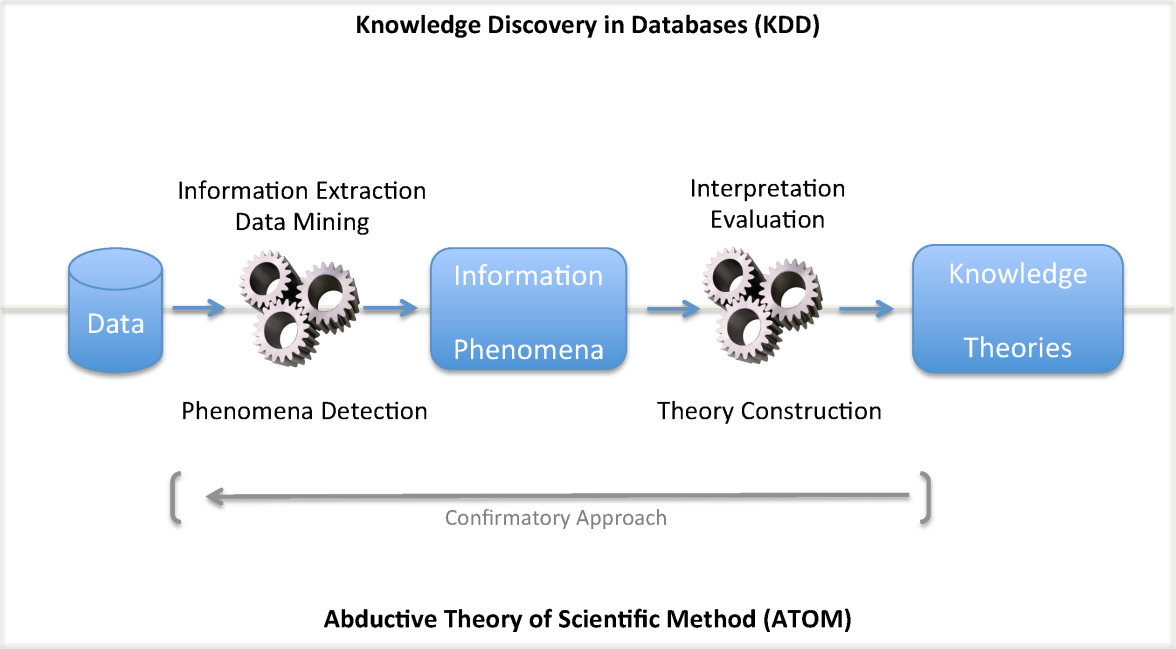


Figure 6.0.1: Schematic process of gaining knowledge and building theories. Both Knowledge Discovery in Databases (KDD; Fayyad et al., 1996b) and the Abductive Theory of Scientific Method (ATOM; Haig, 2005) propose a two-fold procedure to fulfill their goals. Both start at the raw data level. Using techniques such as data mining and inferential statistics, researchers reveal stable patterns in the data, referred to by computer scientists as information and by behavioral scientists as phenomena. A second step of interpretation and evaluation of the results in KDD parallels the step of theory construction in ATOM in order to acquire useful knowledge (KDD) or new theories (ATOM). The described procedure contrasts with the classical approach of hypothesis testing, which ideally constructs hypothesis ex nihilo, and accepts or rejects them based on observations.

By joining exploratory endeavors with confirmatory approaches, SEM Trees will be able to contribute to a vast array of fields beyond psychology, such as gerontology, life-course sociology, molecular genetics, behavior genetics, and behavioral neuroscience.

Finally, a promising outlook for SEM Trees is to allow a set of template models instead of a single model. This extension trades the disadvantage that parameters may no longer be comparable across the resulting partitions of the data set for a broader range of hypotheses that can be searched. For example, factor models with differing numbers of latent factors could serve as a set of template models. The corresponding multi-model SEM Tree would be able to retrieve different numbers of latent factors for different groups in the sample. Extending SEM Trees in this direction requires careful consideration of aspects of model selection and, from a Bayesian point of view, needs to account for prior probabilities of models.

Both PDC and SEM Trees proceed from different assumptions and from different stages of the data analysis process and are based on very different models of the data. Nevertheless, interesting commonalities emerge. Both methods are based on hierarchical structures that visualize a hierarchical partition of the data set. In SEM Trees this partition is model-based, that is, the hierarchy reflects significant differences in the model parameters for the resulting branches. Also, the partition is covariate-based, that is, only interpretable partitions are searched in the tree growing process, thereby allowing researchers to identify covariates that potentially help to explain differences in the data and refine their hypotheses. In the clustering approach, we are in a situation where there are neither a model about the data nor additional covariates. The permutation distribution of a multivariate time series provides a formalization of dissimilarity between time series. From this, a hierarchical clustering that visualizes a hierarchy of similarity can be constructed. For both methods, the likelihood ratio test can serve as a statistical tool to determine cut-off points in the tree structure. In SEM Trees the test is based on the likelihood ratio test for multivariate normal distribution. It can serve as a stopping rule that determines when tree growing should end, that is, when there is no more statistical evidence that any of the covariates explain model differences. When using cross-validation for split candidate selection, we found that the tree chooses those split candidates that are maximally statistically dependent on the model-predicted distribution. For PDC, the test is based on the likelihood ratio of multinomial distributions and can be employed to determine an appropriate number of clusters by successively cutting the hierarchy to separate clusters as long as significant differences between the multinomial distributions between the clusters are found.

Both tree-structured methods and clustering are traditionally attributed to the broad class of data mining methods (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a). The authors outline the process of knowledge discovery in databases (KDD) by claiming that the "KDD field is concerned with the development of methods and techniques for making sense of data." (Fayyad et al., 1996a, p. 37) and that "KDD aims to provide tools to automate (to the degree possible) the entire process of data analysis and the statistician's ´art´ of hypothesis selection" (Fayyad et al., 1996b, p. 29). In their process-oriented view on KDD, they describe data mining as a step in KDD that is concerned with the detection of interesting patterns. Patterns in that sense are any structured representations or descriptions of a data set, e.g., a model fitted to a data set or a test statistic. It is generally acknowledged that interestingness has aspects that are both objective or data-driven and subjective or user-driven (Freitas, 1999). Fayyad et al. (1996a) rephrase interestingness as a postulation of four properties of patterns that are required to enable knowledge gain: Patterns have to be valid, novel, potentially useful, and ultimately understandable. *Validity* demands findings that generalize from the observed sample to the population. A finding is *novel* if it was not known before, that is, trivial findings should be neglected since they do not support knowledge gain. *Usefulness* is usually defined with respect to a user or a task, which can be formalized by monetary savings or prediction accuracy. Lastly, *understandability* is a requirement regarding the accessibility of the findings. For example, take a research group that is investigating genetic influences on cognitive decline. If they used a neural network (cf. Bishop, 1995) that reliably predicted this decline, the result would be highly useful from a clinical point of view. However, the complexity of the neural network is typically a representation of knowledge which humans find difficult to understand and interpret. In contrast, tree structures like SEM Trees allow an understandable reading of the tree as rules that determine the partitions of the original data set. They can guide the researcher to interesting

differences in the data set that were not represented by the initial model and are thus novel. In the neurosciences, a need for meaningful inference was expressed by Brodersen et al. (2011) who use Dynamic Causal Modeling (Stephan et al., 2007) to model brain dynamics. Certainly, SEM Trees could provide a further approach to explore brain dynamics. Validity is reached by basing the tree constructions on statistical and information-theoretic criteria. SEM Trees are also in line with notions of interestingness that discern between user-centered and data-driven definitions of interestingness and usefulness. In a first stage, investigators select models and covariates that are potentially interesting to them, and in a second stage the algorithm retrieves the most interesting and useful model, defined by the choice of the splitting criterion.

In the same framework, clustering is a method that plays a central role in deriving knowledge from data and can serve as a tool for data reduction, hypothesis generation, hypothesis testing and prediction based on the resulting group structure (Halkidi et al., 2001). PDC can reveal structures when there is no information about the data, and therefore has the potential to detect novel findings. Since findings are based on statistical and information-theoretic tests, they can be considered valid. Again, the judgement of understandability is subjective. I argue that the permutation distribution is understandable as a distribution of order patterns with its important properties of invariance to shifts and scaling. Again, a tree structure is a graphical representation that allows the detection of grouping structures at a glance.

The process of data mining is most often a goal-oriented approach. The ultimate goal is maximizing the utility of the mined patterns, without regard to the interpretability of the data. If data miners found out that left-handed customers of a bank were less likely to pay back their loans then right-handed customers, they would happily advise the bank to increase interest rates for left-handed customers to account for that risk. In contrast, scientific theory building aims at finding insights into the processes that cause the phenomena. The abductive theory of scientific method (ATOM; Haig, 2005) describes a process of construction of exploratory theories for the behavioral sciences. The importance for generating exploratory strategies was also emphasized by Behrens and Yu (2003). Despite different goal formulations of KDD and ATOM, both methods are characterized by parallels in central aspects. Both propose a tri-partition of the process (see Figure 6.0.1). In a first step, stable patterns are extracted from the data. This is the heart of the exploratory process, and it can be achieved by the application of various methods, ranging from classical statistics to modern data mining methods. The resulting patterns are referred to as information (KDD) or phenomena (ATOM). In both methods, a second step is required to interpret those to acquire useful knowledge (KDD) or to construct valid theories (ATOM). SEM Trees and PDC contribute to both processes (illustrated by gear symbols in Figure 6.0.1), by extracting information/phenomena from the data, and provide statistics about the data that support their evaluation and interpretation, respectively the building of new theories. However, two warnings are important: (1) Neither SEM Trees nor PDC provide a shortcut from data to theories or from data to knowledge, they merely provide tools for the necessary processes in-between, and (2) as any new pattern from data mining must prove its utility on future data, e.g., by raising future sales of a business, each new theory must be evaluated on a new data set, at best, based on a custom-tailored paradigm to investigate the new hypotheses.

Oftentimes, researchers are interested in a single or, at most, a few hypotheses to be investigated. One can observe that data are painstakingly collected on a large scale, a set of hypotheses is tested, and finally the whole data set is backed up onto a remote hard drive, and

the researcher moves on to new research projects. I argue that exploratory analyses should have a stronger emphasis in the data analysis process. Traditionally, there has been a strong criticism against exploratory analyses in psychology, dating back to the hypothesis testing framework of Neyman and Pearson (1928). Critics tend to allege that exploratory results cannibalize chance. They claim that the exploitation of a large hypothesis space generates findings that are merely random fluctuations. Taking the same line, data mining is insulted as data dredging at times. In response, firstly, the problem of the generalizability of tree structures has been addressed in this thesis, and information-based and statistical criteria have been employed to assert the generalizability of the discovered patterns. Secondly, confirmatory and exploratory analyses are by no means exclusive. In his propositions of ethical data analysis, McArdle (2010) advocates simply performing exploratory analysis *after* the confirmatory analysis. Thereby, the validity of a-priori specified hypotheses is forfended while exploratory analyses can help to gain a better understanding of the data and ultimately extend our knowledge.

In a process of knowledge gain, information is extracted from the data and information is transformed to knowledge. In order to enable this process, methods are required to provide both a descriptive and predictive representation of the data (Fayyad et al., 1996b). Both proposed methods turn data into interesting information, based on statistical and information-theoretic criteria, and represent this information in tree structures, thereby enabling investigators to turn it into knowledge by refining their models about the data and, subsequently, their hypotheses and knowledge about the world. Permutation Distribution Clustering and Structural Equation Model Trees are specifically apt for exploratory data mining purposes in psychological research and beyond, and support researchers in informed theory development.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. Petrov & C. F. (Eds.), *Second International Symposium on Information Theory* (Vol. 1, pp. 267–281). Budapest.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Alcock, R., & Manolopoulos, Y. (1999). Time-series similarity queries employing a feature-based approach. In *7th Hellenic Conference on Informatics.* Ioannina, Greece.

Ali, S., & Silvey, S. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, *28*(1), 131–142.

Aloise, D., Deshpande, A., Hansen, P., & Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, *75*(2), 245–248.

Amari, S. (2007). Integration of stochastic models by minimizing α-divergence. *Neural Computation*, *19*(10), 2780–2796.

Amari, S., & Nagaoka, H. (2007). *Methods of information geometry.* Oxford, UK: Oxford University Press.

Andreassi, J. (2006). *Psychophysiology: Human behavior and physiological response.* Mahwah, NJ: Lawrence Erlbaum Associates.

Bandt, C., & Pompe, B. (2002). Permutation entropy: A natural complexity measure for time series. *Physical Review Letters*, *88*(17), 174102-1–174102-4.

Barry, R., Clarke, A., Johnstone, S., Magee, C., & Rushby, J. (2007). EEG differences between eyes-closed and eyes-open resting conditions. *Clinical Neurophysiology*, *118*(12), 2765–2773.

Behrens, J., & Yu, C. (2003). Exploratory data analysis. In J. A. Schinka & W. F. Velicer (Eds.), *Handbook of psychology* (Vol. 2, pp. 33–64). New York: Wiley.

Bentler, P. (1990). Fit indexes, Lagrange multipliers, constraint changes and incomplete data in structural models. *Multivariate Behavioral Research*, *25*(2), 163–172.

Bentler, P., & Bonett, D. (1980). Significance tests and goodness of fit in the analysis of covariance structures. *Psychological Bulletin*, *88*(3), 588–606.

Bergstrom, T. (2001). Free labor for costly journals? *Journal of Economic Perspectives*, *15*(4), 183–198.

Berndt, D., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In U. Fayyad & U. R. (Eds.), *AAAI94 Workshop on Knowledge Discovery in Databases* (pp. 359–370). Seattle, WA: AAAI Press.

Bishop, C. (1995). *Neural networks for pattern recognition.* Oxford, UK: Clarendon Press.

Bishop, C. (2006). *Pattern recognition and machine learning* (Vol. 4). New York: Springer.

Boker, S., & Laurenceau, J. (2007). Coupled dynamics and mutually adaptive context. In T. Little, J. Bovaird, & N. Card (Eds.), *Modeling contextual effects in longitudinal studies* (pp. 299–324). Mahwah, NJ: Lawrence Erlbaum Associates.

Boker, S., McArdle, J., & Neale, M. (2002). An algorithm for the hierarchical organization of path diagrams and calculation of components of expected covariance. *Structural Equation Modeling: A Multidisciplinary Journal*, *9*(2), 174–194.

Boker, S., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T., et al. (2011). OpenMx: An

open source extended structural equation modeling framework. *Psychometrika*, *76*(2), 306-317.

Boker, S., Neale, M., & Rausch, J. (2004). Latent differential equation modeling with multivariate multi-occasion indicators. In K. van Montfort, J. Oud, & A. Satorra (Eds.), *Recent developments on structural equation models: Theory and applications* (pp. 151–174). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Bollen, K. (1989). *Structural equations with latent variables*. Oxford, UK: John Wiley.

Bollen, K., Curran, P., & Wiley, J. (2006). *Latent curve models: A structural equation perspective*. Hoboken, NJ: John Wiley.

Bortz, J., & Schuster, C. (2010). *Statistik für Human- und Sozialwissenschaftler*. Berlin: Springer.

Box, G., Jenkins, G., & Reinsel, G. (1976). *Time series analysis: Forecasting and control*. San Francisco, CA: Holden Day.

Brandmaier, A. M. (2011). *SEM Trees: Recursive partitioning with Structural Equation Models.* Available from `http://www.brandmaier.de/semtree`

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

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International.

Brodersen, K. H., Haiss, F., Ong, C. S., Jung, F., Tittgemeyer, M., Buhmann, J. M., et al. (2011). Model-based feature construction for multivariate decoding. *NeuroImage*, *56*(2), 601 - 615. (Multivariate Decoding and Brain Reading)

Browne, M. (1974). Generalized least squares estimators in the analysis of covariance structures. *South African Statistical Journal*, *8*, 1–24.

Browne, M. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology*, *37*, 62–83.

Browne, M., & Cudeck, R. (1992). Alternative ways of assessing model fit. *Sociological Methods Research*, *21*(2), 230–258.

Burnham, K., & Anderson, D. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. New York: Springer.

Cao, Y., Tung, W.-w., Gao, J. B., Protopopescu, V. A., & Hively, L. M. (2004). Detecting dynamical changes in time series using the permutation entropy. *Physical Review E*, *70*(4), 046217-1–046217-7.

Caruana, R., Elhawary, M., Nguyen, N., & Smith, C. (2006). Meta clustering. In *Sixth IEEE International Conference on Data Mining*. Hong Kong, China: IEEE Computer Society.

Cattell, R. (1963). Theory of fluid and crystallized intelligence: A critical experiment. *Journal of Educational Psychology*, *54*(1), 1–22.

Cevhe, V., & Beirami, A. (2008). *Proofs of alpha divergence properties, stat 631 / elec 639: Graphical model.* Available from `http://www.ece.rice.edu/~vc3/elec633/proof_alpha_divergence.pdf`

Chan, K., & Fu, A. (1999). Efficient time series matching by wavelets. In *15th International Conference on Data Engineering* (pp. 126–133). Sydney, Australia.

Chernoff, H. (1952). A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, *23*(4), 493–507.

Chow, S., Ram, N., Boker, S., Fujita, F., & Clore, G. (2005). Emotion as a thermostat:

Representing emotion regulation using a damped oscillator model. *Emotion*, *5*(2), 208–225.

Cichocki, A., Cruces, S., & Amari, S. (2011). Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, *13*(1), 134–170.

Cover, T., & Thomas, J. (1991). *Elements of information theory* (Vol. 6). Hoboken, NJ: John Wiley.

Csizár, I. (1974). Information measures: A critical survey. In *Transactions of the Seventh Prague Conference on Information Theory, Statistical Decision Functions, Random Processes* (Vol. B, pp. 73–86). Dordrecht, The Netherlands: Reidel Publishing Company.

Cudeck, R., & Henly, S. (1991). Model selection in covariance structures analysis and the "problem" of sample size: A clarification. *Psychological Bulletin*, *109*(3), 512–519.

Dobra, A., & Gehrke, J. (2001). Bias correction in classification tree construction. In *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 90–97). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Dolan, C., Bechger, T., & Molenaar, P. (1999). Using structural equation modeling to fit models incorporating principal components. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(3), 233–261.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification* (2nd ed.). New York: John Wiley and Sons.

Eads, D. (2008). *hcluster: Hierarchical Clustering for SciPy.* Available from `http://scipy-cluster.googlecode.com/`

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26.

Efron, B., Tibshirani, R., & Tibshirani, R. (1993). *An introduction to the bootstrap.* New York: Chapman & Hall.

Eppstein, D. (1998). Fast hierarchical clustering and other applications of dynamic closet pairs. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 619–628). San Francisco, CA.

Esposito, F., Malerba, D., Semeraro, G., & Kay, J. (2002). A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(5), 476–491.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In R. T. Snodgrass & M. Winslett (Eds.), *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data* (Vol. 23, pp. 419–429). Minneapolis, MN: ACM.

Fan, X. (1997). Canonical correlation analysis and structural equation modeling: What do they have in common? *Structural Equation Modeling: A Multidisciplinary Journal*, *4*(1), 65–79.

Fan, X., Thompson, B., & Wang, L. (1999). Effects of sample size, estimation methods, and model specification on structural equation modeling fit indexes. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(1), 56–83.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37–54.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996b). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27–34.

Ferrer, E., Hamagami, F., & McArdle, J. (2004). Modeling latent growth curves with incomplete data using different types of structural equation modeling and multilevel software. *Structural Equation Modeling: A Multidisciplinary Journal*, *11*(3), 452–483.

Finkbeiner, C. (1979). Estimation for the multiple factor model when data are missing. *Psychometrika*, *44*(4), 409–420.

Fisch, B., & Spehlmann, R. (1999). *Fisch and Spehlmann's EEG primer: Basic principles of digital and analog EEG*. Amsterdam, The Netherlands: Elsevier Science.

Fisher, R. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A*, *222*(594-604), 309–368. Available from `doi:10.1098/rsta.1922.0009`

Fisher, R. (1925). Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, *22*(5), 700–725.

Fletcher, R. (1994). An overview of unconstrained optimization. In S. E. (Ed.), *Algorithms for continuous optimization: The state of the art* (pp. 109–143). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Formisano, E., De Martino, F., & Valente, G. (2008). Multivariate analysis of fMRI time series: Classification and regression of brain responses using machine learning. *Magnetic Resonance Imaging*, *26*(7), 921–934.

Fraley, C., & Raftery, A. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, *41*(8), 578–588.

Frank, B., Pompe, B., Schneider, U., & Hoyer, D. (2006). Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses. *Medical and Biological Engineering and Computing*, *44*(3), 179–187.

Freitas, A. (1999). On rule interestingness measures. *Knowledge-Based Systems*, *12*(5-6), 309–315.

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, *19*(1), 1–67.

Gailly, J., & Adler, M. (2004). *Zlib compression library.* Available from `http://zlib.net/`

Gale, W., & Church, K. (1994). What is wrong with adding one? In N. Oostdijk & P. de Haan (Eds.), *Corpus-based research into language: In honour of Jan Aarts* (pp. 189–198). Amsterdam, The Netherlands: Rodopi.

Gates, K., Molenaar, P., Hillary, F., Ram, N., & Rovine, M. (2010). Automatic search for fMRI connectivity mapping: An alternative to Granger causality testing using formal equivalences among SEM path modeling, VAR, and unified SEM. *Neuroimage*, *50*(3), 1118–1125.

Gates, K., Molenaar, P., Hillary, F., & Slobounov, S. (2010). Extended unified SEM approach for modeling event-related fMRI data. *NeuroImage*, *54*, 1151–1158.

Golay, X., Kollias, S., Stoll, G., Meier, D., Valavanis, A., & Boesiger, P. (1998). A new correlation-based fuzzy logic clustering algorithm for fMRI. *Magnetic Resonance in Medicine*, *40*(2), 249–260.

Goldberg, S. (1986). *Probability: An introduction.* New York: Dover Publications.

Gosper, R. (1978). Decision procedure for indefinite hypergeometric summation. *Proceedings of the National Academy of Sciences of the United States of America*, *75*(1), 40–42.

Haig, B. (2005). An abductive theory of scientific method. *Psychological Methods*, *10*(4), 371–388.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques.

Journal of Intelligent Information Systems, *17*(2), 107–145.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning.* Springer.

Haynes, J., & Rees, G. (2006). Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, *7*(7), 523–534.

Hero, A., Ma, B., Michel, O., & Gorman, J. (2001). *Alpha-divergence for classification, indexing and retrieval* (Tech. Rep. No. CSPL-328). University of Michigan; Communication and Signal Processing Laboratory.

Holden, K. (1995). Vector auto regression modeling and forecasting. *Journal of Forecasting*, *14*(3), 159–166.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, *6*(2), 65–70.

Horn, J., & McArdle, J. (1992). A practical and theoretical guide to measurement invariance in aging research. *Experimental Aging Research*, *18*(3), 117–144.

Hothorn, T., Hornik, K., Strobl, C., Zeileis, A., & Hothorn, M. (2011). *party: A laboratory for recursive partytioning.* Available from `http://cran.r-project.org/web/packages/party/index.html`

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674.

Hu, L., & Bentler, P. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(1), 1–55.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314.

ISO. (1987). *ISO/IEC 8859-1:1998, 8-bit single byte coded graphic character set - Latin alphabet No. 1.* International Organization for Standardization.

Jeffreys, H. (1948). *Theory of probability.* Oxford: Clarendon Press.

Jensen, D., & Cohen, P. (2000). Multiple comparisons in induction algorithms. *Machine Learning*, *38*(3), 309–338.

Johnson, D. H., & Sinanovic, S. (2001). Symmetrizing the Kullback-Leibler distance. *IEEE Transactions On Information Theory*, *1*(1), 1–10.

Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.

Jöreskog, K. G. (1969). A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika*, *34*(2), 183–202.

Jöreskog, K. G. (1970). Estimation and testing of simplex models. *British Journal of Mathematical and Statistical Psychology*, *23*(2), 121–145.

Jöreskog, K. G. (1971). Simultaneous factor analysis in several populations. *Psychometrika*, *36*(4), 409–426.

Jöreskog, K. G. (1979). Statistical models and methods for analysis of longitudinal data. In K. Jöreskog, D. Sörbom, & M. J. (Eds.), *Advances in factor analysis and structural equation models* (pp. 129–169). Cambridge, MA: Abt Books.

Jöreskog, K. G., & Sörbom, D. (1982). Recent developments in structural equation modeling. *Journal of Marketing Research*, *19*(4), 404–416.

Jöreskog, K. G., & Sörbom, D. (1996). LISREL 8 user's reference guide [Computer software manual]. Scientific Software.

Kailath, T. (1967). The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, *15*(1), 52–60.

Kalpakis, K., Gada, D., & Puttagunta, V. (2001). Distance measures for effective clustering of ARIMA time-series. In N. Cercone, T. Lin, & X. Wu (Eds.), *Proceedings of the 2001 IEEE International Conference on Data Mining* (pp. 273–280). San Jose, CA.

Kenny, S., Andric, M., Boker, S. M., Neale, M. C., Wilde, M., & Small, S. L. (2009). Parallel workflows for data-driven structural equation modeling in functional neuroimaging. *Frontiers in Neuroinformatics*, *3*(34), 1–11.

Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2001). Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of ACM SIGMOD Conference on Management of Data* (Vol. 30, pp. 151–162). ACM.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2004). Segmenting time series: A survey and novel approach. In M. Last, A. Kandel, & B. H. (Eds.), *Data mining in time series databases.* Singapore: World Scientific Publishing.

Keogh, E., & Folias, T. (2002). The UCR time series data mining archive [http://www. cs. ucr. edu/˜ eamonn/tsdma/index. html]. riverside ca. *University of California, Computer Science & Engineering Department*.

Keogh, E., & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, *7*(4), 349–371.

Keogh, E., Lonardi, S., & Ratanamahatana, C. (2004). Towards parameter-free data mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 206–215). ACM.

Kim, H., & Loh, W. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, *96*(454), 589–604.

Kleiber, C., & Zeileis, A. (2008). *Applied econometrics with R.* New York: Springer.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. Mellish (Ed.), *International Joint Conference on Artificial Intelligence* (Vol. 14, pp. 1137–1145). Los Altos, CA: Morgan Kaufmann.

Kotter-Grühn, D., Wiest, M., Zurek, P., & Scheibe, S. (2009). What is it we are longing for? Psychological and demographic factors influencing the contents of Sehnsucht (life longings). *Journal of Research in Personality*, *43*(3), 428–437.

Kriegeskorte, N., Simmons, W., Bellgowan, P., & Baker, C. (2009). Circular analysis in systems neuroscience: The dangers of double dipping. *Nature Neuroscience*, *12*(5), 535–540.

Kuha, J. (2004). AIC and BIC. *Sociological Methods & Research*, *33*(2), 188–229.

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 79–86.

Landwehr, N., Hall, M., & Frank, E. (2005). Logistic model trees. *Machine Learning*, *59*(1), 161–205.

Lawley, D. (1956). A general method for approximating to the distribution of likelihood ratio criteria. *Biometrika*, *43*(3/4), 295–303.

Lee, S. (2007). *Structural equation modeling: A Bayesian approach.* New York: John Wiley.

Lee, S., & Song, X. (2003). Maximum likelihood estimation and model comparison for mixtures of structural equation models with ignorable missing data. *Journal of Classification*, *20*(2), 221–255.

Leek, J., & Storey, J. (2008). A general framework for multiple testing dependence. *Proceedings*

of the National Academy of Sciences of the United States of America, *105*(48), 18718–18723.

Lehmer, D. (1960). Teaching combinatorial tricks to a computer. In *Proceedings of Symposia in Applied Mathematics, Combinatorial Analysis* (Vol. 10, pp. 179–193). New York.

Lewis, M. (2004). *Moneyball: The art of winning an unfair game*. New York: W. W. Norton & Company.

Li, M., Chen, X., Li, X., Ma, B., & Vitányi, P. (2004). The similarity metric. *IEEE Transactions on Information Theory*, *50*(12), 3250–3264.

Li, M., & Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*. New York: Springer.

Li, X., Ouyang, G., & Richards, D. (2007). Predictability analysis of absence seizures with permutation entropy. *Epilepsy Research*, *77*(1), 70–74.

Liao, T. (2005). Clustering of time series data: A survey. *Pattern Recognition*, *38*(11), 1857–1874.

Lindenberger, U., Li, S., Lövdén, M., & Schmiedek, F. (2007). The Center for Lifespan Psychology at the Max Planck Institute for Human Development: Overview of conceptual agenda and illustration of research activities. *International Journal of Psychology*, *42*(4), 229–242.

Lindenberger, U., Smith, J., Mayer, K., & Baltes, P. (Eds.). (2010). *Die Berliner Altersstudie*. Berlin: Akademie Verlag.

Little, T., Lindenberger, U., & Nesselroade, J. (1999). On selecting indicators for multivariate measurement and modeling with latent variables: When "good" indicators are bad and "bad" indicators are good. *Psychological Methods*, *4*, 192–211.

Loh, W. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, *12*(2), 361–386.

Lunetta, K., Hayward, L., Segal, J., & Van Eerdewegh, P. (2004). Screening large-scale association study data: Exploiting interactions using random forests. *BMC Genetics*, *5*(1), 32.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

MacCallum, R., & Austin, J. (2000). Applications of structural equation modeling in psychological research. *Annual Review of Psychology*, *51*(1), 201–226.

MacCallum, R., Browne, M., & Sugawara, H. (1996). Power analysis and determination of sample size for covariance structure modeling. *Psychological Methods*, *1*(2), 130–149.

MacCallum, R., Roznowski, M., & Necowitz, L. (1992). Model modifications in covariance structure analysis: The problem of capitalization on chance. *Psychological Bulletin*, *111*(3), 490–504.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In M. L. Le Cam & J. Neyman (Eds.), *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). Berkeley, CA: University of California Press.

McArdle, J. (1988). Dynamic but structural equation modeling of repeated measures data. In J. Nesselroade & R. Cattell (Eds.), *Handbook of multivariate experimental psychology* (Vol. 2, pp. 561–614). New York: Plenum Press.

McArdle, J. (1994). Structural factor analysis experiments with incomplete data. *Multivariate*

Behavioral Research, *29*(4), 409–454.

McArdle, J. (2010). Some ethical issues in factor analysis. In A. Panter & S. Sterber (Eds.), *Quantitative methodology viewed through an ethical lens* (pp. 313–339). Washington, DC: American Psychological Association Press.

McArdle, J., & Aber, M. (1990). Patterns of change within latent variable structural equation modeling. In A. von Eye (Ed.), *New statistical methods in developmental research* (pp. 151–224). New York: Academic Press.

McArdle, J., & Boker, S. (1990). *RAMpath: A computer program for automatic path diagrams.* Hillsdale, NJ: Lawrence Erlbaum Associates.

McArdle, J., & Epstein, D. (1987). Latent growth curves within developmental structural equation models. *Child Development*, *58*(1), 110–133.

McArdle, J., Ferrer-Caja, E., Hamagami, F., & Woodcock, R. (2002). Comparative longitudinal structural analyses of the growth and decline of multiple intellectual abilities over the life span. *Developmental Psychology*, *38*(1), 115–142.

McArdle, J., & Hamagami, F. (2001). Latent difference score structural models for linear dynamic analyses with incomplete longitudinal data: New methods for the analysis of change. In L. M. Collins & A. G. Sayer (Eds.), *New methods for the analysis of change.* Washington, DC: American Psychological Association.

McArdle, J., & McDonald, R. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology*, *37*(2), 234–251.

McArdle, J., & Prescott, C. (1992). Age-based construct validation using structural equation modeling. *Experimental Aging Research*, *18*(3), 87–115.

Mclntosh, A., & Gonzalez-Lima, F. (1994). Structural equation modeling and its application to network analysis in functional brain imaging. *Human Brain Mapping*, *2*(1-2), 2–22.

Meredith, W. (1964). Notes on factorial invariance. *Psychometrika*, *29*(2), 177–185.

Meredith, W. (1993). Measurement invariance, factor analysis and factorial invariance. *Psychometrika*, *58*(4), 525–543.

Meredith, W., & Tisak, J. (1990). Latent curve analysis. *Psychometrika*, *55*(1), 107–122.

Minka, T. (2005). *Divergence measures and message passing* (Tech. Rep. No. MSR-TR-2005-173). Cambridge, UK: Microsoft Research.

Mosier, C. (1951). The need and means of cross validation: I. Problems and designs of cross-validation. *Educational and Psychological Measurement*, *11*(1), 5–11.

Muthen, B., & Curran, P. (1997). General longitudinal modeling of individual differences in experimental designs: A latent variable framework for analysis and power estimation. *Psychological Methods*, *2*(4), 371–402.

Muthen, L., & Muthen, B. (2007). Mplus: Statistical analysis with latent variables: User's guide [Computer software manual]. Los Angeles, CA: Muthen and Muthen.

Neale, M., Boker, S., Xie, G., & Maes, H. (1999). Mx: Statistical modeling [Computer software manual]. Richmond, VA.

Neyman, J., & Pearson, E. (1928). On the use and interpretation of certain test criteria for purposes of statistical inference part I. *Biometrika*, *20*(1-2), 175.

Nielsen, F. (2011). *Chernoff information of exponential families.* (Available as Arxiv preprint arXiv:1102.2684)

von Oertzen, T., Ghisletta, P., & Lindenberger, U. (2009). Simulating statistical power in

latent growth curve modeling: A strategy for evaluating age-based changes in cognitive resources. In M. Crocker & J. Siekmann (Eds.), *Resource-adaptive cognitive processes* (pp. 95–117). Heidelberg: Springer.

OpenMx Project. (2011). *OpenMx: Advanced structural equation modeling.* Retrieved 23.07.2011, from `http://openmx.psyc.virginia.edu`

Osborne, R., & Suddick, D. (1972). A Longitudinal Investigation of the Intellectual Differentiation Hypothesis. *Journal of Genetic Psychology*, *121*(pt 1), 83–9.

Perneger, T. (1998). What's wrong with Bonferroni adjustments. *British Medical Journal*, *316*(7139), 1236.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*, *27*(3), 221–234.

Quinlan, J. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence* (pp. 343–348). Hobart, Australia.

Quinlan, J. (1993). *C4. 5: Programs for machine learning.* San Francisco, CA: Morgan Kaufmann.

Quinlan, J. (1996). Improved use of continuous attributes in C4. 5. *Journal of Artificial Intelligence Research*, *4*, 77–90.

R Development Core Team. (2011). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Available from `http://www.R-project.org`

Rao, C. (1995). A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance. *Questiió: Quaderns d'Estadística i Investigació Operativa*, *19*(1).

Roweis, S., & Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural Computation*, *11*(2), 305–345.

Rubin, D. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Russel, S., & Norvig, P. (2003). *Artificial intelligence: A modem approach* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Salvador, S., & Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with AI* (pp. 576–584). IEEE Computer Society.

Sanchez, G., & Aluja, T. (2010). pathmox: Segmentation trees in partial least squares path modeling [Computer software manual]. Available from `http://CRAN.R-project.org/package=pathmox` (R package version 0.1)

SAS. (1999). SAS: Statistical package version 8e for Windows [Computer software manual]. Cary, NC.

Schafer, J., & Graham, J. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*(2), 147–177.

Schmiedek, F., Lövdén, M., & Lindenberger, U. (2009). On the relation of mean reaction time and intraindividual reaction time variability. *Psychology and Aging*, *24*(4), 841–857.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464.

Segal, M. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, *87*(418), 407–418.

Shannon, C. (1951). Prediction and entropy of printed English. *Bell System Technical Journal*, *30*(1), 50–64.

Shillabeer, A., & Roddick, J. (2006). Towards role based hypothesis evaluation for health data mining. *Electronic Journal of Health Informatics*, *1*(1), e6.

Sonquist, J., & Morgan, J. (1964). *The detection of interaction effects. A report on a computer program for the selection of optimal combinations of explanatory variables* (No. 35). Ann Arbor, MI: Survey Research Centre, The Institute for Social Research, University of Michigan.

Sörbom, D., & Jöreskog, K. (1982). The use of structural equation models in evaluation research. In C. Fornell (Ed.), *A second generation of multivariate analysis: Vol. 2. Measurement and evaluation* (pp. 381–418). New York: Praeger.

Spearman, C. (1904). "General intelligence," objectively determined and measured. *The American Journal of Psychology*, *15*, 201–293.

Steiger, J., & Lind, J. (1980). *Statistically based tests for the number of common factors.* Paper presented at the Annual Meeting of the Psychometric Society, Iowa City, IA.

Stephan, K., Harrison, L., Kiebel, S., David, O., Penny, W., & Friston, K. (2007). Dynamic causal models of neural system dynamics: Current state and future extensions. *Journal of Biosciences*, *32*(1), 129–144.

Stirling, J., & Tweddle, I. (2003). *James Stirling's Methodus Differentialis: An annotated translation of Stirling's text.* London, UK: Springer.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *36*(2), 111–147.

Strobl, C., Boulesteix, A., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25.

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application and characteristics of classification and regression trees, bagging and random forests. *Psychological Methods*, *14*(4), 323–348.

Strobl, C., Wickelmaier, F., & Zeileis, A. (2011). Accounting for individual differences in bradley-terry models by means of recursive partitioning. *Journal of Educational and Behavioral Statistics*, *36*(2), 135.

Tanaka, J. (1987). "How big is big enough?": Sample size and goodness of fit in structural equation models with latent variables. *Child Development*, *58*(1), 134–146.

Thurstone, L. (1947). *Multiple-factor analysis.* Chicago, IL: University of Chicago Press.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411–423.

Tucker, L., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika*, *38*(1), 1–10.

Velicer, W. F., & Jackson, D. N. (1990). Component analysis versus common factor analysis: Some issues in selecting an appropriate procedure. *Multivariate Behavioral Research*, *25*(1), 1-28.

Wang, X., Smith, K., Hyndman, R., & Alahakoon, D. (2004). *A scalable method for time series clustering* (Vol. 1; Tech. Rep.). Monash University, Australia.

Wechsler, D. (1949). *Wechsler Intelligence Scale for Children: Manual.* New York: Psychological

Corporation.

Widaman, K. (1993). Common factor analysis versus principal component analysis: Differential bias in representing model parameters? *Multivariate Behavioral Research*, *28*(3), 263–311.

Wilks, S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, *9*(1), 60–62.

Wismüller, A., Lange, O., Dersch, D., Leinsinger, G., Hahn, K., Pütz, B., et al. (2002). Cluster analysis of biomedical image time-series. *International Journal of Computer Vision*, *46*(2), 103–128.

Witten, I., & Bell, T. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, *37*(4), 1085–1094.

Wright, S. (1934). The method of path coefficients. *Annals of Mathematical Statistics*, *5*(3), 161–215.

Wrzus, C., Brandmaier, A. M., von Oertzen, T., Müller, V., Wagner, G. G., & Riediger, M. (2011). *Monitoring sleep in its natural context: Validation of an ambulatory accelerometry approach.* (Manuscript submitted for publication)

Zeileis, A., Hothorn, T., & Hornik, K. (2006). *Evaluating Model-based Trees in Practice* (Tech. Rep. No. 32). Vienna, Austria: Department of Statistics and Mathematics, WU Vienna University of Economics and Business.

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, *17*(2), 492–514.

Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, *23*(3), 337–343.

Acronyms

AIC Akaike Information Criterion
ALT Auto-regressive Latent Trajectory
AR Auto-Regression or auto-regressive
ATOM Abductive Theory Of scientific Method
BIC Bayesian Information Criterion
CFI Comparative Fit Index
CV Cross-Validation
DTW Dynamic Time Warping
ECG Electrocardiography
EEG Electroencephalography
FA Factor Analysis
FIML Full Information Maximum Likelihood
fMRI functional Magnetic Resonance Imaging
KDD Knowledge Discovery in Databases
KL Kullback-Leibler divergence
LGCM Latent Growth Curve Model
LL Log-Likelihood
LLR Log-Likelihood Ratio
LOO-1NN Leave-One-Out One-Nearest-Neighbor
MI Measurement Invariance
ML Maximum Likelihood
MCAR Missing Completely At Random
MAR Missing At Random
NMAR Not Missing At Random
NNFI Non-Normed Fit Index
OOB Out Of Bag
PCA Principal Components Analysis
PD Permutation Distribution
PDC Permutation Distribution Clustering
RMSE Root Mean-Squared Error
RMSEA RMSE of Approximation
SEM Structural Equation Model
SRMR Standardized Root Mean Square Residual
WAIS-R Wechsler Adult Intelligence Scale Revised
WISC Wechsler Intelligence Scale for Children

A

Path Tracing Rules and Covariance Algebra

Given a graphical model representation of a SEM or a representation of equations relating its variables to each other, the model-implied covariance matrix can be derived. In a graphical representation, variances and covariances can be derived by a set of path-tracing rules (Wright, 1934). Given equations, the usage of covariance algebra allows this computation. In the following, both approaches are shown.

I repeat some rules for covariance algebra. Let X, Y be normally distributed random variables and a, b constants, then the following rules follow from the definition of covariance

$$\begin{aligned} \operatorname{Cov}(aX, bY) &= a \cdot b \cdot \operatorname{Cov}(X, Y) \\ \operatorname{Cov}(X, a) &= 0 \\ \operatorname{Cov}(X + a, Y + b) &= \operatorname{Cov}(X, Y) \\ \operatorname{Cov}(X, X) &= \operatorname{Var}(X) \end{aligned}$$

Using these basic rules, covariance terms of linearly combined random variables can be worked out step by step. Alternatively, path-tracing rules, as introduced by Wright (1934), provide an equivalent to determine covariances from a graphical SEM representation. First, a distinction between asymmetric and symmetric paths in the graph is required. *Asymmetric* paths are paths of zero or more one-headed arrows. *Symmetric* paths are paths of exactly one single two-headed arrow. In order to determine the covariance between two variables, find all asymmetric paths that lead into each variable. For each pair of paths with sources connected by an edge, collect all regression weights on the path and the covariance from the symmetric path and multiply them. The resulting total covariance is obtained as the sum of all products found in the described way. The sum of all incoming paths that are combined by a covariance enters the sum twice if the paths are distinct, and only once if they are equal. This case will be further explained in the second example below. Note that the set of all starting paths can be infinite if there are cyclic asymmetric paths. If the resulting sum does not converge, the model should be deemed invalid.

Otherwise, the sum can be evaluated in the limit and the resulting term for the path can be obtained.

In the following, both ways to obtain the covariance in a SEM are explained on the basis of two examples. The first is an exemplary pseudo-factor-analytic structure with latent and observed variables and no measurement errors. Figure A.1 shows a graphical representation and corresponding Structural Equations. Using path-tracing rules, the covariance between X and Y can be calculated as follows. There are four paths into X: the zero-length-path, the path from B, and two paths from A and C via B. The incoming paths to Y are d and the zero-length path. Only the paths bc and d are connected by a symmetric path, the covariance between C and D. Therefore, the total covariance is the term $Cov(X, Y) = b \cdot c \cdot d \cdot \delta$.

Using covariance algebra, we can also deduce the covariance analytically. Remember that the asymmetric paths describe regression weights, thus describing linear relations of variables. Therefore, we can deduce the following relations from the asymmetric paths

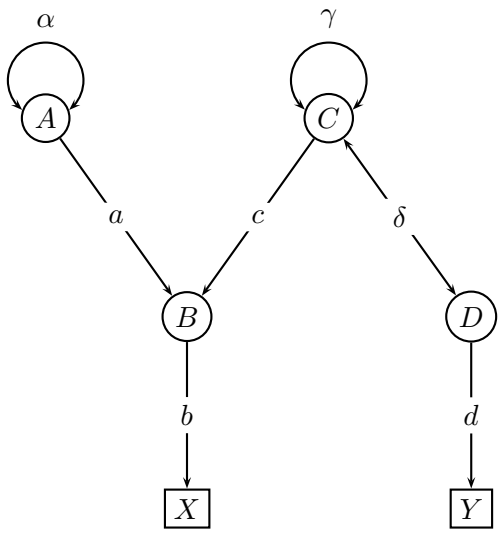
$$\begin{aligned} X &= bB \\ Y &= dD \\ B &= aA + cC \end{aligned}$$

This allows us to use the above rules of covariance algebra to determine covariances between the variables. For example, the covariance between X and Y is determined as shown in the following

$$\begin{aligned} Cov(X, Y) &= Cov(bB, dD) = bd \cdot Cov(B, D) = \\ &= bd \cdot Cov(aA + cC, D) = bd\,[a \cdot Cov(A, D) + c \cdot Cov(C, D)] \\ &= bda \cdot Cov(C, D) = bda\delta \end{aligned}$$

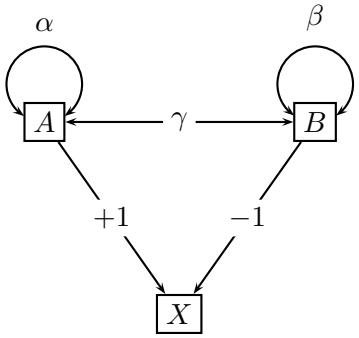
As a second example, we define a difference score X that is defined to be the difference of two random variables A and B with unique variances α and β and a covariance γ. The algebraic definition and a graphical SEM representation are given in Figure A.2. The variance of X can be worked out by path-tracing rules as follows. We imagine a copy of X and all its incoming asymmetric paths, named X'. A total number of three asymmetric paths lead into each variable X and X', the positive path, the negative path and the zero-length path. We obtain four product terms. The symmetric path α connects X with X' by using the positive paths, resulting in the term $1 \cdot \alpha \cdot 1$, likewise β connects X with X' by the negative paths resulting in the term $-1 \cdot \beta \cdot -1$. The last two terms are the symmetric path γ connecting the positive and the negative path and the negative with the positive path, resulting in twice the term $-1 \cdot \gamma \cdot 1$. The total variance yields $Var(X) = \alpha + \beta - 2\gamma$. The same result is derived with covariance algebra

$$\begin{aligned} Var(X) &= Cov(X, X) = Cov(A - B, A - B) \\ &= Cov(A, A) + Cov(A, -B) + Cov(-B, A) + Cov(B, B) \\ &= \alpha - 2Cov(A, B) + \beta \\ &= \alpha - 2\gamma + \beta \end{aligned}$$



- $X = b \cdot B$
- $Y = d \cdot D$
- $B = a \cdot A + c \cdot C$
- $Var(A) = \alpha$
- $Var(C) = \gamma$
- $Cov(C, D) = \delta$

Figure A.1: Graphical representation and Structural Equations of a model with latent variables A, B, C, D, and observed variables X, Y, and some paths, variances, and a covariance.



- $X = A - B$
- $Cov(A, B) = \gamma$
- $Var(A) = \alpha$
- $Var(B) = \beta$

Figure A.2: Graphical representation and Structural Equations of a difference score. The score X is defined as the difference between two random variables A and B with unique variances α and β and a covariance of γ.