John C. Dunn and Michael L. Kalish

State-Trace Analysis

September 6, 2017

Springer

"What is the smallest number of simple undefined things at the start, and the smallest number of undemonstrated premises, out of which you can define the things that need to be defined and prove the things that need to be proved? That problem, in any case that you like to take, is by no means a simple one, but on the contrary an extremely difficult one. It is one that requires a great amount of logical technique."

Bertrand Russell (1956, p. 271)

Acknowledgements

This short book on state-trace analysis is the product of a long period of gestation that has benefited greatly from the thoughtful input of many colleagues. In the first instance, we wish to acknowledge the ground-breaking conceptualizations of psychological measurement developed by Don Bamber and Geoffrey Loftus that led to the initial formulation and development of state-trace analysis. We also wish to acknowledge the original contributions by Kim Kirsner and Andrew Heathcote to the concepts discussed in these pages. And we also would like to thank our other colleagues; Ben Newell, Brett Hayes, Emily Freeman, Greig de Zubicaray, Laura Anderson, Luke Finlay, Oleg Burdakov, Oleg Sysoev, Rachel Stephens, Ralph James, Rik Henson, and Simon Dennis, whose input has been critical in getting us to where we are today.

We also wish to acknowledge the generous support of the Australian Research Council and National Science Foundation who have enabled us to conduct our work and through it, the present development of state-trace analysis.

Preface

Human behavior is complex. One of the aims of experimental psychology is to account for this complexity by developing and testing relatively simple theoretical models. These models are simpler than the behavior they describe because they typically consist of a small number of theoretical constructs that combine according to a set of rules, sometimes in the form of mathematical equations, to predict behavior. A desirable model is one that accounts for the behavior in question using the fewest number of theoretical constructs.

Theories of cognition posit a range of hypothetical constructs, often described in terms of unseen mental processes or processing systems, to account for aspects of observed behavior. This has been a feature of the 'cognitivist' approach to psychological research and appeared relatively early – a well-known example being Atkinson and Shiffrin's division of memory into 'short term' and 'long term' stores. However, such proposals have not been without controversy and have often led to longstanding debates concerning the exact number of constructs that might be required to explain behavior in a particular domain. For example, researchers currently debate the number of components underlying recognition memory (one or two, or perhaps more), the number of processes underlying human reasoning (again, one or two), working memory (three or more), and the number of systems underlying perceptual category learning (again, one or two). This small sample is known to us from our own work – there are likely to be many similar debates in other fields.

Perhaps the most basic modelling question is whether observed changes in two or more, ostensibly distinct, aspects of behavior demand one or more than one theoretical construct. Yet, despite its centrality, the tools researchers have used to address this question are often *ad hoc*. Traditionally, there has been strong reliance on methods based on functional dissociation – a logic originally borrowed from neuropsychology – which have been applied in various ways, including incorporation into the statistical workhorse of psychological research, the analysis of variance. Unfortunately for the great bulk of research that has relied on these methods, they are not fit for purpose

as they are unable to support the inferential burden placed upon them. In contrast, *state-trace analysis* (STA) offers a rigorous procedure for achieving this goal that is firmly grounded in mathematics and logic. For this reason, we believe it is timely to present a comprehensive overview of STA that will provide researchers for the first time with the means to appreciate and to apply this methodology.

STA supersedes all previous methods for identifying the number of theoretical constructs or latent variables (as we will refer to them) that mediate the effects of one or more independent variables on two or more dependent variables. It does so because it is firmly based on a rigorous and explicit logic. Our intention in this short book is to make the argument for STA, clarifying what it can and cannot do, and showing that it is the best solution to some of the irreducible difficulties of psychological measurement. Although the reader will find the occasional mathematical equation, these are presented in the service of clarity rather than obfuscation. Our aim is to present the basic ideas and techniques of STA in a form that allows the user to apply them immediately (or as quickly as possible after a little study). Our book is therefore aimed at graduate students and our fellow researchers in cognitive psychology and related fields. It presupposes familiarity with appropriate research methodology and statistics and some relevant mathematical concepts.

Our book (like Caesar's Gaul) is divided into three main parts. In Part I, we focus on the logical underpinnings of STA. In Chapter 1 we outline a general framework for thinking about the relationships between independent variables, latent variables, and dependent variables. In Chapter 2, we explore the implications of the critical assumption of monotonicity in measurement, including measurement in psychology, and its relevance to STA. In Chapter 3, we demonstrate how methods based on functional dissociation fail and how they are superseded by STA.

In Part II, we turn to practical application of STA. In Chapter 4 we show how to fit and to test the single latent variable model that forms the heart of STA. In Chapter 5 we describe a software package that can be used to fit state-trace models and to test them within a frequentist perspective. In Chapter 6 we apply a modified version of this software to binary data and in Chapter 7 we explore some additional designs. Throughout, we give numerical and computational examples and show how to perform STA using our publicly-available software in both Matlab and R.

In Part III, we discuss some further topics. In Chapter 8 we examine Bayesian approaches to the analysis of STA designs and in Chapter 9 we discuss potential extensions of STA and address some additional aspects of its application.

Perth, Australia — *John C. Dunn*
Syracuse, USA — *Michael L. Kalish*
September, 2017

Glossary

Condition An observable feature of the physical world corresponding to a unique combination of levels of one or more *independent variables*.

Conjoint monotonic regression *Monotonic regression* of two or more dependent variables on to a best-fitting common *total order*.

Degeneracy A dependent variable is said to be degenerate if it has exactly the same value under each *condition* of an *experiment*. See also *near degeneracy*.

Dependent variable An observable feature of the physical world that is measured as part of an *experiment*.

Dependent variable (DV) space The space defined by the Cartesian product of all possible values of a set of dependent variables.

Differential influence Consider an *input mapping* that maps two *independent variables*, a and b to *two latent variables*, u and v. If the ratio of the effects of a and b on u is not equal to the ratio of the effects of a and b on v then u and v are said to be differentially influenced (or affected) by a and b. Similar considerations apply to an *output mapping* that maps u and v to *dependent variables*, x and y. Differential influence implies that the relevant mapping is *functionally independent*.

Dimensionality A set of points in an n-dimensional space has dimension $m \leq n$ if it is the product of an m-to-n *mapping*. For example, a *state-trace* in two-dimensional *DV space* has dimension 1 if it is the product of an *output mapping* from a one-dimensional *LV space*.

Directed acyclic graph A finite graph in which vertices are connected by a directed edge such that there are no cycles (i.e., it is not possible to start at one vertex and, by following the directed edges, eventually return to this vertex). A direct acyclic graph can be used to represent a *partial order*.

Experiment A procedure by which the levels of one or more independent variables are varied in order to measure their effects on one or more dependent variables.

Face inversion effect The greater effect of inversion (at study or test) on memory for pictures of faces compared to pictures of other mono-oriented stimuli such as houses.

Functional dependence A multivariate *mapping* is functionally dependent if one or more of its component functions are functions of the remaining component functions.

Functional dissociation Two *dependent variables*, x and y, are said to be functionally dissociated if a change in an *independent variable* affects x but not y. Two kinds of dissociation are often distinguished: A ***single dissociation*** corresponds to this definition; a ***double dissociation*** consists of two opposite single dissociations. That is, in addition to the single dissociation defined above, there exists a second independent variable that affects the y but not x.

Functional independence A multivariate *mapping* is functionally independent if none of its component functions is a function of the remaining component functions. Functional independence implies *differential influence*.

General structure A theoretical framework in which a set of points in *independent variable space*, corresponding to the *conditions* of an *experiment*, are mapped by an *input mapping* to a set of points in *latent variable space* which are, in turn, mapped to a set of points in *dependent variable space* called the *state-trace*.

Independent variable An observable feature of the physical world that is varied across a set of levels as part of an experiment.

Independent variable (IV) space The space defined by the Cartesian product of all possible levels of a set of independent variables.

Input mapping A hypothetical multivariate function that maps points in *independent variable space* corresponding to the conditions of an experiment to points in *latent variable space*.

Jacobian matrix Consider a mapping, f, with component functions, $f_1(x)$, $\ldots$, $f_n(x)$, for $x = (x_1, \ldots, x_m)$. The Jacobian matrix of f at x is the $n \times m$ matrix of partial derivatives of each component function $f_i(x)$ with respect to each x_j.

Latent variable An unobserved variable that is postulated by a theory to mediate the effects of one or more *independent variables* on one or more *dependent variables*.

Latent variable (LV) space The space defined by the Cartesian product of all possible values of a set of *latent variables*.

Linear model A *model* in which the *input* and/or *output mappings* consist entirely of linear functions.

Mapping A multivariate function consisting of a set of component functions. For example, an *output mapping* from *LV space* to a two-dimensional DV space will consist of two component functions that take as input the values of each *latent variable* and produce as output a value for each *dependent variable*.

Model Specification of an input mapping or an output mapping to predict points in *latent variable space* or *dependent variable space*, respectively. Most often a model specifies an output mapping and is used to predict the observed *state-trace*.

Monotonically increasing function . A function, $f(x)$ is monotonically increasing if, for all $x < \tilde{x}$, $f(x) \leq f(\tilde{x})$.

Monotonic function See *monotonically increasing function*.

Monotonic mapping A multivariate one-to-many mapping in which each component function is *monotonic*.

Monotonic model A *model* in which the *input* and/or *output mapping* is *monotonic*.

Monotonic regression Given a vector, y, a set of weights, w, and a partial order, E, monotonic regression finds the vector, y^*, consistent with E that is maximally close to y in a weighted least squares sense.

Near degeneracy A *dependent variable* that is not significantly different across a set of *conditions* of an *experiment* is said to be near degenerate.

Nomic measurement The 'problem of nomic measurement' is the problem that arises when a dependent variable, x, is a function, g, of an unknown latent variable, u. The problem is that g cannot be discovered empirically (i.e., observed) without knowing the values of both u and x, but u is unknown.

Observational structure The observed part of the *general structure* consisting of the levels of the *independent variables* and the values of *dependent variables*.

Output mapping A hypothetical multivariate function that maps points in *latent variable space* corresponding to the conditions of an experiment to points in *dependent variable space*.

Partial order A binary relation on a set that that formalizes the intuition of an order or sequence. A partial order can be represented by a *directed*

acyclic graph, $G(V, E)$, where V is a set of vertices and E is a set of edges. A vector, x, is said to be consistent with E if $x_i \leq x_j$ for all $(i, j) \in E$.

Russell's teapot The subject of an argument developed by Bertrand Russell that a claim need not be accepted simply because it has not been falsified. This is illustrated with reference to the assertion that there is a teapot in orbit around the sun between Earth and Mars. Russell argues that there is no onus to believe this claim simply because it has not been shown to be false. More generally, Russell's teapot is a version of Occam's razor – if all known facts are consistent with theory A and theory $A + B$ (where B consists of one or more additional claims) then theory A is preferred.

Semantic properties of a model In the context of a theory, the meaning ascribed to the *latent variables* of a *model*.

Signed difference analysis A generalization of *state-trace analysis* to fit and test *models* having more than one *latent variable*.

State-trace The set of points in *dependent variable space* corresponding to the set of conditions in an experiment. Three kinds of state-trace can be distinguished. The ***observed state-trace*** is the set of observed values (usually means) on each *dependent variable* for each *condition* of the *experiment*. The ***model state-trace*** is the corresponding set of predicted values derived from a *model*. The ***true state-trace*** is the corresponding set of true values.

State-trace analysis A procedure to determine whether one or more than one latent variables mediate the effects of one or more independent variables on two or more dependent variables.

Syntactic properties of a model The formal properties of a *model* determined by its structural characteristics, such as the number of *latent variables* and their functional relationships.

Theoretical structure The theoretical (unobserved) part of the *general structure* consisting of the *input mapping*, the values of *latent variables* and the *output mapping*.

Total order A *partial order* in which every pair of elements is comparable. If a vector, x, is consistent with a total order then either $x_i \leq x_j$ or $x_j \leq x_i$ for all i, j.

Contents

Part I
Theory

Chapter 1
The logic of state-trace analysis

1.1 Introduction

State-trace analysis (STA) was introduced by Bamber (1979) as a general method to determine properties of the *latent structure* that mediates the effects of one or more independent variables on two or more dependent variables. This issue frequently arises in psychological research where the latent structure is commonly interpreted in terms of distinct cognitive mechanisms, processes, or processing systems. An example of this is a study by Kahana and Wingfield (2000) which examined the effects of varying the number of learning trials and the age of participants on both the amount and semantic organisation of recalled material. The theoretical question was whether these independent variables differentially affected one or more than one latent variable (or cognitive mechanism) that affected the two dependent variables. Similar questions also arise in current recognition memory research where there is debate whether memory depends on one or two component processes (e.g., Batchelder & Riefer, 1990; DeCarlo, 2002; Parks & Yonelinas, 2007; Wixted & Mickes, 2010; Yonelinas, 1994) as well as research on free recall (Brainerd & Reyna, 2010), perceptual category learning (Ashby, Alfonso-Reese, Turken & Waldron, 1998), visual perception (Goodale & Milner, 1992), reasoning (Evans, 2003; Heit & Rotello, 2010), and reading aloud (Coltheart, Rastle, Perry, Langdon & Ziegler, 2001).

STA offers a principled way of adjudicating between theories proposing different numbers of latent variables under very general conditions and with minimal assumptions. In fact, we will argue that STA provides the method of choice to address this issue. Yet in spite of this, since its inception in 1979 there have been relatively few studies that have applied STA to address these and related questions (for a comprehensive list, see Prince, Brown & Heathcote, 2012a). We believe that there are two main reasons for this. First, although there have been several previous attempts to describe the essentials of STA, it still remains relatively unfamiliar to most researchers. In part,

this has been due to the fact that when STA has been discussed, it has usually been in the context of a particular experimental investigation which has tended to limit its scope of application and its audience (e.g., Busey, Tunnicliff, Loftus & Loftus, 2000; Dunn, 2008; Jang & Nelson, 2005; Loftus & Irwin, 1998; Loftus, Oberg & Dillon, 2004; Newell & Dunn, 2008; Pratte & Rouder, 2012; Prince et al., 2012a). The second reason for the relative obscurity of STA is a practical one – standard statistical tests are of limited applicability to STA and, until recently, no customized statistical procedure has been available. This has had the effect of confining the application of STA to domains in which measurement error is small enough to be ignored or for which no formal statistical tests are required.

Our conceptualization of STA draws on two, largely independent, lines of development. The first of these is associated with the work of Geoffrey Loftus and colleagues (e.g., Loftus & Bamber, 1990; Loftus & Irwin, 1998; Loftus et al., 2004) and is based on the original description of STA by Bamber (1979) and an earlier paper by Loftus (1978) that emphasized the critical assumption of monotonicity in psychological measurement. Both of these papers highlighted the distinction between latent variables (e.g., memory strength, degree of learning, visual acuity) that are the targets of psychological theory and manifest or dependent variables (e.g., proportion correct, hit rates, response time) that are presumed to measure them in some way. Loftus (1978) has argued that, at our present stage of knowledge, all that can be reasonably assumed about the relationship between a latent variable and a dependent variable is that it is monotonically increasing or decreasing. For example, in many contexts, we can assume that the relationship between memory strength (a latent variable) and proportion correct on a recall test (a dependent variable) is monotonically increasing. This means that, all else being equal, if memory strength increases between two experimental conditions then proportion correct cannot decrease. More recently, Loftus et al. (2004) explored the implications of this observation in relation to STA, and proposed an approach to psychological measurement, called *dimensional theory*, which they contrasted with the alternative *linear theory* – the widespread application of analysis of variance and other variants of the general linear model. We discuss these ideas more fully in Chapters 2 and 3.

The second line of development is associated with the work of the present authors and our colleagues (e.g., Dunn & James, 2003; Dunn & Kirsner, 1988; Dunn, 2008; Kalish, Dunn, Burdakov & Sysoev, 2016; Newell & Dunn, 2008; Newell, Dunn & Kalish, 2011), and draws on an analysis of the logic of inference from functional dissociation in light of Loftus's (1978) argument for monotonicity. The conclusion from this analysis is that functional dissociation is neither necessary nor sufficient to infer the existence of more than one latent variable. Although this work initially developed independently of STA and dimensional theory, the underlying concepts proved to be very similar. It is the synthesis of these two lines of development that we present in this volume.

In the remainder of this chapter, we outline the logic of STA. In Section 1.2, we introduce the concept of the *general structure* which provides a framework for representing the functional relationships between sets of independent, latent, and dependent variables. Within this framework, we distinguish between the *observational structure* that consists of the physical components that can be observed and the *theoretical structure* that consists of the theoretical components that must be inferred. In Section 1.3, we discuss the *dimensionality of the state-trace* and show how it can be used to infer properties of the theoretical structure. In particular, we derive the central idea of STA that the dimensionality of the theoretical structure – the number of latent variables – cannot be less than the dimensionality of the state-trace.

1.2 The general structure

The general structure is an abstract framework that represents functional relationships between a set of independent variables $(a, b, c, \ldots)$, a set of latent variables $(u, v, w, \ldots)$ and a set of dependent variables $(x, y, z, \ldots)$. It is illustrated schematically in Figure 1.1.

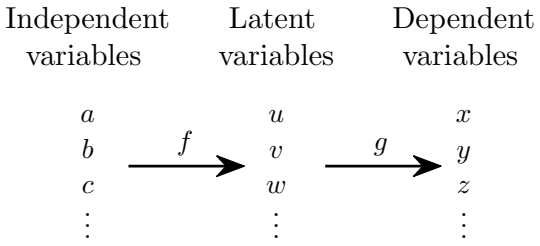

Fig. 1.1 The general structure assumed by state-trace analysis. A set of independent variables, $(a, b, c, \ldots)$, is mapped onto a set of latent variables, $(u, v, w, \ldots)$, by an input mapping, f. The set of latent variables is then mapped onto a set of dependent variables, $(x, y, z, \ldots)$, by an output mapping, g.

Of the three kinds of variable, both independent and dependent variables are observable aspects of the physical world, while latent variables are theoretical constructs that remain forever hidden and whose properties, and very existence, must be inferred. The general structure is thus composed of two parts; an observational structure consisting of the set of independent variables and the set of dependent variables and their observed relationships, and a theoretical structure consisting of the set of latent variables and their functional relationships to the set of independent on the one hand and the set of dependent variables on the other (Bamber, 1979). The goal of STA is to infer properties of the theoretical structure from properties of the observational structure.

In the theoretical structure, the relationship between a set of independent variables and a set of latent variables is governed by a multivariate mapping, f, called the *input mapping*, in which each latent variable is a function of the set of (potentially all possible) independent variables. Thus, we can write, $u = f_u(a, b, c, \dots)$, $v = f_v(a, b, c, \dots)$, and so on, where $f_u, f_v \dots$ are distinct multivariate functions. The relationship between a set of latent variables and a set of dependent variables is also governed by a multivariate mapping, g, called the *output mapping*, in which each dependent variable is a function of the set of (potentially all possible) latent variables. Thus, we can write, $x = g_x(u, v, ...)$, $y = g_y(u, v, ...)$, and so on, where $g_x, g_y \dots$ are potentially distinct functions[1].

The functional relationships contained within the theoretical structure are intended to be comprehensive and to encompass many kinds of psychological theory. For example, one kind of psychological theory specifies only an output mapping. Signal detection theory (SDT; Swets, Tanner & Birdsall, 1961) is an example. In the simplest case, two dependent variables are measured; the hit rate, HR, defined as the probability of correctly detecting a target when it is present and the false alarm rate, FAR, defined as the probability of incorrectly detecting a target when it is absent. Each of these dependent variables is a function of two latent variables, called sensitivity or discriminability, (d'), and a decision criterion, (c), according to the following output mapping (Macmillan & Creelman, 2005):

$$\begin{aligned} FAR &= \Phi(-c) \\ HR &= \Phi(d' - c) \end{aligned} \tag{1.1}$$

where $\Phi(.)$ is the normal cumulative distribution function.

The general structure can be represented geometrically as a set of mappings between different spaces. In any given experiment, a particular observational structure is selected. This determines the set of independent variables that are manipulated and the set of dependent variables that are observed. This is illustrated in Figure 1.2 which shows the functional relationships between sets of points in three different spaces; an *independent variable (IV) space*, defined as the space of all possible values of the (selected) set of independent variables, a *latent variable (LV) space*, defined as the space of all possible values of a set of (proposed) latent variables, and a *dependent variable (DV) space*, defined as the space of all possible values of the set of (observed) dependent variables. In the present example, each space in Figure 1.2 is two-dimensional. Accordingly, the IV space is defined in terms of two independent variables, labelled a and b; the LV space is defined in terms of two latent variables, u and v; and the DV space is defined in terms of two dependent variables, x and y.

In an experiment, as well as selecting a set of independent variables, particular values or levels of these variables are also selected. The Cartesian product

[1] We generally assume that these functions are continuous and differentiable.

(or conjunction) of these values define the *conditions* of the experiment each of which corresponds to a point in IV space. This point is then mapped by the input function to a corresponding point in LV space which, in turn, is mapped by the output function to a point in DV space. The set of observed points in DV space is called the *state-trace* of the experiment. The observational structure consists of the points in IV space, the corresponding points in DV space, and the observed relationships between them. These are observable by virtue of generally agreed measurement operations that determine values of the independent variables, a and b, for each point in IV space and the values of the dependent variables, x and y, for each point in DV space. The theoretical structure consists of the input mapping f, the set of points in LV space, and the output mapping g. Neither f, the LV space (and the points within it), nor g is directly observable.

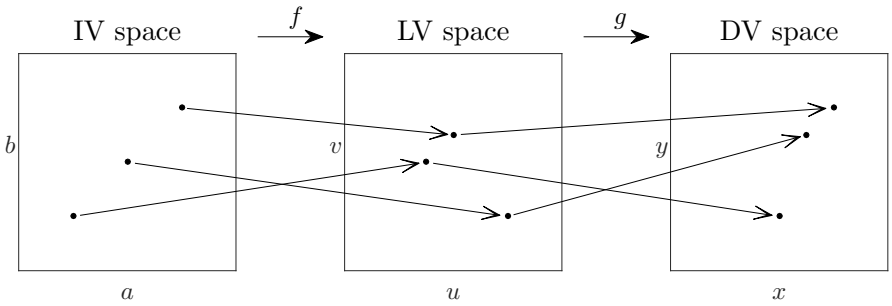

Fig. 1.2 Geometrical interpretation of the general structure for a particular experiment. The independent variable (IV) space is defined as the set of possible levels of two independent variables, a and b. Each realized point in this space corresponds to a *condition* of the experiment. These points are mapped by the input mapping, f, to a set of points in latent variable (LV) space, defined as the set of possible values of two latent variables, u and v. The set of points in LV space is, in turn, mapped by an output mapping, g, to a set of points in dependent variable (DV) space, defined as the set of possible values of two dependent variables, x and y. The set of points in DV space is the *state-trace* of the experiment.

Figure 1.3 is a concrete illustration of the general structure based on the results of Experiments 2 and 5 from Verde and Rotello (2007). In these experiments (here combined into a single virtual experiment), participants were given a list of words to remember. Half the words were *weak* and half were *strong*. Weak words were presented only once while strong words were presented four times throughout the study list. Recognition memory for studied words was then tested on two separate blocks of trials. In the first block, only strong words were tested while in the second block, only weak words were tested. Feedback was also manipulated (in the original study, between experiments 2 and 5). In the feedback condition, after each response

to a test word, participants were told if they were correct or incorrect[2]. In the no feedback condition, no indication of performance was provided.

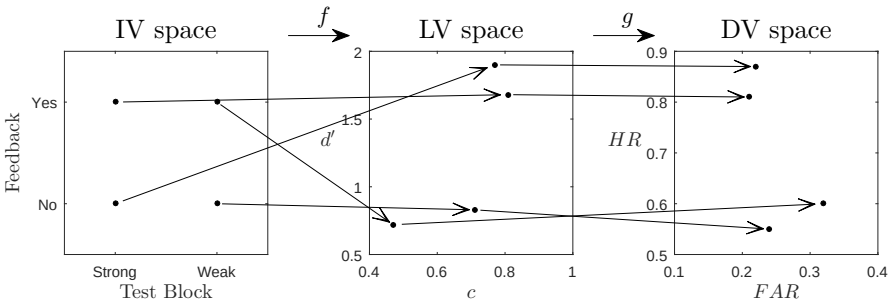

Fig. 1.3 The general structure corresponding to the combination of Experiments 2 and 5 from Verde & Rotello (2007). The independent variables are the memory strength of test items (weak vs. strong) and response feedback (no vs. yes). The latent variables are d' and c, based on the output mapping, g, defined by Equation (1.1). The dependent variables are false alarm rate (FAR) and hit rate (HR). In this instance, the input mapping, f, is not formally specified.

The IV space and the DV space shown in Figure 1.3 form part of the observational structure of this experiment. The independent variables are Test Block with two levels – Strong and Weak, and Feedback with two levels – No and Yes. Each combination of a level of Test Block and a level of Feedback defines a condition of the experiment. The dependent variables are the average hit rates (HR) and false alarm rates (FAR) as reported by Verde and Rotello (2007)[3]. The theoretical structure corresponds to the LV space and associated input and output mappings, f and g. In order to determine the points in this space (and the space itself) we applied the output mapping defined by Equation (1.1). More properly, the set of points in LV space was determined by applying the inverse of this output mapping to the set of points in DV space. The input mapping is undefined although, given the points in LV space, an ad hoc function could be constructed. It is important to note that, in this example, the LV space and the points within it depend entirely on the choice of output mapping. A different output mapping (such as the high-threshold model (Laming, 1973) would generate a different LV space[4].

When fully specified, the theoretical structure provides a complete account, or *model*, of the observational structure, specifying the relationship

[2] They were also given summary information of their total number of correct and incorrect responses after each quarter of the test had been completed.

[3] Verde and Rotello (2007) also reported standard errors for these estimates. These are not show in Figure 1.3.

[4] The high-threshold output mapping is; $FAR = g$, $HR = p(1-p)g$, where p is the probability of correctly detecting that a test item was studied and g is the probability of guessing that an undetected item was studied.

between a set of points in IV space and the corresponding set of points in DV space. However, rarely does an actual psychological model specify both input and output mappings (although for a comprehensive attempt at such a description, see Loftus et al., 2004). More typically, a model specifies only one part of this structure – usually the output mapping. In this case, the data (observed values on the dependent variables) are properly said to be *postdicted* rather than *predicted* as it is only after the values of the dependent variable(s) are known that the values of the (theory dependent) latent variables can be determined. If, on the other hand, a model also proposes an input mapping then, based on the set of conditions in the IV space, the values of the dependent variables can be formally predicted. In our view, this remains largely an aspirational goal in psychological science although some models of memory, such as REM (Shiffrin & Steyvers, 1997), have made significant progress in this respect.

1.3 The dimensionality of the state-trace

The aim of STA is to test models of the observational structure, or 'simple theories of causation' (Bamber, 1979), based on the dimensionality of the state-trace. Recall that the state-trace is the observed set of points in DV space produced by applying an output mapping to a set of points in the LV space. If the DV space is defined by n dependent variables, x, y, z, ..., then it is said to be n-dimensional. Similarly, LV space may be defined by m latent variables, u, v, w, ..., in which case it is said to be m-dimensional. The dimensionality of LV space need not be the same as the dimensionality of DV space.

An output mapping from an m-dimensional LV space to an n-dimensional DV space bestows an intrinsic dimensionality to the state-trace. Specifically, the state-trace may have a lower intrinsic dimensionality than the dimensionality of DV space. A familiar example of this property is a surface (e.g., of a sheet of paper) that exists in three spatial dimensions. Although each point on the surface is defined on each of three spatial dimensions, the surface itself is two-dimensional. This means that it can be defined as a mapping from a two-dimensional space to a three-dimensional space. Similarly, a curve in three spatial dimensions (e.g., that traced out by a piece of string) is one-dimensional. Because the state-trace is defined as the product of an output mapping applied to points in LV space, the dimensionality of the state-trace cannot exceed the dimensionality of LV space. That is,

$$\dim(\text{state-trace}) \leq \dim(\text{LV space}) \tag{1.2}$$

Equation (1.2) forms the inferential basis of state-trace analysis. Specifically, if a theory proposes that the LV space has dimensionality m but the

state-trace has a dimensionality greater than m then this theory is refuted. In practice, STA is applied to theories which assert that $m = 1$. The remainder of this book is concerned with this case although we return to the question of testing theories for which $m > 1$ in Chapter 9.

Example

In this section we illustrate the concept of the dimensionality of the state-trace with a concrete example based on the SDT model given in Equation 1.1. We apply this model to a hypothetical recognition memory experiment in which a set of items is presented for study and, following a retention interval, the same items are presented for test mixed with additional items that had not been studied. Participants are asked to discriminate between studied items (or targets) and non-studied items (or lures).

We suppose that two independent variables are manipulated; k, the number of times a target item is presented for study (either, 1, 2, or 3 times) and p, the proportion of test items that are targets (either, 0.25, 0.5, or 0.75). These variables define the IV space shown in the left panel of Figure 1.4. Each point in this space corresponds to a unique combination of the levels of each independent variable and, because these variables are combined factorially, the set of points forms a grid.

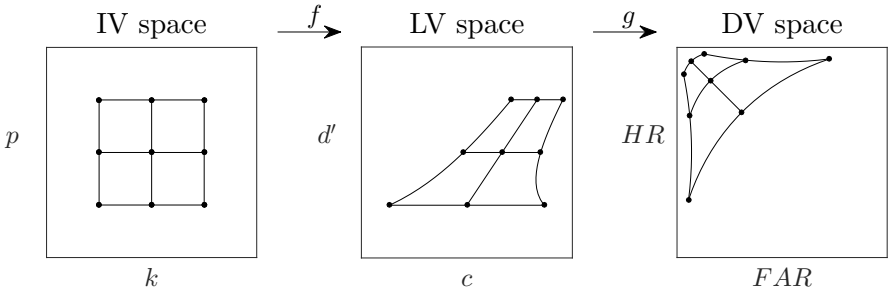

Fig. 1.4 Geometrical interpretation of the general structure for a recognition memory experiment. The IV space is defined by two independent variables; number of study presentations (k) and probability of a target at test (p). The LV space, defined by the signal detection model in Equation (1.1), consists of two latent variables, sensitivity (d') and decision criterion (c). The DV space is defined by the dependent variables of false alarm rate (FAR) and hit rate (HR).

The signal detection model given by Equation (1.1) defines an LV space composed of two latent variables: d', the discriminability of the targets and lures, and c, the level of evidence required to assert that a test item is a target. The input mapping, f, defines the relationship between points in IV

space and points in LV space. For the purposes of this example, we assume that f is defined as follows:

$$f : \begin{cases} d' = k \\ c = k/2 + \ln\left(\dfrac{1-p}{p}\right)/k \end{cases} \tag{1.3}$$

Equation (1.3) states that d' is equal to k, the number of study presentations, and that c is a function of both d' and p, the proportion of targets. According to this function, the decision criterion is placed at a point on the memory continuum representing the memory strength of a test item where the likelihood that it is a target is equal to the likelihood that it is a lure. The middle panel of Figure 1.4 shows how Equation (1.3) maps the set points in IV space to a set of points in LV space. The output mapping, g, defined by Equation 1.1 takes the set of points in LV space and maps them to a corresponding set of points (the state-trace) in DV space.

The state-trace in Figure 1.4 is two-dimensional because the output mapping is composed of functions of two latent variables. Furthermore, as this example makes clear, the arrangement of points in DV space need bear little obvious similarity to the arrangement of points in IV space. Pairs of points in IV space that differ on only one independent variable (e.g., number of study presentations) may differ on both dependent variables (i.e., HR and FAR) when mapped to DV space. We discuss the consequences of this in more detail in Chapter 3 when we consider the concept of functional dissociation. The important point to note at this stage is that because the state-trace is two-dimensional, it is possible to infer that LV space cannot have fewer than two dimensions. In other words, that at least one of the dependent variables is a function of two or more latent variables.

Given the foregoing, one might be tempted to suppose that the dimensionality of the state-trace is equal to the dimensionality of LV space. That is, Equation (1.2) can be replaced with the equality, dim(state-trace) = dim(LV space). However, as we show next, the dimensionality of the state-trace is determined by each of the following:

1. The dimensionality of LV space.
2. Functional dependence of the input mapping.
3. Functional dependence of the output mapping.

We discuss each of these in turn.

Dimensionality of LV space

As stated in Equation (1.2), the dimensionality of the state-trace cannot exceed the dimensionality of LV space. It follows that if LV space is one-

dimensional then the dimensionality of the state-trace cannot be greater than one. Figure 1.5 illustrates this idea. In this case, the observed points in DV space are not distributed over two dimensions but are constrained to fall on a one-dimensional curve. The general structure is based on the same hypothetical recognition memory experiment that forms the basis of Figure 1.4. The difference is that in Figure 1.5 the theoretical structure instantiates an alternative recognition memory model in which d' is always equal to one, resulting in an LV space composed of only one latent variable – the decision criterion, c. The corresponding input mapping, adapted from Equation (1.3) is,

$$f : \left\{ c = k/2 + \ln\left(\frac{1-p}{p}\right)/k \right. \tag{1.4}$$

and the output mapping, adapted from Equation (1.1) is,

$$g : \left\{ \begin{array}{l} FAR = \Phi(-c) \\ HR = \Phi(1-c) \end{array} \right. \tag{1.5}$$

The input mapping, f, maps points in a two-dimensional IV space to points in a one-dimensional LV space. The output mapping maps points in the one-dimensional LV space to points on a one-dimensional curve in two-dimensional DV space, shown in the right panel of Figure 1.5.

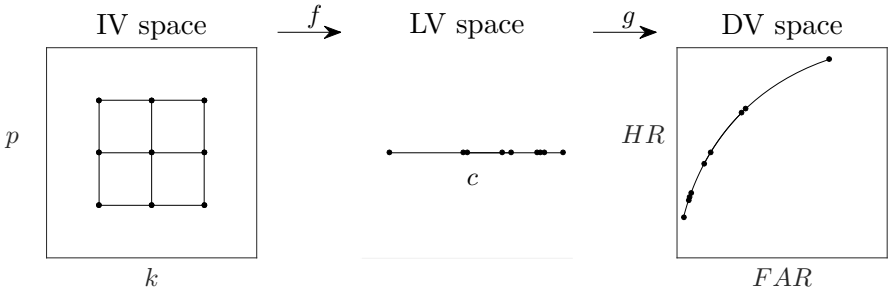

Fig. 1.5 Geometrical interpretation of the general structure defined by a signal detection model in which LV space is one-dimensional. IV space is defined by two independent variables; number of study presentations (k) and probability of a target at test (p). LV space is defined by a single latent variable, corresponding to the decision criterion (c). DV space is defined by the dependent variables of false alarm rate (FAR) and hit rate (HR).

Functional dependence of the input mapping

While the dimensionality of the state-trace cannot be greater that of LV space, it may be less. This can occur if the input mapping is *functionally dependent* (Dunn, Kalish & Newell, 2014). Figure 1.6 illustrates this idea. The general structure shown in this figure is the same as that shown in Figure 1.4 with the exception that the IV space is defined by number of study presentations (1, 2, or 3) and a 'levels of processing' manipulation (low, medium, high). In memory research, levels of processing describes a property of a study task that affects later memory. A study task may require a person to make an explicit judgment about a particular feature of the studied item. If the items are words, judging whether a word contains a particular vowel, judging whether it rhymes with another word, or judging whether it refers to an animate or inanimate object are examples of study tasks that vary in terms of level of processing from low to high (Craik & Lockhart, 1972). Generally, memory is poor at low levels of processing and high for high levels of processing.

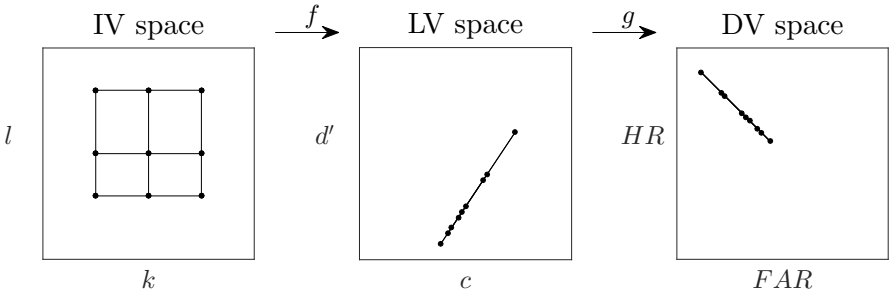

Fig. 1.6 Geometrical interpretation of the general structure defined by a theoretical structure in which the input mapping is functionally dependent. The IV space is defined by two independent variables; number of study presentations (k) and level of processing (l). The LV space is defined by two latent variables; sensitivity (d') and decision criterion (c). The DV space is defined by two dependent variables; false alarm rate (FAR) and hit rate (HR).

The input mapping defined by Equation 1.3 does not accommodate the levels of processing factor and thus a new input mapping is required. We assume that this factor, like number of study presentations, affects d'. The equation for c remains the same but, because the probability of a target at test does not vary, it is fixed at 0.5 in this design. Thus, $\ln\left(\frac{1-p}{p}\right) = 0$. These considerations lead to the following input mapping:

$$f : \begin{cases} d' = kl \\ c = d'/2 \end{cases} \tag{1.6}$$

where k is the number of study presentations, as before, and l is a variable representing low, medium and high level of processing in arbitrary units.

As shown in Figure 1.6, although LV space is defined to be two dimensional, the input mapping defined by Equation (1.6) fails to generate a two-dimensional set of points. Instead, the points fall on a one-dimensional curve (in this case, a straight line). This is because the input mapping is functionally dependent. That is, one or more of the component functions of the input mapping are themselves functions of the remaining component functions. In Equation 1.6, it is clear that c is a function of d'. Formally, given two functions, f_1 and f_2, they are said to be functionally dependent if there exists another function, F, such that $F(f_1, f_2) = 0$ (Loomis & Sternberg, 1990). In this example, $f_1(k, l) = kl$ and $f_2(k, l) = kl/2$ and so the relevant function is $F(f_1, f_2) = f_1 - 2f_2$ which is equal to zero for all k and l.

Because the points in LV space form a one-dimensional object, the dimensionality of the state-trace is necessarily one.

Functional dependence of the output mapping

The dimensionality of the state-trace may be also less than the dimensionality of LV space if the output mapping is functionally dependent (Dunn et al., 2014). This is illustrated in Figure 1.7. Here, the IV space, input mapping and LV space are the same as shown in Figure 1.4. The only difference is in the nature of the DV space and the associated output mapping. To illustrate functional dependence of the output mapping, the DV space is defined by the hit rate (HR) and the so-called z-transform of the hit rate (zHR). This corresponds to the following output mapping:

$$g : \begin{cases} zHR = d' - c \\ HR = \Phi(d' - c) \end{cases} \tag{1.7}$$

The output mapping is functionally dependent because $HR = \Phi(zHR)$ for all values of d' and c. Equivalently, for $f_1 = d' - c$, $f_2 = \Phi(d' - c)$, we obtain the function, $F(f_1, f_2) = \Phi(f_1) - f_2 = 0$ for all d' and c. This functional dependence means that even though each dependent variable is a function of two distinct latent variables, d' and c, the resulting state-trace is one-dimensional. Although this is a trivial example and would represent an obvious problem in the selection of dependent variables, the possibility of functional dependence of the output mapping is not always so readily excluded. For example, in an fMRI study, where the two dependent variables may correspond to levels of activation in different brain regions, it is quite possible for these regions to be functionally dependent. In fact, the possibility of a functional link between the two regions may be of considerable theoretical interest.

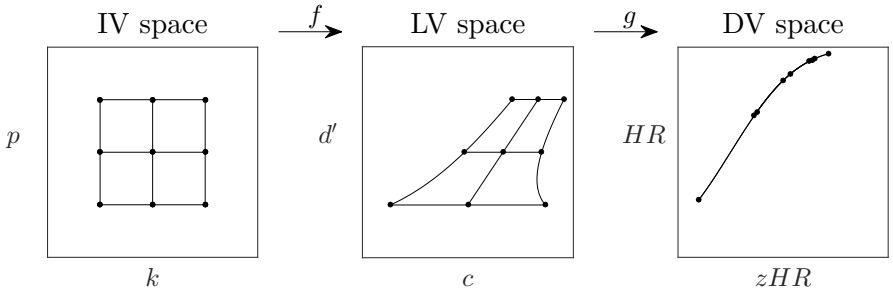

Fig. 1.7 Geometrical interpretation of the general structure defined by the signal detection model in which the output mapping is functionally dependent. The IV space is defined by two independent variables; number of study presentations (k) and probability of a target at test (p). The LV space is defined by two latent variables; sensitivity (d') and decision criterion (c). The DV space is defined by two dependent variables; hit rate (HR) and its z-transform (zHR)

Differential influence

Functional dependence also represents a failure of *differential influence*. This concept can be illustrated using the structure shown in Figure 1.8. This portrays the functional relationship between two independent variables, a and b, and two latent variables, u and v, corresponding to a generic input mapping, $u = f_u(a, b)$ and $v = f_v(a, b)$. A similar graph could be constructed for a generic output mapping relating the two latent variables and two dependent variables (and the same argument would apply).

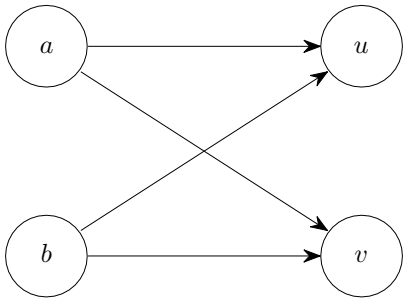

Fig. 1.8 Graph of the functional relationship between two independent variables, a and b and two latent variables, u and v, consistent with the input mapping, $u = f_u(a, b)$, and $v = f_v(a, b)$.

Simply put, differential influence means that the joint effect of a and b on u differs from their joint effect on v. Put another way, we say that u and v are *differentially affected* by a and b. However, we can (and should) be more precise than this and the aim of this section is to develop an explicit definition of differential influence to make its connection to functional dependence clear.

Consider the *linear* input mapping, f, defined as:

$$f : \begin{cases} u = f_u(a,b) = w_{ua}a + w_{ub}b \\ v = f_v(a,b) = w_{va}a + w_{vb}b \end{cases}$$

where $w_{ua}, w_{ub}, w_{va}, w_{vb}$ are weights, or coefficients, that quantify the size of the change of a latent variable caused by a change in an independent variable. For example, a unit change in a will lead to a change of w_{ua} units on u and w_{va} units on v. As an illustration, consider the following linear input mapping:

$$f : \begin{cases} u = 2a + b \\ v = 2a + 4b \end{cases} \tag{1.8}$$

This states that for every unit change in a, both u and v change by 2 units, while for every unit change in b, u changes by 1 unit and v changes by 4 units. This mapping is illustrated in Figure 1.9. Figure 1.9(a) shows three points in IV space – an initial point, (a', b'), and two additional points; the point, $(a' + \delta a, b')$, generated by adding an increment Δa to a' and the point, $(a', b' + \delta b)$, generated by adding an increment Δb, to b'. Figure 1.9(b) shows how these points are mapped to LV space by Equation (1.8). The initial point in IV space, (a', b'), is mapped to the point, $(f_u(a', b'), f_v(a', b'))$, in LV space. This is called the *image* of (a', b') and, according to Equation (1.8), is equal to $(2a'+b', 2a'+4b')$. The image of the point, $(a'+\Delta a, b')$, is found by shifting the image of the initial point by Δu_a on the u axis and by Δv_a on the v axis. Because the input mapping is linear, $\Delta u_a = w_{ua}\Delta a$, and $\Delta v_a = w_{va}\Delta a$. Similarly, the image of the point, $(a', b' + \Delta b)$, is found by shifting the image of the initial point $\Delta u_b = w_{ub}\Delta b$ along the u axis and $\Delta v_b = w_{vb}\Delta b$ along the v axis.

The reason why Equation (1.8) defines a functionally independent mapping is because the three points in Figure 1.9(b) do not all fall on the same straight line – thereby preserving the two-dimensionality of IV space. By examining the geometry of Figure 1.9(b), it is clear that the three points are not collinear whenever the ratio of changes in u and v due to a, $\Delta v_a/\Delta u_a$, differs from the ratio of changes in u and v due to b, $\Delta v_b/\Delta u_b$. In other words, when a and b differentially influence u and v. That these two ratios differ for Equation (1.8) is easily shown:

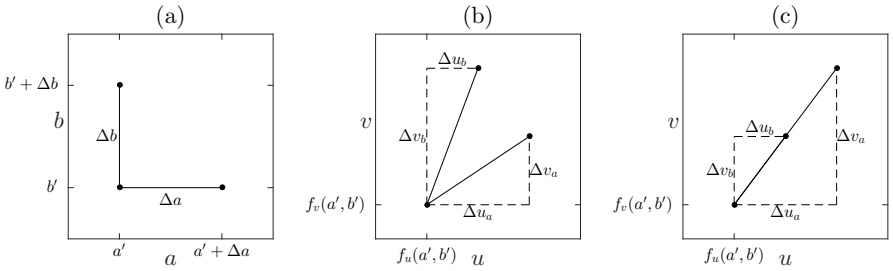

Fig. 1.9 The effects of a functionally independent and functionally dependent linear input mapping. (a) IV space defined by independent variables, a and b, containing three points; an initial point (a', b') and the points $(a' + \Delta a, b')$ and $(a', b' + \Delta b)$. (b) LV space defined by latent variables, u and v, and containing the image of the three points in IV space mapped according to the functionally independent linear mapping given by Equation (1.8). (c) The same LV space containing the image of the three points in IV space mapped according to the functionally dependent linear mapping given by Equation (1.9).

$$\begin{aligned}
\Delta v_a / \Delta u_a &= (w_{va} \Delta a) / (w_{ua} \Delta a) \\
&= w_{va} / w_{ua} \\
&= 2/2 \\
&= 1 \\
\Delta v_b / \Delta u_b &= (w_{vb} \Delta b) / (w_{ub} \Delta b) \\
&= w_{vb} / w_{ub} \\
&= 4/1 \\
&= 4
\end{aligned}$$

Figure 1.9(c) shows what happens if the points in IV space are mapped to LV space by the functionally dependent linear mapping given by:

$$f : \begin{cases} u = 2a + b \\ v = 4a + 2b \end{cases} \tag{1.9}$$

As can be seen, the three points in LV space are now collinear (fall on the same straight line) thereby reducing the dimensionality to one. Being collinear, the ratio of changes in u and v due to a, $\Delta v_a / \Delta u_a$, is the same as the ratio of changes in u and v due to b, $\Delta v_b / \Delta u_b$. In other words, a and b do not differentially influence u and v. Again, this is easily shown:

$$\begin{aligned}
\Delta v_a/\Delta u_a &= (w_{va}\Delta a)/(w_{ua}\Delta a) \\
&= w_{va}/w_{ua} \\
&= 4/2 \\
&= 2 \\
\Delta v_b/\Delta u_b &= (w_{vb}\Delta b)/(w_{ub}\Delta b) \\
&= w_{vb}/w_{ub} \\
&= 2/1 \\
&= 2
\end{aligned}$$

The foregoing provides a geometric intuition for functional independence based on differential influence under a linear mapping. But what if the mapping is not linear? Fortunately, both the intuition and the mathematics can be generalized in a straightforward way.

As mentioned earlier, a linear input mapping from two-dimensional IV space to two-dimensional LV space can be stated in general form as,

$$f : \begin{cases} u = w_{ua}a + w_{ub}b \\ v = w_{va}a + w_{vb}b \end{cases}$$

which can also be written in vector-matrix form as follows,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} w_{ua} & w_{ub} \\ w_{va} & w_{vb} \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix}.$$

The requirement of functional independence is that $w_{va}/w_{ua} \neq w_{vb}/w_{ub}$. Rearranging these terms, this is equivalent to,

$$w_{ua}w_{vb} - w_{va}w_{ub} \neq 0$$

That is, a linear mapping is functionally independent if the determinant of the matrix of weights is not equal to zero.

We can generalize this result to a non-linear mapping by noting that the matrix of weights corresponds to the *Jacobian matrix*, Df, of the linear mapping, f. The Jacobian matrix of a multivariate function is the matrix of partial derivatives of each component function with respect to each argument (parameter or latent variable). That is,

$$Df = \begin{bmatrix} \frac{\partial u}{\partial a} & \frac{\partial u}{\partial b} \\ \\ \frac{\partial v}{\partial a} & \frac{\partial v}{\partial b} \end{bmatrix} = \begin{bmatrix} w_{ua} & w_{ub} \\ w_{va} & w_{vb} \end{bmatrix}$$

The requirement that the determinant of the weight matrix not be zero is therefore equivalent to the requirement that,

$$\frac{\partial u}{\partial a}\frac{\partial v}{\partial b} - \frac{\partial v}{\partial a}\frac{\partial u}{\partial b} \neq 0 \tag{1.10}$$

Suppose f is a non-linear mapping. The Jacobian matrix of f at a point in its domain (here, in the IV space) defines a linear mapping to the neighbourhood of the image of the point in its codomain (here, the LV space). If Equation (1.10) is true at every point in the domain of f then f is functionally independent. If Equation (1.10) is false at every point in the domain of f then f is functionally dependent[5]. The intuition behind this is that if, at every point, p, in LV space, a change in each of two independent variables yields points that lie on a straight line in the neighbourhood of p then, as we move from point to point in LV space, the mapping must trace out a one-dimensional curve in DV space.

We can now use Equation (1.10) to show that the two input mappings, given in the previous sections by Equation (1.3) and Equation (1.6), are functionally independent and functionally dependent, respectively. Equation (1.3) defines the input mapping,

$$f_1 : \begin{cases} d' = k \\ c = k/2 + \ln\left(\dfrac{1-p}{p}\right)/k \end{cases}$$

Its Jacobian matrix is given by,

$$Df_1 = \begin{bmatrix} 1 & 0 \\ \frac{k^2 - \ln\left(\frac{1-p}{p}\right)}{2k^2} & \frac{-1}{kp(1-p)} \end{bmatrix}$$

And, applying Equation (1.10) we find,

$$\frac{-1}{kp(1-p)} - 0 \neq 0$$

Thus, f_1 is functionally independent. Equation (1.6) defines the input mapping,

$$f_2 : \begin{cases} d' = kl \\ c = kl/2 \end{cases}$$

Its Jacobian matrix is given by,

$$Df_2 = \begin{bmatrix} l & k \\ \frac{l}{2} & \frac{k}{2} \end{bmatrix}$$

And, applying Equation (1.10) we find,

$$\frac{lk}{2} - \frac{kl}{2} = 0$$

Thus, f_2 is functionally dependent.

[5] This is a strong condition. Clearly, Equation (1.10) may be true for some points and false for others.

In summary, if an input mapping is functionally independent then the latent variables are differentially influenced by the independent variables[6]. Similarly, if an output mapping is functionally independent then the dependent variables are differentially influenced by the latent variables.

At this point, we note a special case of differential influence where one independent (or latent) variable affects only one latent (or dependent) variable. As an example, suppose that the latent variable, u, is affected by changes in a but unaffected by changes in b. Then we say that u is *selectively influenced* by a. In this case, because $\frac{\partial u}{\partial b} = 0$, Equation (1.10) reduces to,

$$\frac{\partial u}{\partial a}\frac{\partial v}{\partial b} \neq 0$$

.

It follows that if b affects v (i.e., $\frac{\partial v}{\partial b} \neq 0$) then the relevant mapping must be functionally independent. We will discuss selective influence in Chapter 3 in the context of functional dissociation. For now, it is sufficient to note that selective influence is simply a special case of differential influence which is itself another way of describing functional independence.

Differential influence is an important consideration in the interpretation of the dimensionality of a state-trace. Specifically, if the dimensionality of the state-trace is less than the number of latent variables postulated by a model then either (a) the model is wrong, or (b) there is a failure of differential influence of the independent variables on the latent variables, or (c) there is failure of differential influence of the latent variables on the dependent variables. In the case of (b), a remedy is to repeat the experiment with a different set of independent variables in the hope that these will differentially affect the latent variables. Obviously, if no such variables can be found, confidence in the model may decline. In the case of (c), a remedy is to repeat the experiment with a different set of dependent variables (if possible) in the hope that these will differentially affected by the latent variables. We discuss this issue more fully in Section 9.4.

[6] We have explored this in detail for two-dimensional spaces. Similar considerations apply to higher-dimensional spaces.

Chapter 2
Monotonicity

2.1 Introduction

In the previous chapter, we showed that the dimensionality of the state-trace cannot be greater than the dimensionality of the LV space. This suggests that it may be possible to place an upper bound on the dimensionality of LV space based on the dimensionality of the state-trace without having to specify an output mapping. In particular, we consider the following two mutually exclusive hypotheses[1]:

Hypothesis 1: dim(LV space) = 1
Hypothesis 2: dim(LV space) > 1

The key idea is that if dim(state-trace) > 1 then Hypothesis 1 can be rejected. However, because any observed state-trace necessarily consists of a finite number of points, without placing additional constraints on the form of the output mapping, it is not possible to determine its dimensionality. The nature of the problem is illustrated in Figure 2.1 which shows a finite state-trace embedded in a two-dimensional DV space. This set of points was generated by applying the two-dimensional output mapping defined by Equation (1.1) to the factorial combination of three distinct values of d' and four distinct values of c. The shaded region in Figure 2.1(a) is the two-dimensional patch consisting of the set of points in DV space corresponding to all possible values of d' and all possible values of c. The observed state-trace is a subset of this patch. But, as shown in Figure 2.1(b), this set of points is also consistent with a one-dimensional LV space. Here, each point in the observed state-trace falls on a single one-dimensional curve defined by the following output mapping:

[1] We exclude for the present the hypothesis that dim(LV space) = 0 where the state-trace consists of a single point in DV space.

$$g^* : \begin{cases} FAR = g_1^*(u) \\ HR = g_2^*(u) \end{cases} \tag{2.1}$$

where g_1^* and g_2^* are polynomial functions constructed to pass through the observed state-trace.

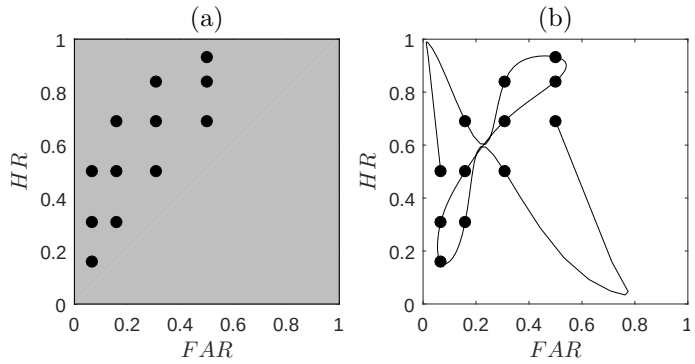

Fig. 2.1 Dimensionality of the state-trace. (a) An observed state-trace (filled circles) defined as a subset of the two-dimensional patch (shaded region) corresponding to the output mapping defined by Equation (1.1). (b) The same observed state-trace and a one-dimensional curve corresponding to the output mapping defined by Equation (2.1).

Because a finite state-trace does not uniquely determine an intrinsic dimensionality, we consider a modification of Hypothesis 1 in which a constraint is added to the output mapping. Let G be the set of all possible one-dimensional mappings from LV space to DV space. Let P be a subset of G such that each mapping in P is constrained in a specified way. For example, P may be the set of all linear mappings, or all curvilinear mappings, or all monotonically increasing mappings. Let g be the output mapping proposed to have generated the observed state-trace. Then Hypothesis 1 can be modified as follows:

Hypothesis 1′: dim(LV space) = 1 and $g \in P$

Let S be the set of finite state-traces generated by each output mapping, $g \in G$. Let $Q \subseteq S$ be the set of finite state-traces generated by each output mapping, $p \in P$. Let $s \in S$ be an observed state-trace. Then if $s \notin Q$, Hypothesis $1'$ can be rejected. In other words, either dim(LV space) > 1 or $g \notin P$. Because it is of greater theoretical interest to conclude that dim(LV space) > 1, P should be defined in such a way that this conclusion is overwhelmingly more likely than the alternative, that $g \notin P$.

Figure 2.2 illustrates the underlying logic of this problem in the form of a binary tree. The first branch of the tree contrasts Hypothesis 1 and Hypothesis 2; namely, whether the dim(LV space) = 1 or dim(LV space) > 1. If dim(LV space) = 1 then the next branch contrasts whether the output mapping, g, is an element of P or is not an element of P. If $g \in P$ then the observed state-trace s must be an element of Q. If $g \notin P$ then there is a further branch – either s is an element of Q or it is not an element of Q. On the rightmost branch, if dim(LV space) > 1 then the observed state-trace, s, is also either an element of Q or it is not an element of Q.

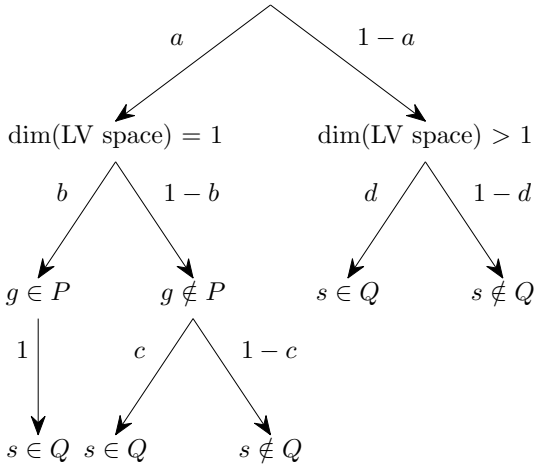

Fig. 2.2 A binary tree representing the decision space for state-trace inference.

Associated with each branch of the binary tree in Figure 2.2 is a probability, indicated by the letters, a, b, c, and d. Here, a, is the (prior) probability that the dimensionality of LV space is one. Because we have assumed that Hypotheses 1 and 2 are mutually exclusive, the (prior) probability that the dimensionality is greater than one is $(1 - a)$; b is the probability that the output mapping g is an element of P given that the dimensionality of the LV space is one; c is the probability that the observed state-trace, s, is an element of Q given that it was generated by a one-dimensional output mapping not an element of P; and d is the probability that the observed state-trace is an element of Q given that the dimensionality of LV space is greater than one. Although it is impossible to attach numerical estimates to these probabilities, they do enable us to evaluate the implications of different choices of P.

First, suppose $s \in Q$. Then the (posterior) probability that dim(LV space = 1) given $s \in Q$ is,

$$P(\dim(\text{LV space} = 1)|s \in Q) = \frac{a(b + (1 - b)c)}{(a(b + (1 - b)c) + (1 - a)d}$$

Similarly, the probability that dim(LV space) > 1 given $s \in Q$ is,

$$P(\text{dim(LV space} > 1) | s \in Q) = \frac{(1-a)d}{(a(b+(1-b)c)+(1-a)d}$$

And the corresponding odds ratio is,

$$\frac{P(\text{dim(LV space} = 1)/s \in Q)}{P(\text{dim(LV space} > 1)/s \in Q)} = \frac{a}{(1-a)} \cdot \frac{b+(1-b)c}{d} \tag{2.2}$$

The first term on the right hand side of the odds ratio is the ratio of a, the prior probability that dim(LV space) $= 1$, to $(1-a)$, the prior probability that dim(LV space) > 1. The second term is the *likelihood ratio* which tells us how much the ratio of prior probabilities is changed by the evidence that $s \in Q$. The larger the odds ratio, the greater the evidence that dim(LV space $= 1$). We can use Equation (2.2) to assess the implications of different choices of P. To do this, we consider two extreme choices; one where P is a highly restricted set of mappings and one where P is inclusive of almost all possible mappings.

Suppose that P is a highly restrictive set, such as the set of linear mappings. In this case, Q is the set of all linear state-traces[2]. If $g \notin P$ then there are vastly many ways that a state-trace resulting from g is not in Q. Therefore, both c and d will be small and the likelihood ratio will be approximately equal to b/d. However, unless there are compelling reasons for believing that g is in fact a linear mapping, b will also tend to be close to zero. Even so, d may still be orders of magnitude smaller than b in which case the likelihood ratio given $s \in Q$ may be very large, strongly supporting the hypothesis, dim(LV space) $= 1$. But, because b, c and d are all small, observation of $s \in Q$ will be a (potentially very) rare event.

Suppose that P is a highly inclusive set, such as the set of polynomials of high degree. Leaving to one side the problem of defining Q[3], because it will contain nearly all possible state-traces, both c and d are close to one. For similar reasons, b is also close to one and hence the outcome $s \in Q$ will be frequently observed. However, for b, c, and d close to one, the likelihood ratio is also close to one. The result is that if $s \in Q$, neither Hypothesis 1 nor Hypothesis 2 is supported.

However, $s \in Q$ is just only one kind of evidence. We can also consider what can be concluded from the opposite, when $s \notin Q$. Suppose $s \notin Q$. Then the (posterior) probability that dim(LV space $= 1$) given $s \notin Q$ is,

$$P(\text{dim(LV space} = 1) | s \notin Q) = \frac{a(1-b)(1-c)}{a(1-b)(1-c)+(1-a)(1-d)}$$

[2] If x is a linear function of u and y is a linear function of u then y is a linear function of x and vice versa.

[3] That is, defining a testable property of the state-trace that identifies it as an element of Q.

Similarly, the probability that dim(LV space) > 1 is,

$$P(\dim(\text{LV space} > 1)|s \notin Q) = \frac{(1-a)(1-d)}{a(1-b)(1-c)+(1-a)(1-d)}$$

And the corresponding odds ratio is,

$$\frac{P(\dim(\text{LV space} = 1)/s \notin Q)}{P(\dim(\text{LV space} > 1)/s \notin Q)} = \frac{a}{(1-a)} \cdot \frac{(1-b)(1-c)}{(1-d)} \tag{2.3}$$

When P is the restrictive set considered earlier, the outcome, $s \notin Q$, is the more likely. But because c and d are close to zero, the likelihood ratio is approximately equal to $(1-b)$ and, since b is small, this will be close to one. Thus, if P is a restrictive set then even though $s \notin Q$ is the most likely outcome, neither Hypothesis 1 nor Hypothesis 2 is supported by it.

When P is the inclusive set considered earlier, both c and d are also close to one and so the likelihood ratio will depend on $(1-b)$ which, since b is close to one, is close to zero[4]. In other words, if $s \notin Q$ then Hypothesis 2, that dim(LV space) > 1, is strongly supported. However, because b, c and d are all large, observation of $s \notin Q$ will be a (potentially very) rare event.

In summary, if P is chosen to be a restrictive set then the observation that $s \in Q$ is strong evidence that dim(LV space) $= 1$. But $s \in Q$ will rarely be observed and nothing can be concluded from the more likely outcome, $s \notin Q$. On the other hand, if P is chosen to be an inclusive set then the opposite applies. The observation that $s \notin Q$ is strong evidence that dim(LV space) > 1 but this outcome will rarely be observed and little if anything can be concluded from the more likely outcome, $s \in Q$.

The quandary concerning a choice of P arises because we have, in both cases, defined it without reference to the actual state of nature. For example, if all one-dimensional mappings involving psychological measures were, as a matter of fact, linear then defining P as the set of linear mappings would be optimal. The difference from the restrictive case considered above is that b should now be close to one and the observation that $s \in Q$, while retaining its diagnosticity, would not be rare. Furthermore, because $b \approx 1$, observation that $s \notin Q$ is also highly diagnostic, supporting the hypothesis that dim(LV space) > 1.

The above argument applies to any set P defined with respect to the actual state of nature. In this case, b will be large and c and d relatively small. As $b \to 1$, the diagnosticity of $s \not\in Q$ increases. As $c, d \to 0$, the diagnosticity of $s \in Q$ increases. Of course, the state of nature could be that there is no systematicity to the set of one-dimensional output mappings – that is, each dependent variable may be any function at all of a single latent variable. In this case, as discussed earlier in this section (see Figure 2.1), state-trace analysis would be impossible.

[4] This assumes that $c \leq d$.

Fortunately, we believe that a case can be made that in many situations, the state of nature is such that P can be defined as the set of *monotonic* output mappings. A single-valued function, $g_i(u)$, is said to be *monotonically increasing* if, for all $u_1 < u_2$, $g_i(u_1) \leq g_i(u_2)$. Similarly, the function is said to be *monotonically decreasing* if, for all $u_1 < u_2$, $g_i(u_1) \geq ig(u_2)$. Where the meaning is clear, we use the term monotonic to describe a monotonically increasing function. A mapping, $g = (g_1, \ldots, g_n)$, is said to be monotonic if each of its component functions, g_1, ..., , g_n, is monotonic. We justify our choice of the monotonicity assumption based on two principal reasons.

First, monotonicity strikes an appropriate balance between generality on the one hand and tractability on the other. In terms of generality, the set of monotonic output mappings is broad enough to include a large set of plausible psychometric functions. This means that b will be close to one. Monotonicity is also tractable. Given a monotonic output mapping, an element of the set, Q, is readily identified: If dependent variables x and y are both (distinct) monotonic functions of a latent variable u then the resulting state-trace is itself monotonic. In addition, monotonicity is not over-inclusive. If dim(LV space) > 1 then there many ways for the observed state-trace not to be monotonic (i.e., $d \ll 1$).

Second, there are good arguments that, given the current state of knowledge in psychological science, a more restrictive assumption is unjustifiable (Loftus, 1978). This is despite the fact that much research practice is predicated on the assumption that each component function of an output mapping is in fact *linear* and, in some contexts, that each component function is *identically linear* (Wagenmakers, Krypotos, Criss & Iverson, 2012). For this reason, in the remainder of this chapter, we detail the argument for why we believe monotonicity is the appropriate assumption for STA. In Chapter 3 we discuss the implications of monotonicity for two alternative procedures that have been frequently been used to determine the dimensionality of the state-trace – functional dissociations and interaction terms in the analysis of variance (ANOVA). We show that both of these approaches depend upon unjustified assumptions and that both are superseded by STA.

2.2 The problem of nomic measurement

We have introduced the assumption of monotonicity as a plausible, general, and tractable constraint on the output mapping. While it may be desirable to adopt a more restrictive assumption than monotonicity, in most situations this cannot be justified. It turns out that this problem is not peculiar to psychological science but is a general problem that affects all forms of scientific measurement. Chang (2004) has called this the problem of *nomic measurement*. It arises whenever there is interest in measuring a quantity, u, which is not directly observable but can be inferred from another quantity,

x, which is directly observable. In order to draw the required inference, the form of the function that relates u and x, $x = g(u)$, must be known. The problem is that g cannot be discovered or tested empirically because that would involve knowing the values of both u and x, and u is the unknown variable that we are trying to measure.

Chang (2004) has described how the problem of nomic measurement historically bedevilled attempts to measure temperature over a 250-year period from 1600 to the middle of the nineteenth century. In this case, temperature is the unobserved quantity, u, and changes in the volume or extent of a suitable substance (such as mercury) is the quantity, x, that can be directly observed and measured. The problem is to determine how changes in the volume of a substance (i.e., the dependent variable) are related to changes in temperature (i.e., the latent variable). The familiar textbook accounts of fixing a thermometer scale (e.g., in relation to the freezing and boiling points of water as in the Celsius scale) forms only one part of this story and is relevant only after the form of the function relating temperature to volume has been established. Interestingly, this did not occur until the very end of the period in question, long after working thermometers had been developed and employed. In fact, a considerable amount of scientific work was undertaken without having first solved the measurement problem. In part, this was made possible by virtue of the so-called *zeroth law of thermodynamics* which states that two substances in thermal equilibrium must be at the same temperature (Reif, 1965). Consequently, it is possible to determine that the unknown temperatures of two different substances (or of the same substance at different times) are equal if the volume of a third substance (such as that contained in a thermometer) placed in proximity to each remains the same. Unfortunately, no such law applies to psychological measurement.

Although some kinds of scientific investigations involving temperature were undertaken without solving the problem of nomic measurement, the lack of an agreed functional relationship was increasingly viewed as problematic. Various attempts were made to solve the problem empirically (e.g., using the 'method of mixtures' as discussed by Chang, 2004, pp. 60-64) but these were unsuccessful. They failed not from a lack of effort or ingenuity but because the measurement problem is fundamentally theoretical, not empirical. Its solution in the case of temperature depended ultimately on the development of the kinetic theory of heat in the middle of the nineteenth century. This led to a deeper theoretical understanding of the nature of temperature, replacing the earlier view that it was a kind of fluid, called *caloric*, with the now-accepted view that it is the average kinetic energy of the component particles of an *ideal gas*. Based on this theory, it became possible to show that the volume of an ideal gas at a constant pressure must be a linear function of its temperature. By identifying some actual gases as close to ideal (i.e., composed of independent point particles), researchers were able to use changes in volume (at a constant pressure) or changes in pressure (at constant

volume) of gas thermometers to calibrate changes in other substances such as mercury[5] (Beattie, Blaisdell, Kaye, Gerry & Johnson, 1941).

The fact that the problem of nomic measurement dogged the measurement of a familiar physical quantity such as temperature should give us pause that the same problem is unlikely to be avoided in the measurement of psychological quantities such as memory strength, selective attention, learning, preference, and so on. The implications of not solving or rather, ignoring, this problem in psychological science is discussed in the next section.

The face-inversion effect

In this section, we illustrate implications of the failure to solve the problem of nomic measurement in psychology with reference to the *face inversion effect* (see Loftus et al., 2004, for a more extensive discussion of this issue in relation to STA).

The face-inversion effect refers to a greater effect of inversion (at study or test) on memory for pictures of faces compared to pictures of other mono-oriented stimuli such as houses. In the original experiment by Yin (1969), participants studied upright and inverted pictures of faces or houses (as well as other stimuli). Later, their memory for these pictures was tested in a two-alternative forced choice task in which participants were required to select the studied item from a pair of items of the same type (both faces or houses) and in the same format (both upright or inverted). It was found that accuracy was less for inverted items than for upright items and, critically, that this difference was greater for faces than for houses. This has generally been interpreted as showing that faces are perceived, represented, or processed differently to houses (and other mono-oriented stimuli) such that the difference in memory strength between upright and inverted faces is greater than the corresponding difference in memory strength for houses (Valentine, 1988). In terms of the general structure introduced earlier, the face-inversion effect is interpreted as requiring the existence of two latent variables – one corresponding the memory strength of faces, the other corresponding to the memory strength of houses[6].

In order to appreciate the implications of the problem of nomic measurement for psychological phenomena such as the face-inversion effect, it is helpful draw an analogy with the measurement of temperature (as previously discussed). Accordingly, a fixed period of study of faces and houses is analogous to the application of a fixed quantity of heat to two different

[5] As it turns out, mercury and many other substances, including water, do not expand linearly with temperature. Hence the need for careful calibration

[6] Although different interpretations are possible. For example, one latent variable may correspond to memory for mono-oriented objects and the second to the difference between this an memory for faces.

substances, say alcohol and water. The orientation of faces and houses is analogous to effectiveness of the heat source; study of an upright stimulus corresponds to exposure to a more effective heat source, e.g., one that is closer, while study of an inverted stimulus corresponds to exposure to a more distant, and hence less effective, heat source. Because there is no corresponding zero'th law of thermodynamics in psychology, there is no equivalent of a thermometer. Consequently, we draw an analogy between the accuracy of recognition of faces and houses to changes in volume of the corresponding substances – alcohol and water.

Armed with these correspondences, we can translate the face-inversion effect into the analogous temperature domain. Associating alcohol with faces and water with houses, the face-inversion effect corresponds to the following statement about temperature; a change in the *volume* of alcohol between a proximal and distal application of a fixed quantity of heat is greater than the corresponding change in the same *volume* of water. The theoretical claim that the difference in memory strength between upright and inverted faces is greater than the corresponding difference in memory strength for houses is therefore analogous to the claim that the difference in the *temperature* of alcohol after proximal and distal heat exposure is greater than the corresponding difference in the *temperature* of water. The fallacy in this inference is now apparent. Because the form of the function that relates a change in temperature to a change in volume differs for alcohol and water[7], changes in the volume of these liquids cannot be compared as if they directly correspond to changes in temperature. Furthermore, because the form of the function that relates a change in memory strength to a change in memory accuracy is unknown and, crucially, may differ for faces and houses, changes in memory accuracy for these stimuli cannot be compared as if they correspond to changes in memory strength (Loftus et al., 2004).

The problem of nomic measurement undermines interpretation of the face-inversion effect as a stimulus-specific effect on memory strength. It also affects the interpretation of phenomena where differences are found within the same kind of stimulus. This can be illustrated using the results of a recent study by citetLDBKR12 shown in Figure 2.3. In this study, the difference in recognition accuracy for upright and inverted faces was measured in two groups of participants. A trained group was given extensive practice at identifying inverted faces while an untrained group was not. Both groups were subsequently tested on their memory for equally novel upright and inverted faces. The results are shown in Figure 2.3(a) and demonstrate a reduced face inversion effect in the trained group compared to the untrained group. In line with almost everyone else in this field, the authors interpreted this interaction as revealing that training had a specific effect on the processing of faces, concluding that 'the face processing system maintains the capacity

[7] This is quantified by a constant known as the *thermal expansion coefficient* which is about 3.5 times greater for alcohol than for water – which is why alcohol is used in thermometers instead of water.

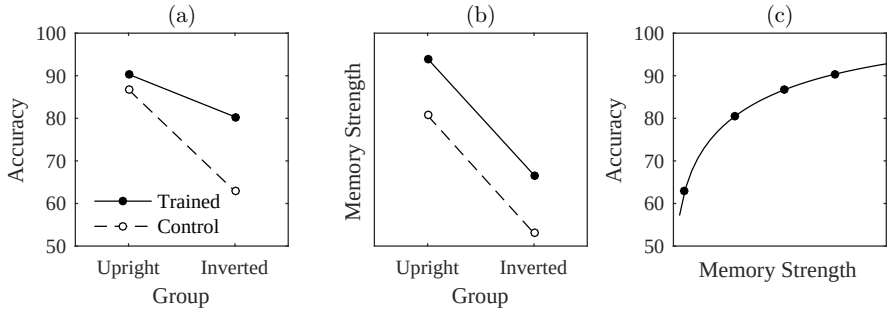

Fig. 2.3 The interpretation of interactions. Results found by Laguesse et al. (2012). (a) Interaction of stimulus orientation and group for memory accuracy. (b) Additivity between stimulus orientation and group for hypothetical memory strength. (c) Proposed (monotonic) function relating hypothetical memory strength and accuracy.

to be modulated by visual experience even well into adulthood' (p. 11). However, this conclusion depends upon the assumption that the function relating memory strength to accuracy is linear because only then can the interaction between group and orientation found for accuracy be interpreted as demonstrating a similar interaction for memory strength.

Although the memory-accuracy function may be linear, there is no reason to suppose that it is. This is the point made by Loftus (1978) in his seminal article on the interpretation of interactions. Because the function relating memory strength to accuracy is not known to be linear, the interaction found for accuracy does not imply a similar interaction in memory strength. This is illustrated in Figure 2.3(b) which shows an additive relationship between group and orientation for a hypothetical measure of memory strength. The function that determined these values is shown in Figure 2.3(c). Given this function, there is no change in the face inversion effect on memory strength due to training. This is the point originally made by Loftus (1978) and recently re-iterated by Wagenmakers, Ratcliff, Gomez and Iverson (2004).

2.3 Is monotonicity too restrictive?

While we have argued that a more restrictive assumption than monotonicity is unwarranted, it is possible that monotonicity is itself too restrictive. In a recent critique, Ashby (2014) proposed that in many situations there are good reasons to suppose that the one-dimensional output mapping is non-monotonic. In each of these situations, Ashby (2014) listed five examples, performance is optimal for some intermediate value of the latent variable of interest. A relatively simple case concerns accuracy in a signal detection task which first increases as a function of the response criterion until it reaches

an intermediate (optimal) value and then decreases. Following Dunn et al. (2014), we propose the following two points.

First, the non-monotonic functions highlighted by Ashby (2014) arise, at least in part, because of the composite nature of accuracy. In signal detection theory, for example, accuracy, A, is a function of the hit rate, HR), and the false alarm rate, FAR. Specifically, $A = HR - FAR + 1$. As shown in Equation (1.1), both HR and FAR are monotonically decreasing functions of the response criterion. Their difference, $HR - FAR$, is therefore potentially non-monotonic. However, by replacing accuracy with measures of hit and false alarm rates, this non-monotonicity can be removed.

Second, if a non-monotonic function cannot be removed in a simple way, and there are strong reasons for believing it to reflect the true state of nature, then monotonicity is genuinely too restrictive an assumption (similar to the linear mapping discussed earlier). In this case, it would be foolish to apply STA based on monotonicity. However, it would be necessary to develop an alternative approach based on the assumption that each output function is (or could be) a non-monotonic function of a particular kind – for example, one that increases up to a value of the latent variable and then decreases. While, in principle, it should be possible to implement the logic of STA under this assumption, it would be necessary to develop a test, analogous to the test for monotonicity, that could be applied to an observed state-trace in order to determine whether or not it has been generated by the relevant output mapping. In terms of the decision tree shown in Figure 2.2, it would be necessary to develop a test to determine whether or not $s \in Q$. The development of such a test may be a useful aim for future research. However, with this proviso in mind, for the remainder of this book, we assume that the one-dimensional output mapping is monotonic.

Chapter 3
Functional dissociation

3.1 Introduction

As described in Section 1.1, the aim of STA is to place a lower bound on the dimensionality of LV space based on the dimensionality of the state-trace. An alternative approach that has enjoyed far greater popularity and range of application is based on observation of *functional dissociations*. Two dependent variables, x and y, are said to be functionally dissociated if, across two or more levels of an independent variable there is a change in x but no change in y (Shallice, 1988; Dunn & Kirsner, 2003). This pattern is illustrated in the left panel of Figure 3.1 which shows that a functional dissociation corresponds to a particular kind of state-trace, namely a horizontal line or, if the axes are interchanged, a vertical line. It is commonly argued that a functional dissociation indicates that the two dependent variables depend on more than one latent variable (Dunn & Kirsner, 1988, 2003; Shallice, 1988). Such inferences have played an important role in theory development in a wide range of fields such as perceptual learning (Ashby & Maddox, 2005), recognition memory (Yonelinas, 2002), and syllogistic reasoning (Goel, Buchel, Frith & Dolan, 2000), and have become particularly prominent in neuroimaging studies where the central focus is to associate distinct psychological processes with different brain regions (for relevant reviews, see Price, 2010; Rajah & D'Esposito, 2005; Vilberg & Rugg, 2008).

However, a functional dissociation is neither necessary nor sufficient to infer that changes in two or more dependent variables are mediated by more than one latent variable. That it is not sufficient is apparent from the state-trace shown in Figure 3.1(a). The two points connected by the horizontal line segment are consistent with a monotonic state-trace derived from a single latent variable, u, and a monotonically increasing output mapping, $x = g_x(u)$, and $y = g_y(u)$, where the rate of change of y with respect to u is as close to zero as measurement error allows.

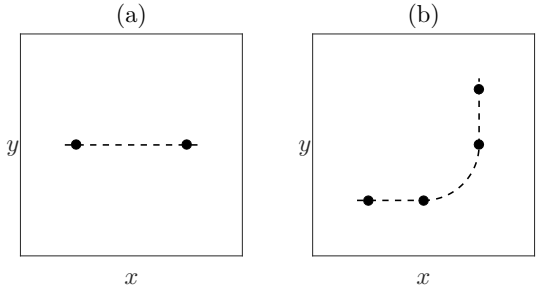

Fig. 3.1 State-trace plots of functional dissociations. (a) A single dissociation. (b) A double dissociation. In both cases, the state-trace is consistent with a monotonically increasing curve.

A functional dissociation corresponding to a horizontal or vertical line in DV space is commonly called a single dissociation and is generally acknowledged to offer relatively weak evidence against a single latent variable. It is frequently argued that stronger evidence is provided by a *double dissociation* defined as two complementary single dissociations (Shallice, 1988). Figure 3.1(b) illustrates a state-trace that, while incorporating a double dissociation, is also consistent with a monotonically increasing one-dimensional curve. Therefore, through these simple demonstrations, it is clear that a functional dissociation, whether single or double, is not sufficient to reject the hypothesis of a single latent variable.

A functional dissociation is also not necessary to reject the hypothesis of a single latent variable. This is demonstrated by the state-trace shown in Figure 1.3. Since no two points are aligned horizontally or vertically, there are no functional dissociations. Nevertheless, it is not possible to connect these points with a monotonically increasing one-dimensional curve. Therefore, these data are inconsistent with the hypothesis of a single latent variable despite the lack of any kind of dissociation.

3.2 The general linear model

The logic of functional dissociation is sometimes extended to situations in which the dependent variables of interest correspond to levels of a factor within a factorial design that is then typically analyzed using the general linear model (e.g., analysis of variance). The face-inversion effect, described in Chapter 2, is a case in point. The typical experiment is analyzed in terms of the effects of two independent variables, orientation (upright vs. inverted) and stimulus type (faces vs. houses), on memory accuracy. The effect of interest corresponds to an interaction between these variables such that the effect of orientation is greater for faces than for houses (Loftus et al., 2004). As

noted previously, this often interpreted as showing that different processes are involved in the perception and/or memory of these two kinds of stimuli.

The relationship between the general linear model, in the form of analysis of variance (ANOVA), and STA was discussed by Loftus et al. (2004) in their study of the face-inversion effect. This study included a third factor, stimulus duration – the length of time that a stimulus was presented during the study phase. Consequently, in this design the face-inversion effect corresponded to either a two-way interaction between stimulus orientation and stimulus type or a three-way interaction between stimulus orientation, stimulus duration, and stimulus type (or both). However, each of these interactions depends critically on the assumption of a linear output mapping.

ANOVA assumes that the dependent variable of interest in this design, memory accuracy, can be decomposed into a linear sum of 'effects' of the different independent variables and their interactions. Let p be memory accuracy (e.g., proportion correct), let a be a (dummy) variable that codes the level of stimulus duration, let b be a variable that codes for orientation (e.g., 0 = inverted, 1 = upright), and let c be a variable that codes for stimulus type (0 = houses, 1 = faces). Then the general linear model predicts p in terms of the following generic equation,

$$p = \mu + \alpha a + \beta b + \mu' c + \gamma ab + \alpha' ac + \beta' bc + \gamma' abc \tag{3.1}$$

Here, μ is the intercept term, α is a set of weights corresponding to the effects of stimulus duration (because duration may have more than one level), β is the effect of orientation, μ' is the effect of stimulus type, γ is a set of weights corresponding to the interaction of stimulus duration and orientation, and α', β' and γ' are the weights associated with the various interactions with stimulus type.

Although the linear model given by Equation (3.1) is routinely applied, in terms of the general structure of STA, it assumes that both the input and output mappings are linear. To see this, let x and y be the accuracy of recognition for houses ($c = 0$) and faces ($c = 1$), respectively. Then Equation (3.1) can be written in terms of x and y, as the following linear output mapping,

$$g : \begin{cases} x = u \\ y = u + v \end{cases} \tag{3.2}$$

where u is the latent variable obtained by setting $c = 0$ in Equation (3.1) and v is the latent variable obtained by setting $c = 1$ in Equation (3.1). The input mapping for these variables is accordingly,

$$f : \begin{cases} u = \mu + \alpha a + \beta b + \gamma ab \\ v = \mu' + \alpha' a + \beta' b + \gamma' ab \end{cases} \tag{3.3}$$

Based on the output mapping given by Equation (3.2) and the input mapping given by Equation (3.3), we can analyze the linear model in terms of STA. The first thing to note is that it is correct to conclude that a non-zero interaction between one or more of the independent variables and stimulus type indicates the existence of more than one latent variable. This is because if there are no interactions with stimulus type then all the interaction weights in the input function for v, α', β', γ', are all equal to zero with the result that $v = \mu'$. The output mapping is therefore,

$$g : \begin{cases} x = u \\ y = u + \mu' \end{cases} \tag{3.4}$$

The state-trace corresponding to Equation (3.4) is a straight line with a slope of one. This is illustrated in Figure 3.2(a) which shows simulated data from an experiment that factorially combines the independent variables of stimulus (house vs. face), orientation (inverted vs. upright), and five levels of stimulus study duration[1]. The filled circles correspond to the five upright conditions (each associated with one stimulus duration). The unfilled triangles correspond to the five inverted conditions. The line in the Figure is the expected straight line with slope of one and intercept of μ'. The data points fall on this line because there is no interaction between stimulus (the levels of which define the axes of DV space) and any of the other independent variables[2].

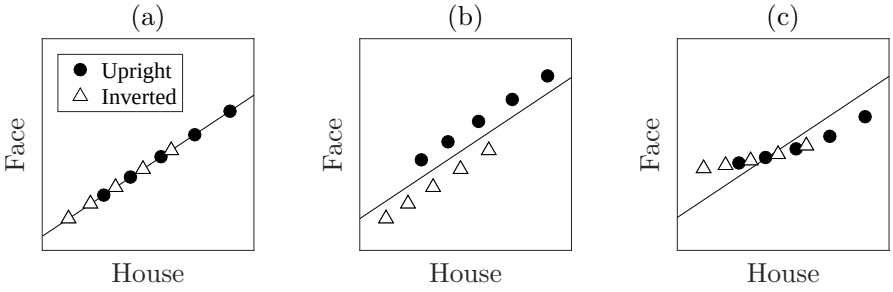

Fig. 3.2 Simulated state-trace plots of the face-inversion effect based on Equation (3.1). (a) State-trace plot for linear output mapping given by Equation (3.4). (b) State-trace plot for linear output mapping given by Equation (3.2). (c) State-trace plot for non-linear output mapping given by Equation (3.6).

In contrast, if at least one of the interaction weights in the input function for v is non-zero then the linear output mapping is a function of two distinct

[1] For ease of presentation, duration was modelled as a covariate with 5 levels rather than as a set of four dummy variables

[2] In these simulated data, there is a significant interaction between duration and orientation but this does not affect the dimensionality of the state-trace

latent variables (u and v) and the resulting state-trace is two-dimensional. This is shown in Figure 3.2(b). These data points were generated by the output mapping given by Equation (3.2) and the input mapping given by Equation (3.3) with weights, $\alpha = 0$, $\gamma = 0$, and $\beta = 1$, recalling that β is the weight corresponding to a stimulus by orientation interaction (see Equation (3.1)). The straight line corresponds to the best-fitting line of unit slope and conforms to the null-hypothesis of no interaction between stimulus and any other independent variable. The fact that the points in the state-trace depart from this line means that ANOVA will identify the appropriate stimulus $\times$ orientation interaction. In this case, the conclusions of ANOVA and STA (a two-dimensional state-trace) are in agreement.

However, as pointed out by Loftus et al. (2004), the problem in using ANOVA to test for deviations from a one-dimensional state-trace is that it depends critically on the assumption of a linear output mapping, as defined by Equation (3.2). If the actual mapping is not linear then ANOVA may detect a significant interaction even though the state-trace is one-dimensional. As discussed earlier in the context of the problem of nomic measurement, we have no reason to suppose that the output mapping is linear. Although researchers may go to great lengths to match the independent variables on as many features as possible, such as discriminability, salience, complexity etc., this does not solve the problem, even in principle. Thus, Equation (3.2) is replaced by the following,

$$g : \begin{cases} x = g_x(u) \\ y = g_y(u + v) \end{cases} \tag{3.5}$$

where $g_x(\cdot)$ and $g_y(\cdot)$ are (potentially distinct) monotonic functions. The first thing to note about this equation is that it does not alter the conclusion that if any of the interaction weights in the input function for v is non-zero then the state-trace corresponding to Equation (3.5) is still two-dimensional. On the other hand, if all the interaction weights are zero, Equation (3.5) reduces to,

$$g : \begin{cases} x = g_x(u) \\ y = g_y(u + \mu') \end{cases} \tag{3.6}$$

While this output mapping defines a one-dimensional state-trace, it will not, in general, be a straight line with unit slope. This is shown in Figure 3.2(c) where different monotonic functions (g_x and g_y) were applied to the corresponding coordinates of the data points shown in Figure 3.2(a). This results in a one-dimensional monotonic state-trace in DV space but not one that lies on a straight line of unit slope. As in Figure 3.2(b), the straight line in this graph is the best-fitting line of unit slope. But, because the data points depart from this line, ANOVA will find a significant interaction between the

stimulus factor and one or more of the other independent variables. In this case, the conclusions of ANOVA and STA are in disagreement.

The inference that is drawn from STA does not depend on whether the output mapping is linear or not – only that it is monotonic. In contrast, the inference drawn from ANOVA critically depends on the assumption of a linear output mapping. This is because ANOVA is sensitive to any deviation of the state-trace from the predicted straight line of unit slope. This is the same point that was discussed previously in Chapter 2. Although the state-trace in Figure 3.2(c) is not consistent with a specific linear additive model (i.e., no interactions involving the stimulus independent variable), it can be made so by applying appropriate (inverse) monotonic transformations to the coordinates of the DV space thereby re-creating Figure (3.2)(a). This kind of interaction was called *uninterpretable* by Loftus (1978) and *removable* by Wagenmakers et al. (2012). In contrast, the interaction shown in Figure 3.2(b) cannot be reduced to the additive model shown in Figure (3.2)(a). It therefore corresponds to an *interpretable* or *nonremovable* interaction.

3.3 Example

The previous two sections contrasted the logic of STA with that of functional dissociation and ANOVA. In this section, we illustrate the implications of this logic in the context of a real experiment. This is a recent study by Nakabayashi, Burton, Brandimonte and Lloyd-Jones (2012) that investigated the effects of verbal processing on memory for pictures of faces and buildings[3].

We focus our re-analysis on the first two experiments conducted by citetNBBL12 that were broadly similar in design to the face-inversion study conducted by Loftus et al. (2004). In these two experiments, the researchers examined the effects of two independent variables, item familiarity and type of study task, on two dependent variables, memory for faces (in Experiment 1) and memory for buildings (in Experiment 2). Three different study tasks were compared: In the *control* (Con) task participants were asked to tap the desk at a rate of 3-4 times per second, in the *articulatory suppression* (AS) task participants were asked to repeat the word 'la' at the same rate as they had tapped in the control task, and in the *description* (Des) task participants were asked to verbally describe the presented stimulus without tapping. The aim of the study was to examine the role of verbal processing in supporting memory of faces and other objects. As in other studies of the face-inversion effect, there was theoretical interest in whether faces are processed in a qualitatively different way to other objects. In other words, whether memory for pictures of faces and buildings depended on more than one latent variable.

[3] We wish to thank Kayuzo Nakabayashi and his colleagues for generously sharing their data with us.

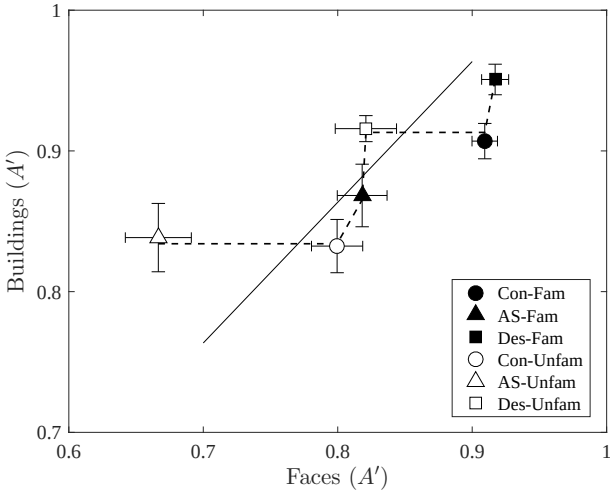

Fig. 3.3 State-trace plot of results from (Nakabayashi et al., 2012, Experiments 1 and 2). Recognition memory performance for faces and buildings was measured using the A' statistic. The six conditions of the experiment are defined as follows: *Con-Fam* – Control task/Familiar stimulus; *AS-Fam* – Articulatory suppression/Familiar stimulus; *Des-Fam* – Verbal description/ Familiar stimulus; *Con-Unfam* – Control task/Unfamiliar stimulus; *AS-Unfam* – Articulatory suppression/Unfamiliar stimulus; *Des-Unfam* – Verbal description/Unfamiliar stimulus. Error bars correspond to standard errors. The dashed line shows the best-fitting monotonic model. The solid line shows the best-fitting additive model.

Figure 3.3 is the state-trace plot based on the combined results of Experiments 1 and 2. Memory for faces and buildings was measured using the A' statistic, a commonly used measure of discriminability between targets and lures that attempts to control for decision bias (Pollack & Norman, 1964). Superimposed on these data are the best-fitting monotonic model based on STA (dashed line – we present details of how this was done in Section 4) and the best-fitting linear model based on ANOVA (solid line). From the perspective of STA, it is apparent that the monotonic model fits the data very well, leading to the inference that there is little evidence to reject the hypothesis of a single latent variable. From the perspectives of functional dissociation and the general linear model, the data reveal several dissociations and interactions which suggest the exact opposite – that the dependent variables are functions of several different latent variables.

As noted by the authors, the results reveal a double dissociation between unfamiliar faces and buildings. First, in comparison to the control task, articulatory suppression impaired memory for unfamiliar faces but had no effect on memory for unfamiliar buildings. This dissociation corresponds to the horizontal line segment in Figure 3.3 that connects the unfilled triangle (*AS-Unfam*) with the unfilled circle (*Con-Unfam*). Second, overt verbal description improved memory for unfamiliar buildings but had little

effect on memory for unfamiliar faces. This dissociation corresponds to the near vertical line segment in Figure 3.3 that connects the unfilled circle (*Con-Unfam*) with the unfilled square (*Des-Unfam*). A slightly different picture emerges for familiar stimuli. In this case, while verbal description again uniquely enhanced memory for buildings (corresponding to the vertical alignment of the *Con-Fam* and *Des-Fam* conditions in Figure 3.1), articulatory suppression impaired memory for both faces and buildings (*AS-Fam* is displaced diagonally from *Con-Fam* in Figure 3.3). Yet, because this set of dissociations is consistent with a monotonically increasing state-trace, they offer no grounds for inferring the existence of more than one latent variable.

Similar conclusions apply to the observation of significant interaction terms. When the data from the two experiments were submitted to a combined ANOVA, the authors found significant interactions between both stimulus type and study task and between stimulus type and familiarity. As specified by Equation (3.4), this is necessarily a straight line with positive unit slope in the state-trace plot. It is apparent from Figure 3.3 that this model fits the data poorly, consistent with the two statistically significant interactions. However, because the data are consistent with a monotonically increasing state-trace, they do not offer grounds for inferring the existence of more than one latent variable.

In summary, when analyzed using STA, the results found by Nakabayashi et al. (2012) are consistent with a model in which the effects of the independent variables on the two dependent variables are mediated by a single latent variable. In other words, there is no evidence that faces are processed in a qualitatively different way to other objects.

Part II
Application

Chapter 4
Statistical methodology

4.1 Introduction

Until recently, the application of state-trace analysis had been limited by the lack of an appropriate statistical treatment to test hypotheses in the face of measurement error. The approach we describe in this section is based on the null hypothesis testing procedure proposed by Kalish et al. (2016). In Chapter 8, we discuss an alternative Bayesian model selection procedure proposed by Prince et al. (2012a).

We introduce the statistical approach to STA in four parts. First, we discuss fitting a linear model. This is a conceptually simpler, or at least more familiar, case which helps to motivate what is to follow. Second, we discuss a procedure for fitting a monotonic model based on *monotonic regression*. Third, we apply this procedure to observed data involving sample means and variance, and fourth, we describe a hypothesis testing procedure based on bootstrap re-sampling.

4.2 Fitting a linear model

We motivate the statistical analysis of STA data by first describing how to fit a state-trace using a linear model. In Chapter 2, we briefly discussed such a model and explored it in greater detail in Chapter 3. Following from this, a linear one-dimensional output mapping, g, has the form,

$$g : \begin{cases} x = \alpha_x u + \beta_x \\ y = \alpha_y u + \beta_y \end{cases} \tag{4.1}$$

where u is the (unknown) latent variable. The observed values of the dependent variables, x' and y', contain measurement error. That is,

$$x' = x + \varepsilon_x$$
$$y' = y + \varepsilon_y$$

Note that this differs from ordinary least squares regression where one of the dependent variables, say x, is assumed to be measured without error. That is, $x' = x$. To fit this linear model to data, values of the parameters, u, α_x, α_y, β_x, and β_y are sought that minimize a nominated goodness-of-fit or error function.

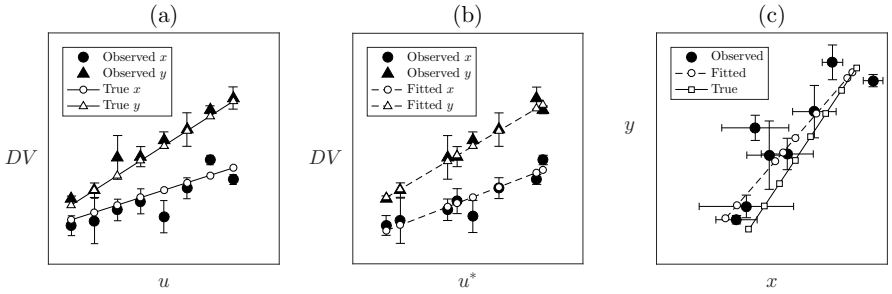

Fig. 4.1 Simulated data derived from the one-dimensional linear output mapping in Equation (4.1). (a) Open circles and triangles show true values of the dependent variables x and y as linear functions of the true values of the latent variable, u. Filled circles and triangles show observed values of x' and y', derived from the true values by adding a random error component. Error bars show standard errors. (b) Open circles and triangles show best-fitting values of x' and y' as linear functions of u^*, the best-fitting values of the latent variable, u. Filled circles and triangles show observed values of x' and y'. (c) Filled circles and error bars show the observed state-trace. Open circles show the model state-trace. Open squares show the true state-trace.

Figure 4.1 illustrates fitting a linear model to simulated data generated from parameters, $\alpha_x = 1$, $\alpha_y = 2$, $\beta_x = 0$, and $\beta_y = 1$. Figure 4.1(a) plots x and y as functions of eight different values of u corresponding to eight different conditions of an experiment. These points are shown by open circles and triangles and are labelled as 'True x' and 'True y', respectively. The filled circles and triangles show the values of x' and y' and are labelled as 'Observed x' and 'Observed y', respectively. Each of these points is shown with an accompanying error bar corresponding to the standard error of the associated measurement error. In this simulation, $\varepsilon_x \sim \mathcal{N}(0, \sigma_x^2)$ and $\varepsilon_y \sim \mathcal{N}(0, \sigma_y^2)$, where σ_x^2 and σ_x^2 may vary from condition to condition. That is, we do not assume that the data are homoscedastic.

Let u^*, α_x^*, α_y^*, β_x^* and β_y^* be the best-fitting parameter values and let,

$$x^* = \alpha_x^* u^* + \beta_x^*$$
$$y^* = \alpha_y^* u^* + \beta_y^*$$

be the predicted values of x' and y', respectively. These parameter values are chose to minimize the following weighted least squares error function,

$$F = \sum_i v_i(x'_i - x^*_i)^2 + \sum_i w_i(y'_i - y^*_i)^2 \tag{4.2}$$

where v and w are vectors of weights given by the inverse of the squared standard error in each condition for dependent variables, x' and y', respectively.

A point in passing: The linear model given by Equation (4.1) is not identifiable which means that the parameters must be appropriately constrained. One option is to set α to one and β to zero for one of the dependent variables. This approach is most often used in *orthogonal regression* where one of the dependent variables is conceptualized as a predictor of the other (Carroll, Ruppert, Stefanski & Crainiceanu, 2006). Alternatively, constraints may be placed on the latent variable, u. We take this approach because we do not distinguish between the dependent variables as both are viewed as functions of the same latent variable. Accordingly, we constrain u^* to have a mean of zero and variance of one[1].

Figure 4.1(b) plots the predicted data, x^* and y^*, and the observed data, x' and y', as functions of u^*. Careful comparison of the observed data points between Figure 4.1(a) and Figure 4.1(b) shows that their locations are not the same. This is due to the fact that u^* is a noisy estimate of u. A useful way of conceptualizing the model fitting process is to imagine the set, U, of all possible 8-element vectors (because there are eight conditions) constrained to have a mean of zero and variance of one. Then one way to find u^* is to draw elements from this set, one by one. Let $u' \in U$ be such an element. Then, x' and y' are regressed on u' to obtain the best-fitting estimates of the linear coefficients for each dependent variable and the overall goodness-of-fit, F, is calculated. The procedure terminates when the minimum of F is found in which case the corresponding u' is u^*. Naturally, the actual search is undertaken in a much more efficient manner than this. However, thinking about the procedure in this way helps to understand how a monotonic output mapping is fit to the same data.

Figure 4.1(c) shows the resulting state-trace plot. In fact, it shows three state-traces in the one plot. First, the filled circles and accompanying error bars represent the observed data and associated uncertainty on both dependent variables. To distinguish it from the other state-traces, we call this the *observed* state-trace. Second, the open circles represent the state-trace defined by the best-fitting model parameters. Again, in order to distinguish it, we call this the *model* or, in this case, the *one-dimensional linear* state-trace. Lastly, the open squares represent the state-trace defined by the true model parameters. This is called the *true* state-trace although, outside of a simulation, it is never observed.

Figure 4.1(c) demonstrates that the model state-trace is not the same as the true state-trace and neither corresponds to the observed state-trace,

[1] There is nothing special about these constraints on the location and scale of u – other choices could equally be made.

given non-zero measurement error. The model state-trace shares one crucial property with the true state-trace – namely, they are both linear. This means that if the model state-trace is sufficiently distant from the observed state-trace (in some defined statistical sense) then it may be concluded that the underlying output mapping is either not one-dimensional or not linear. If on the other hand, the model state-trace is not sufficiently distant from the observed state-trace (i.e., it provides an acceptable fit) then it may be concluded that there is at least one one-dimensional linear output mapping that does so. It does not necessarily follow that the model state-trace need be particularly similar to the true state-trace, other than sharing the property of being linear. We make this point because in the next section we will apply the same logic to a one-dimensional *monotonic* output mapping. In this case, the model state-trace need not be similar to any plausible true state-trace. The only property they share is that they are both monotonic.

4.3 Fitting a monotonic model

In this section, we describe a procedure to fit a monotonic model. As discussed in Section 4.2, when a linear model is fit to data, the predicted values of each dependent variable are linear functions of the *values* of the latent variable. In contrast, when a monotonic model is fit to data, the predicted values of each dependent variable are monotonic functions of the *order* of the values of the latent variable. With these differences in mind, the procedure to fit a monotonic model can be similarly conceptualized as involving the following steps:

1. Let K be the set of all possible orders of k conditions. Ignoring ties for the moment, there are $k!$ elements in this set.
2. Select an element of K and find best-fitting values, x^* and y^* for the observed x' and y' using *conjoint monotonic regression* (described below). Calculate the weighted least squares goodness-of-fit, F, in the same way as for the linear model.
3. Find the element of K such that the corresponding F is a minimum. The corresponding predicted values, x^* and y^*, constitute the best-fitting monotonic model state-trace.

In practice, this exhaustive search procedure is highly inefficient but fortunately there is a more efficient algorithm to do this (Burdakov, Dunn & Kalish, 2012). In the following sections, we briefly outline three key aspects of this algorithm: the concepts of partial and total orders, monotonic regression, and conjoint monotonic regression.

Partial and total orders

Fitting a linear model involves linear regression; fitting a monotonic model involves monotonic regression. In linear regression, the predicted values are constrained to be a linear function of the predictor variable. In monotonic regression, the predicted values are constrained to have the same order as the predicted variable, or more generally, to have some specified order. Importantly, this order need only be partial. A concrete example may help. Suppose $y = (3.6, 1.4, 2.7)$ are the values of a dependent variable which we want to predict using monotonic regression. That is, we want to find a vector of predicted values, y^*, that are as close as possible to y in a least squares sense but also satisfy a nominated partial order. Suppose this partial order requires that $y_1^* \leq y_2^*$. Then, because $y_1 > y_2$ prediction cannot be perfect. Indeed, the best-fitting predicted values are, $y^* = (2.5, 2.5, 2.7)$.

A partial order is a binary relation on a set that is consistent with our intuitions of inequality. It can be conveniently represented as a *directed acyclic graph* (or *DAG*). A DAG, $G(V, E)$, is defined as a set of *vertices*, V, and a set of directed *edges*, E, that connect pairs of vertices such that there are no cycles – that is, it is not possible to start at one vertex and, by following a sequence of edges, return to that vertex. Figure 4.2 shows the DAG representing the partial order on y^* mentioned above. In this case, $V = \{1, 2, 3\}$ and $E = \{(1, 2)\}$. The directed edge connecting the vertices, 1 and 2, corresponds to the requirement that $y_1^* \leq y_2^*$.

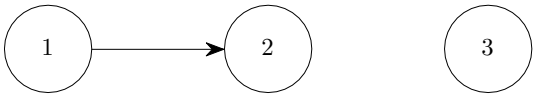

Fig. 4.2 A directed acyclic graph corresponding to the partial order on y^*.

In general, let I be the set, $\{1, \ldots, k\}$. We define a partial order on I as a set of edges E of the DAG, $G(I, E)$, and we say that a k-vector, x, is *consistent* with E if $x_i \leq x_j$, for all $(i, j) \in E$. If $E = \{(i, j)\}$ for all $i, j \in I$ then E is called a *total* or *linear* order. Let E and E' be two partial orders on I. If $E' \supseteq E$ then E' is called an *extension* of E. If E' is also a total order then it is called a *linear extension* of E.

Finally, let x be a vector of length k. The *order* of x, $O(x)$, is defined as,

$$O(x) \equiv \{(i, j) | x_i \leq x_j\}$$

It is clear that $O(x)$ is a total order. Let E be a partial order defined on $I = \{1, \ldots, k\}$. Then x is consistent with E if and only if $O(x) \supseteq E$. Let y be a vector of length k. If there exists a total order, T on I, such that $O(x) \supseteq T$ and $O(y) \supseteq T$ then we say $O(x)$ is congruent with $O(y)$ or $O(x) \cong O(y)$. In other words, x and y are monotonically related.

Monotonic Regression

Monotonic regression (MR) is a known problem with an extensive literature. Given a vector of values, x, indexed by I, a vector of weights, v, and a partial order E defined on I, the MR problem is to find a vector, x^* consistent with E, that minimizes the following function,

$$F = \sum_i v_i (x_i - x_i^*)^2 \tag{4.3}$$

We write the problem of finding x^* given x, v, and E as $\mathrm{MR}(x, v, E)$ and the corresponding minimum value of F as $\omega(x, v, E)$, or, in shorthand form, as ω_x. If E is empty then $x^* = x$ and $\omega_x = 0$. Finding the solution to the MR problem for a non-empty E is not trivial, but fast algorithms have been developed. If E is a total order then the MR problem can be solved using the *pool-adjacent-violators algorithm* (PAVA), a version of which was used in the original development of non-metric multidimensional scaling (Kruskal, 1964). Otherwise, the problem can be solved using various approaches (Block, Qian & Sampson, 1994; Burdakov, Sysoev, Grimvall & Hussian, 2006; Luss, Rosset & Shahar, 2010; Maxwell & Muchstadt, 1983; Roundy, 1986; Stout, 2013) including quadratic programming algorithms (de Leeuw, Hornik & Mair, 2009).

Conjoint monotonic regression

Conjoint monotonic regression (CMR) is an extension of monotonic regression to two or more variables which are required to be monotonically related. This problem was first formulated and solved by Burdakov et al. (2012). The CMR problem can be defined as follows: Given two vectors of values, x and y, associated vectors of weights, v and w, and a partial order, E, we wish to find a linear extension, $E^* \supseteq E$, and vectors, x^* and y^* consistent with E^*, that minimize the following function,

$$F = \sum_i v_i (x_i - x_i^*)^2 + \sum_i w_i (y_i - y_i^*)^2 \tag{4.4}$$

Note that in this form, Equation (4.4) is identical to Equation (4.2) with the vectors, x' and y', in that equation written here as x and y. In both cases, they refer to the observed values of the dependent variables. We write the problem of solving Equation (4.4) as $\mathrm{CMR}(x, y, v, w, E)$, shorthand $\mathrm{CMR}(E)$, and the minimum value as $\omega(x, y, v, w, E)$, shorthand ω_{xy}.

In contrast to the MR problem, solving the CMR problem is non-trivial even when E is empty. This is because E^* is required to be a total order. In fact, the problem is maximally difficult in this case. Because E^* is a linear

extension of E, the closer E is to being a total order, the smaller the search set for E^*. In the limit, if E is a total order then $E^* = E$ and the CMR problem reduces to the two MR problems, MR(x, v, E) and MR(y, w, E).

Figure 4.3 illustrates the results of fitting a monotonic model using conjoint monotonic regression to the same observed vectors and weights as used in Section 4.2. In this case, E is empty. Figure 4.3(a) and Figure 4.3(b) correspond to Figure 4.1(b) and Figure 4.1(c), respectively. In fitting the linear model, the observed variables, x' and y', were regressed on the values of common latent variable, u^*, using (weighted) linear regression. In fitting the monotonic model, the observed variables are regressed on the best-fitting total order, E^*, using conjoint monotonic regression. The results are shown in Figure 4.3(a) as a function of a vector, $u*$, with arbitrary values consistent with E^* (these are just the numbers 1 to 8). The corresponding model state-trace is shown in Figure 4.3(b) which shows that the set of fitted points is non-decreasing in DV space.

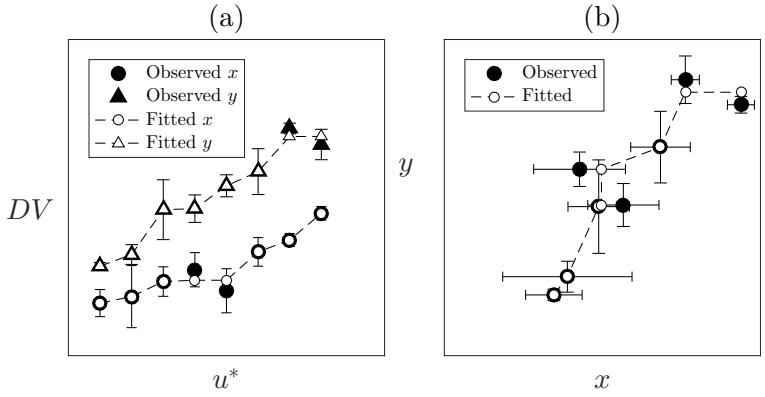

Fig. 4.3 One-dimensional monotonic model fit to simulated data derived from the one-dimensional linear output mapping in Equation (4.1). (a) Open circles and triangles show best-fitting values of x' and y' as monotonic functions of $u*$, a vector consistent with the best-fitting total order, E^*. Filled circles and triangles show observed values of x' and y'. (c) Filled circles and error bars show the observed state-trace. Open circles show the model state-trace.

A note in passing: The fitted points in Figure 4.3 are shown connected by dashed lines. In contrast to the linear model where these lines correspond to (meaningful) linear interpolations between the points, this is not true for the monotonic model. The purpose of these lines is purely for display – to facilitate locating the points in the graph and identifying that they are monotonically ordered.

While direct search is guaranteed to find a solution, it is obviously inefficient, requiring exhaustive evaluation of a potentially very large number of total orders. For example, if $k = 10$ and E is empty, there are $k! =$

3,628,800 total orders to search. To circumvent this difficulty, Burdakov et al. (2012) devised the *CMR algorithm* that finds a solution in approximately exponential rather than factorial time.

Rather than searching orders directly, it uses the fact that the solutions to the CMR problem, x^* and y^*, must themselves be monotonically related. That is,

$$\begin{aligned} x_i^* < x_j^* &\Rightarrow y_i^* \leq y_j^* \\ y_i^* < y_j^* &\Rightarrow x_i^* \leq x_j^* \end{aligned} \tag{4.5}$$

This constraint can also be succinctly expressed by the requirement that there are no $i, j \in I$ such that, $(x_i^* - x_j^*)(y_i^* - y_j^*) < 0$.

The CMR algorithm is a branch-and-bound algorithm which, given an initial partial order, E, progressively adds additional partial order constraints until an optimal solution is reached. On each iteration, a new extension, $E' \supset E$, is considered. Let x' and y' be the solutions to $\mathrm{MR}(x, v, E')$ and $\mathrm{MR}(y, w, E')$, respectively. If they violate the conjoint monotonicity constraint given by Equation (4.5) then E' is said to be *infeasible* and the algorithm selects a pair of indices, $(i, j) \notin E'$, and generates two new extensions, $E' \cup \{(i, j)\}$ and $E' \cup \{(j, i)\}$, one of which must be a subset of the solution, E^*. If, during this process, a feasible E' is found then the corresponding fit value provides an upper bound for the optimal fit, ω_{xy}. If the fit of any infeasible E' is greater than the current upper bound then E', along with all its extensions, can be eliminated from the search. The algorithm continues branching and eliminating until there are no more infeasible partial orders to be tested or until the fit of the best-fitting infeasible partial order is greater than the current upper bound. In both cases, the current upper bound is the value of the optimal least-squares solution, ω_{xy}. Simulations confirm that the CMR algorithm converges to an optimal solution in exponential time rather than factorial time as entailed by an exhaustive search. By way of illustration, for $k = 10$, uncorrelated x and y, and E empty, the CMR algorithm evaluates on average about 25 sub-problems before finding a solution whereas an exhaustive search must evaluate over three million sub-problems. Furthermore, if a non-empty E is specified, this reduces the number of extensions to be tested resulting in a further reduction in computation time.

4.4 Fitting a monotonic model to data

In this section, we discuss how to fit a monotonic model to a data set consisting of the set of observations on $n > 1$ dependent variables under each of k conditions. To keep the exposition simple, we explicitly treat the $n = 2$ case although the results are easily generalized to more than two

dependent variables. We discuss three main kinds of data types; independent observations on continuous variables, related observations on continuous variables, and independent observations on binary variables.

Independent observations on continuous variables

This corresponds to a classic 'between-participants' or 'randomized' design with continuous dependent variables where each observation is considered to be independent of every other observation both within and between conditions. Each observed value on each dependent variable is viewed as a random draw from an underlying continuous distribution. While this is often assumed to be the normal distribution, this is not required in the present approach.

We use the following terminology. Let $X = (X_1, \ldots, X_k)$ and $Y = (Y_1, \ldots, Y_k)$ be vectors consisting of the sample means in each of k conditions for two dependent variables, X and Y. We reserve lowercase x and y for the corresponding population means. Let $S^2_X = (S^2_{X,1}, \ldots, S^2_{X,k})$ and $S^2_Y = (S^2_{Y,1}, \ldots, S^2_{Y,k})$ be the corresponding sample variances in each condition for each dependent variable and let $N_X = (N_{X,1}, \ldots, N_{X,k})$ and $N_Y = (N_{Y,1}, \ldots, N_{Y,k})$ be the number of observations in each condition for each dependent variable. In terms of Equation (4.4), the observed vectors, x and y, correspond to the respective sample means, X and Y, and the weight vectors, v and w, correspond to the *precision* of these means. That is, $v = (N_{X,1}/S^2_{X,1}, \ldots, N_{X,k}/S^2_{X,k})$ and $w = (N_{Y,1}/S^2_{Y,1}, \ldots, N_{Y,k}/S^2_{Y,k})$. The precision is a natural choice for the weight of each mean and it can be readily shown that if the distribution of each mean is normal then the solution to Equation (4.4) is also the maximum likelihood solution (Kalish et al., 2016).

The corresponding CMR problem is to find a linear extension, $E^* \supseteq E$, and vectors, X^* and Y^* consistent with E^*, that minimize the function,

$$F = \sum_{i=1}^{k} v_i (X_i - X_i^*)^2 + \sum_{i=1}^{k} w_i (Y_i - Y_i^*)^2$$

We can now say something about E, the initial partial order. This is a relatively unfamiliar concept but one that falls out naturally from conjoint monotonic regression[2]. In a given experiment, E represents a set of order constraints on the dependent variables that is informed by the nature of the experimental conditions. For example, in a category learning experiment, X and Y may correspond to categorization performance on two different category structures. One of the independent variables may correspond to blocks of trials: 1, 2, and so. Suppose that condition 1 corresponds to the

[2] Prince et al. (2012a) provide an extensive treatment of partial orders in their Bayesian hypothesis testing approach to STA.

first block of trials and condition 2 corresponds to the second block of trials. Then, given that performance is not expected to decrease with learning, it follows that $X_1 \leq X_2$ and $Y_1 \leq Y_2$. This requirement could then be included in E which would force $X_1^* \leq X_2^*$ and $Y_1^* \leq Y_2^*$. We discuss the implications of this idea more fully in Section 4.5.

Related observations on continuous variables

This corresponds to a 'repeated measures' or 'within-participants' design with continuous dependent variables. In this case, across a subset of $r \leq k$ conditions, some or all of the observations are correlated or dependent. Typically, this is because they are made on the same participant. If all k conditions have this feature then the design is described as (fully) repeated measures or within-participants, otherwise it is often described as 'mixed'. For ease of explication, we assume that the design is fully within. The data correspond to two $n \times k$ matrices of observations where each row corresponds to one participant and each column to one condition.

As described by Kalish et al. (2016), dependent observations across conditions requires that the weight vectors, v and w, be replaced by matrices, $\mathbf{V}$ and $\mathbf{W}$. This captures the fact that the covariance matrix for each dependent variable is not diagonal. Let $\mathbf{S}_X$ be the $k \times k$ sample covariance matrix of X and let $\mathbf{S}_Y$ be the sample covariance matrix of Y. Let $N_{X(i,j)}$ and $N_{Y(i,j)}$ be the numbers of participants observed under both conditions i and j on X and Y, respectively. We define the matrices, $\mathbf{N}_X = [N_{X(i,j)}]_{ij}$ and $\mathbf{N}_Y = [N_{Y(i,j)}]_{ij}$. Then the weight matrices, $\mathbf{V}$ and $\mathbf{W}$ are defined as follows,

$$\mathbf{V} = \mathbf{N}_X \mathbf{S}_X^{-1}$$
$$\mathbf{W} = \mathbf{N}_Y \mathbf{S}_Y^{-1}$$

The corresponding CMR problem is to find a linear extension, $E^* \supseteq E$, and vectors, X^* and Y^* consistent with E^*, that minimize the function,

$$F = (X - X^*)^T \mathbf{V} (X - X^*) + (Y - Y^*)^T \mathbf{W} (Y - Y^*)$$

Independent observations on binary variables

These data arise when the dependent variables are counts of 'successes' and 'failures' across the set of k conditions. Let H_X and H_Y be vectors of the number of successes (or hits) in each condition for variables X and Y, respectively. Similarly, let F_X and F_Y be vectors of the corresponding

number of failures (or misses), and let $N_X = H_X + F_X$ and $N_Y = H_Y + F_Y$. Let P and Q be the proportions of hits for each variable. That is,

$$P_i = H_{X,i}/N_{X,i}$$
$$Q_i = H_{Y,i}/N_{Y,i}$$

Then the corresponding CMR problem is to find a linear extension, $E^* \supseteq E$, and vectors, P^* and Q^* consistent with E^*, that minimize the function,

$$F = \sum_{i=1}^{k} N_{X,i}(P_i - P_i^*)^2 + \sum_{i=1}^{k} N_{Y,i}(Q_i - Q_i^*)^2$$

Here the weight vectors, v and w, are equal to the number of observations N_X and N_Y. The reason for this is that it can be shown that the solution to this CMR problem is also the maximum likelihood solution under the assumption that each P_i and Q_i is a random draw from a binomial distribution (Kalish et al., 2016; Robertson, Wright & Dykstra, 1988).

4.5 Hypothesis test

While the CMR algorithm returns a goodness-of-fit (or rather badness-of-fit) value, ω_{XY}, the problem remains of determining whether this value is large enough to reject the null hypothesis that the population means are monotonically related. Specifically, we are interested in two distinct hypotheses, corresponding to the monotonic and conjoint monotonic regression problems, respectively. We define each of these in terms of the orders, $O(x)$ and $O(y)$, on the population means, x and y, respectively.

We define the following two models with respect to a partial order, E.

1. **Partial order model:**

   $$M_1 : O(x), O(y) \supseteq E$$

   This states that the orders of x and y are both consistent with E. The fit of this model to sample means, X, Y, and weights, v, w, is given by the sum of the fits of the appropriate monotonic regression problems. That is, $\omega_{X+Y} = \omega(X, v, E) + \omega(Y, w, E)$.

2. **Monotonic model:**

   $$M_2 : O(x), O(y) \supseteq E \;\; \& \;\; O(x) \cong O(y)$$

   This states that the orders of x and y are both consistent with E and monotonically related to each other. The fit of this model to sample

means, X and Y, is given by the fit of the solution to the appropriate conjoint monotonic regression problem. That is, $\omega_{XY} = \omega(X, Y, v, w, E)$.

We now consider a test of the fit of the partial order model, M_1, for E not empty, and a test of the difference in fit between the monotonic model, M_2, and the partial order model, M_1. This directly tests the hypothesis that $O(x) \cong O(y)$.

At present the relevant test statistics for each of these model comparisons is unknown. Concerning M_1, some work has been done on developing a test of the individual hypothesis, $O(x) \supseteq E$, against an unconstrained alternative based on the sampling distribution of $\omega(X, v, E)$. It is known that under this hypothesis, the test statistic follows a $\bar{\chi}^2$ (chi-bar squared) distribution (Robertson et al., 1988). This is a mixture of χ^2 distributions of different degrees of freedom with mixture weights, called level probabilities, that depend in complex ways on the number of conditions, the number of participants, and the partial order, E. As a result, $\bar{\chi}^2$ distributions have only been calculated for a few, relatively simple, cases. While it may be possible to extend this approach to conjoint monotonic regression, we have not attempted this, as it seems likely that calculation of the theoretical distribution would encounter even greater difficulties. Instead, we approach this problem by testing each hypothesis using a bootstrap sampling procedure derived from the double bootstrap method described by Wagenmakers et al. (2004).

A test of the partial order model

The partial order model tests whether the population means of variables X and Y are both ordered in a specified way. As noted by Kalish et al. (2016), a partial order is relevant for two reasons. First, if there are good reasons for believing it to be true then it eliminates error associated with violations of monotonicity between the dependent variables. Second, if the partial order model is true, the statistical power of the test of the difference in fit between the monotonic and partial order models is increased.

Let $\mathbf{X}$ and $\mathbf{Y}$ be two data sets each consisting of N_X and N_Y observations, respectively, under each of k conditions. Let X_{ij} be the ith observation in the jth condition in $\mathbf{X}$ and similarly Y_{ij} for $\mathbf{Y}$. Let X and Y be the corresponding vectors of sample means, let v and w be the corresponding precisions of these means, and let E be a specified, non-empty, partial order. Recall, that if E is empty then the partial order model fits trivially (i.e., $\omega_{X+Y} = 0$). Although we refer below to the weights as v and w, respectively, for a mixed design, these are understood to correspond to the weight matrices, $\mathbf{V}$ and $\mathbf{W}$. The steps in the test of the partial order model are as follows:

1. Solve the problems, $\mathrm{MR}(X, v, E)$ and $\mathrm{MR}(Y, w, E)$, and calculate the sum of optimal fits, $\omega_{X+Y} = \omega(X, v, E) + \omega(Y, w, E)$.

2. Generate two non-parametric bootstrap samples, $\mathbf{X}_B$ and $\mathbf{Y}_B$, from $\mathbf{X}$ and $\mathbf{Y}$, respectively. That is, each of the $N_{X,j}$ observations in the jth condition of $\mathbf{X}_B$ is a sample drawn with replacement from the $N_{X,j}$ observations in the corresponding condition of $\mathbf{X}$. Similarly for $\mathbf{Y}_B$ and $\mathbf{Y}$. Calculate the corresponding bootstrap sample means, X_B and Y_B, and corresponding precisions, v_B and w_B. If some conditions are repeated across participants then participants rather than observations are sampled.

3. Solve the problems, $\mathrm{MR}(X_B, v_B, E)$ and $\mathrm{MR}(Y_B, w_B, E)$, finding the best-fitting values, X^*_B and Y^*_B. Then, for each observation, $X_{ij} \in \mathbf{X}$, and each observation, $Y_{ij} \in \mathbf{Y}$, create the transformed observation, $X'_{ij} = X_{ij} - X_j + X^*_{B,j}$, and similarly, $Y'_{ij} = Y_{ij} - Y_j + Y^*_{B,j}$. The resulting sample means of the transformed data sets are necessarily consistent with the partial order, E.

4. Generate two non-parametric bootstrap samples, $\mathbf{X}'_B$ and $\mathbf{Y}'_B$, from the transformed data sets, $\mathbf{X}'$ and $\mathbf{Y}'$, respectively. Calculate the corresponding sample means, X'_B, Y'_B, and corresponding precisions, v'_B, w'_B. Again, if some conditions are repeated across participants then participants rather than observations are sampled.

5. Solve the problems, $\mathrm{MR}(X'_B, v'_B, E)$ and $\mathrm{MR}(Y'_B, w'_B, E)$ and calculate the fit of M_1 to these bootstrap data, $\omega_{X'_B+Y'_B}$.

6. Repeat Steps 2-5 N times where N is a sufficiently large number (e.g., 10,000).

7. Calculate, p, the proportion of values of $\omega_{X'_B+Y'_B}$ that are greater than or equal to ω_{X+Y}. If $p < \alpha$ then reject the null hypothesis that M_1 is true.

A test of the monotonic model

The following describes a test of the difference in fits between the partial order model (M_1) and the monotonic model (M_2). That is, the difference between ω_{X+Y} and ω_{XY}. Note that if E is empty then $\omega_{X+Y} = 0$ necessarily. The steps in the test of the monotonic model are as follows:

1. Solve the two MR problems and find the fit of M_1, $\omega_{X+Y} = \omega(X, v, E) + \omega(Y, w, E)$. Solve the CMR problem and find the fit of M_2, $\omega_{XY} =$

$\omega(X, Y, v, w, E)$. Calculate the observed difference, $\delta = \omega_{XY} - \omega_{X+Y}$.

2. Generate two non-parametric bootstrap samples, $\mathbf{X}_B$ and $\mathbf{Y}_B$, from $\mathbf{X}$ and $\mathbf{Y}$, respectively. Calculate the corresponding bootstrap sample means, X_B and Y_B, and corresponding precisions, v_B and w_B.

3. Solve the CMR problem for each of the bootstrap samples and find the best-fitting values, X_B^* and Y_B^*. Then, for each observation, $X_{ij} \in \mathbf{X}$, and each observation, $Y_{ij} \in \mathbf{Y}$, create the transformed observation, $X'_{ij} = X_{ij} - X_j + X_{B,j}$, and similarly, $Y'_{ij} = Y_{ij} - Y_j + Y_{B,j}$. The resulting sample means of the transformed data sets then conform exactly to the monotonic model.

4. From the transformed data sets, $\mathbf{X}'$ and $\mathbf{Y}'$, draw a second set of non-parametric bootstrap samples, $\mathbf{X}'_B$ and $\mathbf{Y}'_B$ and calculate the corresponding sample means, X'_B, Y'_B, and precisions, v'_B, w'_B.

5. Solve the two MR problems for the bootstrap samples and find the fit of M_1, $\omega_{X'_B+Y'_B} = \omega(X'_B, v'_B, E) + \omega(Y'_B, w'_B, E)$. Solve the corresponding CMR problem and find the fit of M_2, $\omega_{X'_BY'_B} = \omega(X'_B, Y'_B, v'_B, w'_B, E)$. Calculate and store the difference, $\delta'_B = \omega_{X'_BY'_B} - \omega_{X'_B+Y'_B}$.

6. Repeat Steps 2-5 N times where N is a sufficiently large number (e.g., 10,000).

7. Calculate, p, the proportion of values of δ'_B that are greater than or equal to δ. If $p < \alpha$ then reject the null hypothesis that $O(x) \cong O(y)$.

Step 3 in both of the above tests, in which the re-sampled distribution is shifted to conform to the model under test, assumes that the data are continuous. However, if the data are discrete (e.g., conforming to Bernoulli trials), this step requires modification. In this case, the model-consistent data sets, $\mathbf{X}'$ and $\mathbf{Y}'$, are derived from a parametric bootstrap of the observed data. For binomially distributed data, this is not a substantial concern because these distributions are given in their entirety by the data and hence parametric and non-parametric re-sampling are equivalent.

Chapter 5
Mixed designs with continuous dependent variables

5.1 Introduction

In this chapter we demonstrate the application of STA to mixed designs with continuous dependent variables using the STACMR software package that we have developed. As previously described in Chapter 4, mixed designs involve a combination of independent observations ('between-participants' or randomized factors) and related observations ('within-participants' or repeated measure factors) on continuous variables. We begin with the mixed design because of its generality – fully randomized and fully repeated measure designs are easily seen as special cases.

In this chapter, we consider two dependent variables, x and y, concerning which there is theoretical interest in whether the effects of different independent variables are mediated by one or more than one latent variable. We assume that for this question to arise, x and y are expected to be positively correlated across changes in most variables. For example, if x and y are measures of categorization performance then it is expected that both will increase with increased experience at the, respective, categorization tasks. This means that it will be unlikely that variation in a single independent variable will be sufficient to generate a non-monotonic state-trace (Dunn, 2008). For this reason, a more useful design is one in which the conditions are defined by the combination of levels of two or more independent variables. Table 5.1 illustrates a minimal design consisting of three conditions defined by the non-factorial combination of two levels (1, 2) of each of two independent variables (IV), A and B.

Designs need not be balanced nor fully crossed as the underlying factorial structure, if any, is irrelevant to STA. The model-fitting procedure we describe in Chapter 4 explicitly accommodates differences in both the variance-covariance and number of observations between conditions. However, at the present time, correlations between conditions on different dependent variables are ignored. For example, even if observations on both x and y are made on

Table 5.1 A minimal state-trace design

	IV	
Condition	*A*	*B*
1	1	1
2	1	2
3	2	1

the same group of participants and are thus correlated, the present procedure makes no explicit allowance for this – the dependent variables are treated as if they are independent. We acknowledge that this may reduce statistical power and discuss the issue further in Chapter 9.

We use data from the mixed factorial design employed by Dunn, Newell and Kalish (2012) as our examples in this chapter. Participants in this study completed one of two category-learning tasks over 4 blocks of training (a within-participant factor) under one of 2 different conditions (a between-participant factor). The dependent variables are the proportions correct for each of two tasks defined according to the category structure that participants learned. For the rule-based (RB) group, the category structure was defined by a simple rule. For the information-integration (II) group, the category structure was more complex and could not be defined by a simple rule (for details, see Dunn et al., 2012).

The experiment by Dunn et al. (2012) addressed the theoretical question of whether learning different category structures depended upon multiple systems. In terms of the general structure introduced in Chapter 1, this view implied the existence of multiple latent variables involved in the two kinds of learning. This in turn implied that it should be possible to observe a non-monotonic state-trace. The purpose of the experiment was to combine several different independent variables in order to see if any of these led to the expected non-monotonic state-trace. The data from one of these combination of independent variables will be used to illustrate the application of the STACMR software package we describe next.

5.2 Using the STACMR software package

We have implemented the analysis methods described in Chapter 4 in a suite of functions contained in the software package STACMR (`github.com/ michaelkalish/STA`). This is available in both a Matlab version (STACMR-matlab) and an R version (STACMR-R). Both of these rely on Java© routines to speed numerical computing. To run them, it is necessary to download and install Java from http://www.java.com. In this section, we describe the Matlab implementation of STACMR. A summary of the function calls is included in the Appendix.

Prior to running the STACMR package, the current STACMR Java runtime library must be made available to Matlab. To do this, download and unzip STACMR-matlab.zip, and place the folder `STACMR-matlab` and its subfolders on the Matlab path. Then run `staCMRsetup.m`. This will locate and install the current java runtime library. At the time of writing, this is `fxMR-0.3.35.jar`.

Input data structures

To begin, we assume that we have a set of data, organized as a *data structure* containing a set of observations on $n > 1$ dependent variables across three or more conditions. As mentioned earlier, the logic of STA does not place any requirements on the factorial structure of the conditions which are simply considered as enumerated. For example, if there are two independent variables, one with 2 levels, the other with 3 levels, then there will be a total of $2 \times 3 = 6$ conditions, numbered simply from 1 to 6. In a mixed design, the conditions are numbered by STACMR in the following way. If there are b between-participant conditions and w within-participant conditions then there are a total of $b \times w$ conditions ordered such that the first w conditions correspond to the set of within-participant conditions in the first between-participant condition, the next w conditions correspond to the set of within-participant conditions in the second between-participant condition, and so on.

At present, STACMR accepts two kinds of data structure:

1. **General format.** This structure is useful if the data are already in some kind of fixed column format. STACMR expects the data to be organised as a matrix in which each row corresponds to an observation unit (e.g., a participant) and each column is defined as follows:

   - column 1 = observation unit number (numeric; for identification only, not used directly)
   - column 2 = between-participant condition or group (numeric; if none, then all values in column 2 should be set to 1)
   - column 3 = dependent variable index (numeric; at least two values)
   - columns 4 to end = values of the dependent variable specified in column 3 for the between-participant condition specified in column 2 for each within-participant condition (if present)

2. **Cell array format.** In this format, the data are organised as a $b \times n$ *cell array* (a data structure specific to Matlab) where b is the number of between-participant conditions and n is the number of dependent variables. Each component of this cell array is an $N \times w$ matrix of observations where N is the number of observation units, such as participants (which

may vary across the b conditions and n dependent variables), and w is the number of within-subjects conditions (assumed to be fixed across all between-participant conditions and dependent variables).

The data from Dunn et al. (2012) are in general format, contained in the file `delay.dat`. There are $b = 2$ between-participant conditions, labelled `delay` and `no delay`, and $w = 4$ within-participant conditions corresponding to four sequential blocks of training trials. To load these data, use the following Matlab command:

```
>> delay = load('delay.dat');
```

Internally, STACMR converts data in general format to cell array format. To do this manually, use the function, `gen2cell.m`.

```
>> y = gen2cell(delay);
```

The user can verify that `y` is a 2-by-2 cell array.

The state-trace plot

With the data loaded, it is often a good idea to visualize it using a state-trace plot. STACMR provides the function `staPLOT` to generate the state-trace plot and also to show the predicted values of a specified model, as we will see below. To begin, we can just plot the data.

```
>> staPLOT(delay, 'groups', {1:4, 5:8}, 'labels', ...
   {'No delay', 'Delay'}, 'axislabels', {'RB','II'});
```

This command produces the plot shown in Figure 5.1.The `staPLOT` function takes one required and a several optional arguments. The `readme` file provided with the STACMR package explains all of the arguments and their forms. The call that produced Figure 5.1 used the following arguments.

- `delay` specifies the data and is the one required argument. The function accepts a data structure in either general or cell-array format, or in summary statistics form (i.e., the output of `staSTATS` described below). If the input is a data structure then `staPLOT` invokes `staSTATS` and accepts an optional covariance shrinkage parameter (also explained below). The plot includes error bars; see *Summary statistics* below.
- `'groups'` identifies the conditions to be distinguished in the plot with different markers. For the delay data, conditions 1 to 4 correspond to the *no delay* group and conditions 5 to 8 correspond to the `delay` group. This is passed to `staPLOT` by setting the value of `'groups'` to the cell array, {`1:4, 5:8`}. Up to four groups can be defined in this way.

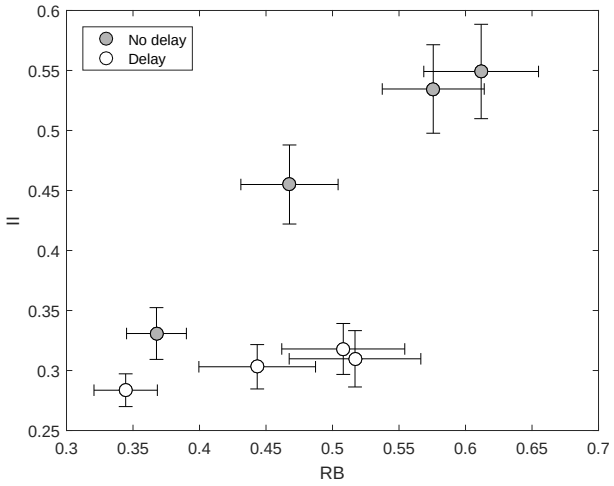

Fig. 5.1 Output from staPLOT based on delay data.

- `'labels'` is a cell array consisting of the labels for the groups defined by `'groups'` to appear in the legend. In the present example, this is specified by the cell array, {`'No delay','Delay'`}.
- `'axislabels'` is a cell array that defines the labels on the x-axis and y-axis, respectively (note: the same result can be achieved using the Matlab functions, `xlabel` and `ylabel`). In the present example, the labels are defined by the cell array, {`'RB','II'`}, for *rule based* and *information integration*, respectively.

A word of caution is in order here. The STACMR package assumes that the dependent variables are measured in such as way that the monotonic model applies – that is, the state-trace is expected to be one-dimensional and monotonically increasing in the dependent variables. If your plot appears to be decreasing, now is the time to reverse-code one of your outcomes.

Summary statistics

We mentioned above that `staPLOT` calls a provided function to compute the error bars that it displays. That function, `staSTATS` computes summary statistics of each of the dependent measures from a data structure in either general or cell array format. For the present data set:

```
>> delaystats = staSTATS(delay, shrink);
```

Input

staSTATS takes two input arguments: the name of the data structure, delay, and an optional parameter, shrink, that governs covariance shrinkage, required during bootstrap re-sampling. In a design containing two or more within-participant conditions, the bootstrap sample covariance matrix may be ill-conditioned and its inverse, if it can be calculated, is no longer a good estimate of the inverse of the population covariance matrix. STACMR deals with this in three (optional) ways, determined by the optional argument, shrink:

- shrink = 0. No shrinkage is applied and an adjusted covariance matrix is created which is equal to the sample covariance matrix.
- shrink = 1. Maximum shrinkage is applied and an adjusted covariance matrix is created which is equal to the diagonal of the observed covariance matrix with all off-diagonal entries set to zero.
- shrink < 0. An optimal shrinkage value (between zero and one) is estimated using the algorithm developed by Ledoit and Wolf (2004) and then applied to produce the adjusted covariance matrix. The diagonal of this matrix is equal to the diagonal of the sample covariance matrix while the off-diagonal values are replaced by smaller absolute values, as determined by the algorithm. This is the default.

Output

staSTATS, returns a cell array, here called delaystats, that has the same length as the number of dependent variables in the data structure. Each component of delaystats is a structured array containing statistics for the corresponding dependent variable, dvar. The following lists the most relevant (see readme for more details):

- delaystats{dvar}.means = vector of means.
- delaystats{dvar}.cov = the sample covariance matrix.
- delaystats{dvar}.shrinkage = a vector of length b (where b is the number of between-participant conditions) containing the specified or estimated *shrinkage* values as determined by the optional argument, shrink.
- delaystats{dvar}.regcov = the adjusted covariance matrix.
- delaystats{dvar}.n = a block diagonal matrix containing the number of observation units (participants) measured under each pair of conditions. Each block corresponds to a level of the between-participant factor. Within each block, the (i, j) entry is the number of participants who are observed under both levels i and j of the within-participant factor. This may not be the same for all i and j because of missing values.

- `delaystats{dvar}.weights` = matrix of weights corresponding to the precision of the data defined by, `delaystats{dvar}.n.*delaystats{dvar}.regcov^-1`.

Because the `delay` data structure contains two dependent variables, the cell array `delaystats` has two components. We list these below.

```
>> delaystats = staSTATS(delay);
>> disp(delaystats{1})
        means: [0.3676 0.4676 0.5757 0.6118
                0.3445 0.4434 0.5081 0.5169]
          cov: [8x8 double]
       regcov: [8x8 double]
            n: [8x8 double]
      weights: [8x8 double]
           lm: [8x8 double]
    shrinkage: [0.0520 0.0314]

>> disp(delaystats{2})
        means: [0.3308 0.4550 0.5346 0.5492
                0.2836 0.3031 0.3180 0.3098]
          cov: [8x8 double]
       regcov: [8x8 double]
            n: [8x8 double]
      weights: [8x8 double]
           lm: [8x8 double]
    shrinkage: [0.0491 0.2650]
```

Partial order

As discussed in Chapter 4, the observed means on each dependent variable may be required to conform to a given partial order. This serves to ensure that any misfit in the monotonic model is not due to observed, but theoretically uninteresting, non-monotonic effects of the independent variables on the latent variables. In the current `delay` experiment, participants are observed over four blocks of training. Fatigue effects are not of interest here, so we can restrict our attention to practice effects by placing a partial order constraint such that mean performance on each block is required to be no worse than performance on the previous block. By imposing this constraint, we shift some of the misfit of the monotonic model to the partial-order model, allowing a more sensitive measure of the underlying dimensionality. We will see this effect when we fit the models below. For now, we prepare by defining the appropriate partial order. This can be done in STACMR in two (equivalent) ways:

1. As a cell array consisting of a set of (i, j) pairs such that, for a vector x consistent with this set, $x_i \leq x_j$. An example might be:

   ```
   >> E = {[1 2], [2 3], [1 4]};
   ```

   Applied to a variable, x, this means that $x_1 \leq x_2$, $x_2 \leq x_3$, and $x_1 \leq x_4$. The first two components of this cell array specify a transitive order and van also be written in a shorthand form as {`1:3`}. That is,

   ```
   >> E = {1:3, [1 4]};
   ```

2. As an *adjacency matrix*, $[a_{ij}]$, where $a_{ij} = 1$ if $x_i \leq x_j$, otherwise $a_{ij} = 0$. For example, the partial order coded as the cell-array, `E`, above, is equivalent to the adjacency matrix, `A`, where:

   ```
   >> A =  [[0,1,0,1];[0,0,1,0];[0,0,0,0];[0,0,0,0]]
   A =
     0 1 0 1
     0 0 1 0
     0 0 0 0
     0 0 0 0
   ```

   The function, `cell2adj`, can be used to convert a partial order in cell array form into its corresponding adjacency matrix form. For example:

   ```
   >> A = cell2adj (1:4, E);
   ```

   In this call to `cell2adj`, the vector, `1:4`, specifies the index numbers of the conditions.

For the `delay` data, we wish to enforce a constraint requiring that performance does not decrease over each of the four training blocks in each between-participant condition (i.e., conditions 1 to 4 and conditions 5 to 8). That is,

```
>> E = {1:4,5:8};
```

In addition, it is also reasonable to presume that performance is better in the `no-delay` than in the `delay` condition at every block of training. We can add this constraint as follows:

```
>> E = {1:4,5:8,[5 1],[6 2],[7 3],[8 4]};
```

Fitting the partial order model

Having plotted the data, and specified a partial order, we can now find the best fitting partial order model. STACMR provides the `staMR` function which conducts monotonic regression on a data structure according to a given partial order, as described in Section 4.3. It fits the partial order model to each

dependent variable separately and returns both the best-fitting values and overall fit statistic. For example, using the data structures we have previously defined we could call:

```
>> [pred, fval, shrinkage] = staMR (delay, partial, ...
  shrink);
```

Input

The `staMR` function takes a number of arguments, three of which are used here. These are:

- `data` is the name of the data structure in either general or cell array format or the name of structured output from `staSTATS`.
- `partial` is a partial order in either cell array or adjacency matrix format. The default of `partial` is empty.
- `shrink` is the optional shrinkage parameter, as described previously. The default value is -1.

Output

The output of the `staMR` is contained in three variables:

- `pred` is a n-element cell array, where n is the number of dependent variables. Each element of `pred` contains the best-fitting predicted values for the corresponding dependent variable.
- `fval` is the value of the least-squares fit as defined in Chapter 4. Larger values represent worse fits. If the fit is perfect, `fval = 0`.
- `shrinkage` is a $b \times n$ matrix of shrinkage values (where b is the number of between-participant conditions).

From the foregoing specification of `delay` and `E`, we have:

```
>> [x1, f1, shrinkage] = staMR (delay, E);
>> disp([x1{:}])
    0.3676    0.3308
    0.4676    0.4550
    0.5757    0.5346
    0.6118    0.5492
    0.3445    0.2830
    0.4434    0.3034
    0.5081    0.3149
    0.5169    0.3149

>> disp(f1)
```

```
    0.1721

>> disp(shrinkage)
    0.0520    0.0491
    0.0314    0.2650
```

We can use staPLOT to visualize the partial order model in relation to the data means as follows:

```
>> staPLOT(delay, 'group', {1:4, 5:8}, ...
   'label', {'No delay', 'Delay'}, ...
   'axis', {'RB','II'}, 'pred', x1);
```

This produces Figure 5.2. The argument `'pred'` identifies the values in `x1` as the predicted values. In order to visualize which predicted value corresponds to which observed means, `staPLOT` connects the predicted values with a dashed line. There is no implication that the means of any conditions between any two connected means should fall on the connecting line. The plot places the observed fit value of 0.1721, in context. Although not exactly zero, Figure 5.2 shows that the deviation of the predicted values from the observed means is very small relative to the observed standard errors.

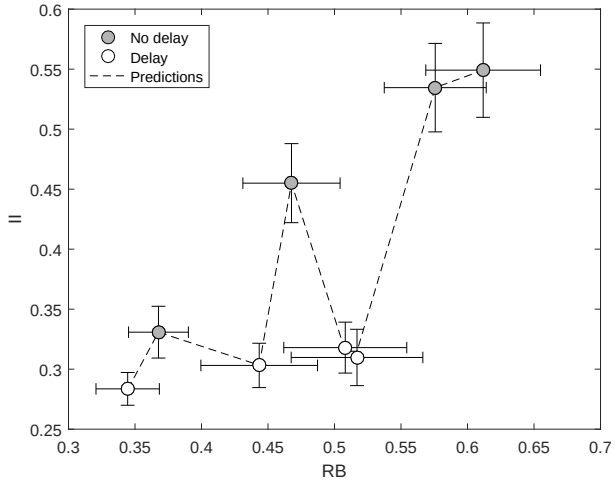

Fig. 5.2 Output from staPLOT based on delay data. The monotonic regression model is shown as a dashed line.

Testing the fit of the partial order model

We can test the observed fit of the partial order model using the bootstrap re-sampling Monte-Carlo procedure described in Section 4.5 to estimate the empirical distribution and the associated p-value. The function `staMRFIT` performs this test. We invoke it as follows:

```
>> [p, datafit, fits]=staMRFIT(data, name-value pairs );
```

Input

The `staMRFIT` function takes the following arguments:

- `data` is the name of a data structure (either in general format, cell array format), or structured output from `staSTATS`. If a data structure is specified then the bootstrap re-sampling is non-parametric. If only the summary statistics are provided (e.g., `delaystats`) then the bootstrap is parametric and assumes that observations are distributed normally for each dependent variable in each condition.
- `'partial'`, `'part'`, `'p'` is a required partial order. This may be in either cell array or adjacency matrix form.
- `'nsamples'`, `'ns'`, `'n'` is the number of Monte-Carlo samples to be drawn in computing the empirical sampling distribution of the fit value. We recommend using about 10,000 for estimating p to the nearest 100th.

Output

The outputs of `staMRFIT` are:

- `p` is the estimated p-value for the hypothesis that the fit of the model is zero. It is the proportion of bootstrap fit values that are greater than or equal to the observed fit value. Note that it will be be different from run to run, as it is a Monte Carlo estimate.
- `datafit` is the observed fit value. It is the same as `f1` returned by `staMR` above.
- `fits` is a vector of length `nsample` of computed bootstrap fit values. Thus, `p` is the proportion of values of `fits` that are greater than or equal to `datafit`.

Applying `staMRFIT` as follows produced the following outputs:

```
>> [p, datafit, fits] = staMRFIT(delay, 'p', E, ...
  'n', 10000);
>> disp([p, datafit])
    0.7470    0.1721
```

Because of the logic of null hypothesis testing, all that can be concluded when $p > \alpha$ is that there is not sufficient evidence to reject the hypothesis that the population means of each outcome are consistent with the partial order given by `E`. Figure 5.3 shows the histogram of fit values contained in the output variable, `fits`.

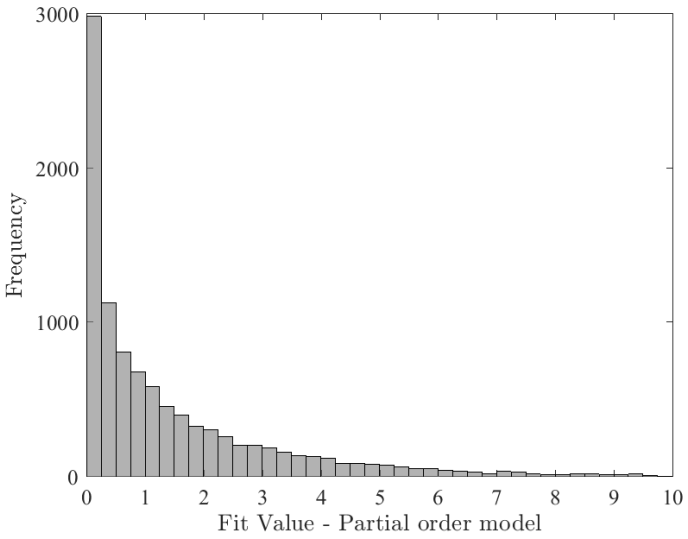

Fig. 5.3 Output from staMRFIT. Histogram of bootstrap fit values for the partial order model. The observed fit value is 0.1721.

Fitting the monotonic model

Having visualized and analyzed the partial order model, we now examine the monotonic model. The function `staCMR` solves the conjoint monotonic regression problem described in Section 4.3. It takes a data structure (in either form) or a cell array of structured output from `staSTATS`, and an optional partial order, and returns the best fitting values to the multivariate data and the least squares fit. For example:

```
>> [x2, f2, stats] = staCMR(delay, E);
>> disp([x2{:}])
    0.3759    0.3150
    0.4850    0.4358
    0.5898    0.5167
```

```
    0.6265    0.5318
    0.3353    0.2856
    0.4186    0.3150
    0.4806    0.3227
    0.4850    0.3227

>> disp(f2)
    1.7493

>> disp(stats.shrinkage)
    0.0520    0.0491
    0.0314    0.2650
```

Notice that the fit value, `f2`, found for the monotonic model is greater than the fit value, `f1`, found for the partial order model. The difference is the cost of the added monotonicity constraints which require the fitted means to have a common order on the two outcomes. The third output variable, `stats`, contains a number of statistics concerning the operation of `staCMR`. One of these contains the computed shrinkage values, `stats.shrinkage`. Because these do not depend upon the model, they are identical to the values in `shrinkage` obtained from `staMR`.

We can inspect the fit of the monotonic model by using `stPLOT` to generate the state-trace plot:

```
>> staPLOT(delaystats, 'group', {1:4, 5:8}, 'label', ...
  {'No delay', 'Delay'}, 'axis', {'RB','II'}, ...
  'pred', pred);
```

This produces Figure 5.4.

Testing the fit of the monotonic model

The function `staCMRFIT` is used to test the hypothesis that there is no difference in fits between the monotonic and partial order models. This function uses the bootstrap resampling Monte-Carlo procedure described in Section 4.5 to estimate the empirical distribution and a p-value of the difference in fits of the monotonic and partial order models. The function invokes both staMR and staCMR. The function call is analogous to `staMRFIT`:

```
>> [p, datafit, fits]=staCMRFIT(data, name-value pairs );
```

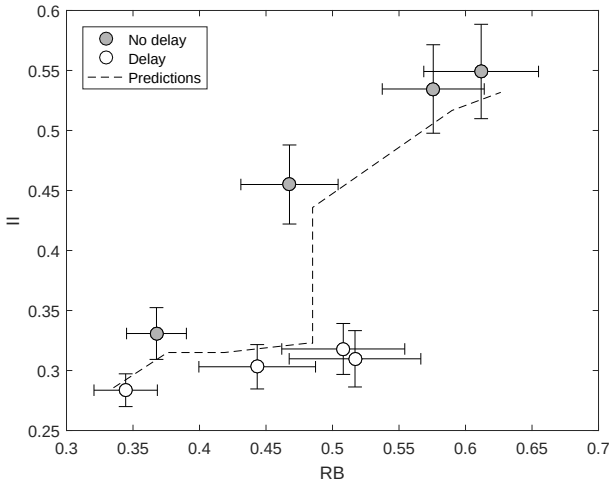

Fig. 5.4 Output from staPLOT based on delay data. The conjoint monotonic regression model is shown as a dashed line.

Input

- `data` is the name of the data structure (in either general or cell array format) or the name of structured output from `staSTATS`). If specified as a data structure then the bootstrap re-sampling is non-parametric, otherwise the bootstrap is parametric and assumes that observations are distributed normally for each dependent variable in each condition.
- `'partial'`, `'part'`, `'p'` is an optional partial order in either cell array or adjacency matrix form.
- `'nsamples'`, `'ns'`, `'n'` is the number of Monte-Carlo samples to be drawn in computing the empirical sampling distribution of the fit value. We recommend using about 10,000 for estimating p to the nearest 100th.

Output

- `p` is the estimated p-value for the hypothesis that the difference in fit between the monotonic model and the partial order model is zero. It is the proportion of differences of bootstrap fits values that are greater than or equal to the observed difference in fit value.
- `datafit` is the observed difference in fit value. If the partial order is omitted, or is empty, then the difference is just the value of the fit of the monotonic model fit because the fit of an empty partial order model is zero.

- `fits` is a vector of length `nsample` of computed differences in bootstrap fit values. Thus, `p` is the proportion of components of `fits` that are greater than or equal to `datafit`.

Applying this function to the delay data produced the following output:

```
>> [p,datafit,fits]=staCMRFIT(delay, 'partial', E,...
  'nsamples', 10000);
>> disp([p datafit])
    0.1753    1.5772
```

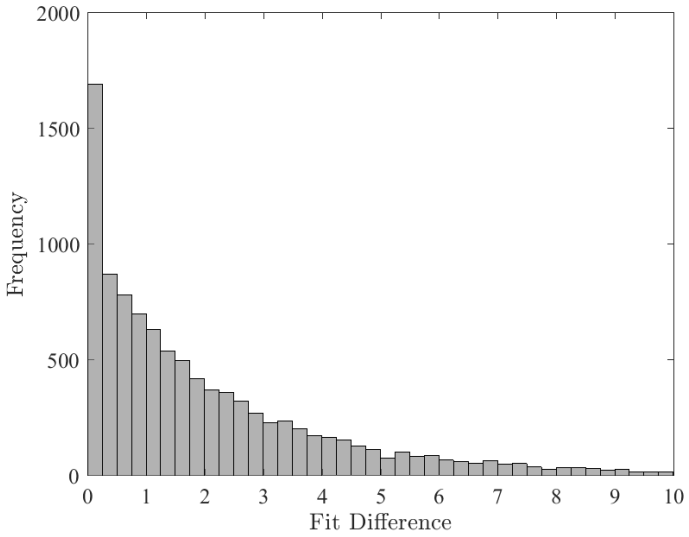

Fig. 5.5 Output from staCMRFIT. Histogram of bootstrap difference in fit values for monotonic model and partial order model.

Figure 5.5 shows the resulting histogram of fit difference values (contained in the output variable, `fits`). The observed difference in fit between the monotonic model and the partial order model is 1.5772 and the estimated p-value, based on 10,000 random samples is 0.1753 (because the bootstrap sampling is random, this estimate will differ slightly from run to run). It can be concluded that there is not sufficient evidence to reject the hypothesis that the population means have the same order across the conditions.

5.3 On Type I and Type II errors

We conclude this chapter with an observation concerning Type I and Type II errors. Visual inspection of Figure 5.1 strongly suggests that the state trace is two-dimensional. There is a sequence of points (corresponding to the four training blocks) for the no-delay condition that appears to be distinct from the sequence of points for the delay condition. However, inspection of Figure 5.4 also shows that the best-fitting monotonic model captures these points quite well. This is confirmed by the statistical analysis which returns a p-value (approximately, 0.18) that is not significant. The lesson to be drawn from these observations is that visual inspection of the state-trace plot may be deceptive and that the monotonic model is surprisingly flexible. This can be considered both a vice and a virtue.

On the negative side, being able to conclude that a state-trace is (truly) two-dimensional requires both sufficient precision in the mean estimates and a configuration of points in outcome space that includes a sufficient number of violations of the monotonic model. The delay data suffer to some degree on both counts. First, as shown by the error bars in Figures 5.1 and 5.4, the precisions of the mean estimates are not large. Simulation suggests that if these precisions are increased 2.5 times (corresponding to 2.5 times the number of participants in each group) then the resulting p-value would be significant, assuming that the configuration of means remains the same. This leads to the second problem. The configuration of points in the state-trace shown in Figure 5.1 offers few violations of monotonicity. In part, this is because three of the points – the leftmost in the delay condition and the two rightmost in the no delay condition – fall beyond the area in DV space where violations can occur. If the means were independent (corresponding to a fully randomized or between-participants design) then these data points would be fit perfectly. It is only because of correlations between the data points (due to the presence of a within-participant independent variable) that this does not happen in the present case. This leaves only two of the four points in the no delay condition and three of the four points in the delay condition producing violations of monotonicity in relation to each other. As noted at the start of this chapter, the optimal state-trace design may be non-factorial. This could be achieved in the present case by designing an experiment with additional no delay conditions having means falling between 0.4 to 0.5 on the RB task and between 0.35 and 0.35 on the II task. Similarly, additional delay conditions may be included whose means are greater than 0.5 on the RB task.

However, on the positive side the flexibility of the monotonic model is such that if it can be rejected then a very strong conclusion can be drawn. Namely, that the data require more than one latent variable under any possible input and (monotonic) output mappings. As discussed in Chapter 3, a major problem with the application of the logic of functional dissociation and analysis of variance is that it is 'too easy' to reject the hypothesis of a

single latent variable, leading to an unacceptably high Type I error rate. In the case of functional dissociation this is in part because a single dissociation is formally consistent with the monotonic model, and also due to the fact that it requires acceptance of the null hypothesis of no effect on one dependent variable. Similar problems arise in the analysis of variance where the too stringent requirement of a linear output mapping also leads to a too easy rejection of the one-dimensional model. In light of these practices, it may appear that STA makes rejection of the monotonic model 'too hard', leading to an unacceptably high Type II error rate. However, we believe that this is a welcome result and that by using STA, researchers will become better accustomed to producing the level of evidence that is actually required to conclude that variation in two or more dependent variables is due to more than one latent variable.

Chapter 6
Independent observations with binary dependent variables

6.1 Introduction

In this chapter we outline the application of STA to binary dependent variables using the STACMR software package. Binary variables consist of counts of 'hits' (or 'successes") and 'misses' (or 'failures") over a set of independent Bernoulli trials. In a given experimental design, it is supposed that such counts are obtained across a set of experimental conditions defined by the levels of one or more independent variables. Apart from the form of the data, the treatment of binary variables by STACMR differs from that of continuous variables in two ways:

1. The monotonic regression weights correspond to the number of trials in each condition rather than the precision of the means.
2. Because the distribution of counts is known to be binomial, the bootstrap re-sampling step is parametric (rather than distribution-free as in the case of continuous variables).

6.2 STACMR software for binary data

In this section we describe the STACMR functions for state-trace analysis of binary data. For a more detailed treatment of this and the example we use, see Kalish et al. (2016).

Input data structure

The data structure for binary variables is simpler than for the continuous case. As in that case, it may be in either general or cell array format.

If in general format, the data will consist of five columns: column 1 is the participant number, column 2 is the condition number, column 3 is the dependent variable number, column 4 is the number of successes, and column 5 is the number of failures.

If in cell array format, the data is organized as a $N \times n$ cell array where N is the number of observation units (typically, participants) and n is the number of dependent variables. Each element of this cell array is a $k \times 2$ matrix containing the number of hits and misses in each of the k conditions.

The function `staSTATSBN` calculates summary statistics relevant to STA. These include, for each participant (indexed by `isub`) and dependent variable (indexed by `dvar`), the following:

- `y{dvar}.means` = vector of 'means' corresponding to the proportion of successes.
- `y{dvar}.weights` = vector of weights corresponding to the number of trials (i.e., number of successes + number of failures).

Face-inversion data

We illustrate the application of the STACMR functions for binary dependent variables with data from a study of the face-inversion effect conducted by Prince, Hawkins, Love & Heathcote (2012b). In this study, the dependent variables were accuracy of memory for faces and accuracy of memory for houses. There were six conditions defined by the combination of two factors; stimulus orientation (upright, inverted), and study duration (short, medium, and long). There were $N = 18$ participants each of whom were tested under all $k = 6$ conditions on each of $n = 2$ dependent variables. The data from each participant therefore can be analyzed individually.

The data (counts of hits and misses) for each participant, condition, and dependent variable are contained in the text file, `dfie.dat`, in general format. We use the function, `BNgen2cell.m` to convert these data from general to cell array format and the function `staSTATSBN` to convert these data to a structured cell array, as described above. That is,

```
>> dfie = load('dfie.dat');
>> data = BNgen2cell(dfie);
>> dfiestats = staSTATSBN(data);
```

The state-trace plot

The state-trace plot for a binary data structure is generated by the function `staPLOTBN` is a similar manner to `staPLOT`. Example call:

```
>> staPLOTBN (data, name-value pairs );
```

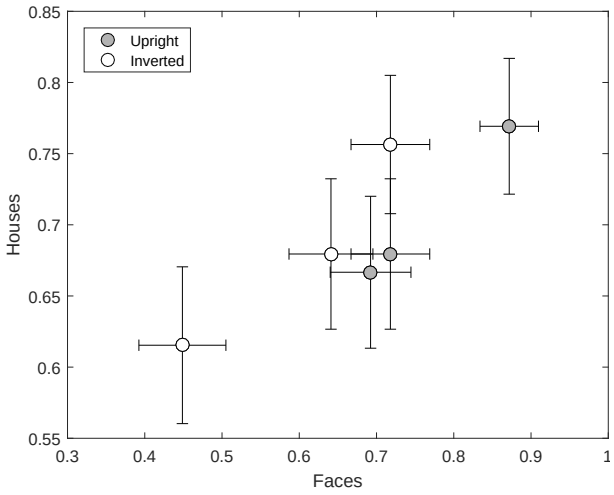

Fig. 6.1 Output from `staPLOTBN` for the first participant in the `dfie` binary data structure.

Input

- `data` is the name of the data structure.
- `'subj'`, `'s'` identifies the participant number to be plotted.
- `'groups'`, `'group'`, `'g'` is a cell array that defines the conditions that will be indicated by different symbols on the plot. In the present example (using `dfiestats`), conditions 1 to 3 correspond to the upright orientation and conditions 4 to 6 correspond to the inverted orientation.
- `'labels'`, `'label'`, `'lab'` is a cell array that defines the labels for the different groups of conditions defined by `'groups'`..
- `'axislabels'`, `'axislabel'`, `'axis'` is a cell array that defines the labels of the x-axis and y-axis, respectively (note: the same result can be achieved using the Matlab functions, `xlabel` and `ylabel`).

The state-trace plot shown in Figure 6.1 is produced by the following command:

```
>> staPLOTBN(dfiestats, 's',1, 'g',{1:3,4:6}, 'lab', ...
  {'Upright','Inverted'}, 'axis',{'Faces','Houses'});
```

Fitting the partial order model

The function `staMRBN` fits a partial order model to a data structure. An example call:

```
>> [pred, fval, gsq] = staMRBN (data, partial);
```

Input

- `data` is the name of the binary data structure.
- `partial` is a partial order (required) in either cell array or adjacency matrix format.

Output

- `pred` is an n-element cell array, where n is the number of dependent variables. Each component of `pred` contains the best-fitting predicted values (proportion of hits) for the corresponding dependent variable.
- `fval` is the value of the least squares fit.
- `gsq` is the corresponding G^2 fit, defined as,

$$G^2 = 2\sum_{i=1}^{n}\sum_{j=1}^{k} y_{i,j}\ln(y_{i,j}/x_{i,j})$$

where n is the number of dependent variables, k is the number of conditions, and y_{ij} and x_{ij} are the observed and predicted means, respectively, on dependent variable i in condition j. When the weights are the total number of trials in each condition, x also maximizes the likelihood and hence minimizes G^2 (Kalish et al., 2016).

We now apply `staMRBN` to the summary data set, `dfiestats`. In order to do so, we have to specify an appropriate partial order. Following Prince et al. (2012b), we define a partial order that constrains performance not to decrease as a function of study duration. That is,

```
>> E = {1:3,4:6};
>> [x1, f1, g1] = staMRBN (dfiestats, E);
```

For participant 1, the observed means are:

```
>> [dfiestats{1,1}.means, dfiestats{1,2}.means]
ans =
    0.7179    0.6795
    0.6923    0.6667
```

```
    0.8718    0.7692
    0.4487    0.6154
    0.6410    0.6795
    0.7179    0.7564
```

The fitted means are:

```
>> [x1{1,:}]
ans =
    0.7051    0.6731
    0.7051    0.6731
    0.8718    0.7692
    0.4487    0.6154
    0.6410    0.6795
    0.7179    0.7564
```

The least-squares fit and G^2 values for this participant are:

```
>> [f1(1), g1(1)]
ans =
    0.0321    0.1525
```

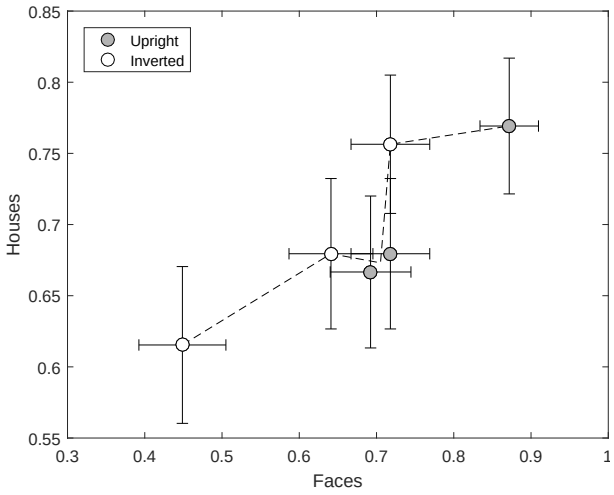

Fig. 6.2 Output from `staPLOTBN` for the first participant in the `dfie` binary data structure showing fit of the partial order model.

The state-trace plot shown in Figure 6.2 is produced by the following command:

```
>> x = staMRBN (dfiestats, E);
>> staPLOTBN(dfiestats,'subj',1,'group',{1:3,4:6},...
  'label',{'Upright','Inverted'},'axis', ...
```

```
{'Faces','Houses'},'pred',x,);
```

The additional option, `'pred'`, specifies the predicted values to be plotted (i.e., the output from `staCMRBN`).

Testing the fit of the partial order model

The fit of the partial order model to binary data is evaluated using the bootstrap re-sampling Monte-Carlo procedure described in Section 4.5. The STACMR function is called staMRFITBN. Example call:

```
>> [p, datafit, fits] = staMRFITBN (data, name-value pairs );
```

Input

- `data` is the name of the binary data structure.
- `'partial'`, `'part'`, `'p'` is the partial order in cell array or adjacency matrix form.
- `'nsamples'`, `'ns'`, `'n'` is the number of Monte-Carlo samples (about 10,000 for p to the nearest 100th).

Output

- `p` is the estimated p-value for the hypothesis that the fit of the model is zero. It is the proportion of bootstrap fit values that are greater than or equal to the observed fit value.
- `datafit` is the observed G^2 value.
- `fits` is a vector of length `nsample` of computed bootstrap G^2 values. Thus, `p` is the proportion of components of `fits` that are greater than or equal to `datafit`.

Example output (note that `p` and `fits` will be differ from run-to-run as they are Monte Carlo estimates):

```
>> [p, datafit, fits] = staMRFITBN (dfiestats, 'p', E, ...
  'n', 10000);
>> disp([p, datafit])
   0.7361    0.1525
```

The observed p-value is 0.7361. It can be concluded that there is no evidence to reject the hypothesis that the population means are consistent with the partial order given by `E`. Figure 6.3 shows the resulting histogram of fit values (contained in the output variable, `fits`).

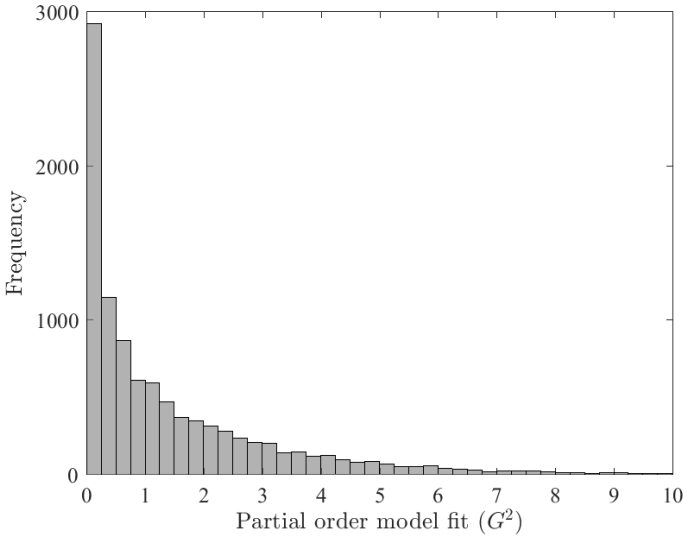

Fig. 6.3 Output from staMRFITBN. Histogram of bootstrap fit values for the fit of the partial order model for participant 1 in the `dfie` data set.

The fit of the monotonic model

The monotonic model is fit to a binary data structure using the function, `staCMRBN`. Example call:

```
>> [pred, fval, gsq] = staCMRBN (data, partial);
```

Input

- `data` is the name of a binary data structure.
- `partial` is an optional partial order in either cell array or adjacency matrix format.

Output

- `pred` is a $N \times n$ cell array, where N is the number of observation units (participants) and n is the number of dependent variables. Each element of `pred` contains the best-fitting predicted values (proportion of hits) for the corresponding participant and dependent variable.
- `fval` is the value of the least squares fit.
- `gsq` is the corresponding G^2 fit.

Applying the function,

```
>> [x2, f2, g2] = staCMRBN (dfiestats, E);
```

For participant 1, the fitted means are:

```
>> [x2{1,:}]
ans =
    0.7051    0.6752
    0.7051    0.6752
    0.8718    0.7692
    0.4487    0.6154
    0.6410    0.6752
    0.7179    0.7564
```

The least-squares fit and G^2 values for this participant are:

```
>> [f2(1), g2(1)]
ans =
    0.0342    0.1622
```

We observe that `f2` is greater than `f1` and that `g2` is greater than `g1`. This difference is the cost, for the data from this participant, of the added monotonicity constraint.

The corresponding state-trace plot, shown in Figure 6.4 can be generated using the following call to `staPLOTBN`:

```
>> x = staCMRBN (dfiestats, E);
>> staPLOTBN(dfiestats, 'subj', 1, 'group', ...
  {1:3, 4:6}, 'label', {'Upright', 'Inverted'}, ...
  'axis', {'Faces', 'Houses'}, 'pred', x,);
```

Testing the fit of the monotonic model

The function `staCMRFITBN` uses the bootstrap re-sampling Monte-Carlo procedure described in Section 4.5 to estimate the empirical distribution and a p-value of the difference in fits of the monotonic model and the partial order model for binary data. Example call:

```
>> [p,datafit,fits] = staCMRFITBN (data, name-value pairs );
```

Input

- `data` is the name of the binary data structure.
- `'partial`, `'part'`, `'p'` is an optional partial order (default = none)

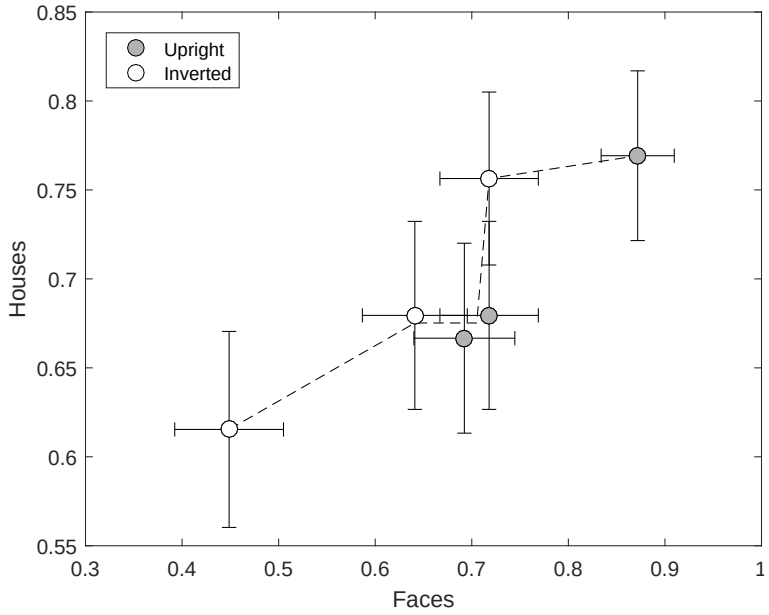

Fig. 6.4 Output from `staPLOTBN` for the first participant in the `dfie` binary data structure showing fit of the monotonic model.

- `'nsample'`, `'ns'`, `'n'`, is the number of Monte-Carlo samples (about 10,000 for p to the nearest 100th)

Output

- `p` is the estimated p-value for the hypothesis that the difference in fit between the monotonic model and the partial order model is zero. It is the proportion of differences of bootstrap fits values that are greater than or equal to the observed difference in fit value.
- `datafit` is the observed difference in fit value.
- `fits` is a vector of length `nsample` of computed differences in bootstrap fit values. Thus, `p` is the proportion of components of `fits` that are greater than or equal to `datafit`.

Example output (note that `p` and `fits` will be differ from run-to-run as they are Monte Carlo estimates):

```
>> [p, datafit, fits] = staCMRFITBN (dfiestats, 'p', ...
  E, 'n', 10000);
>> disp([p, datafit])
   0.8207    0.0098
```

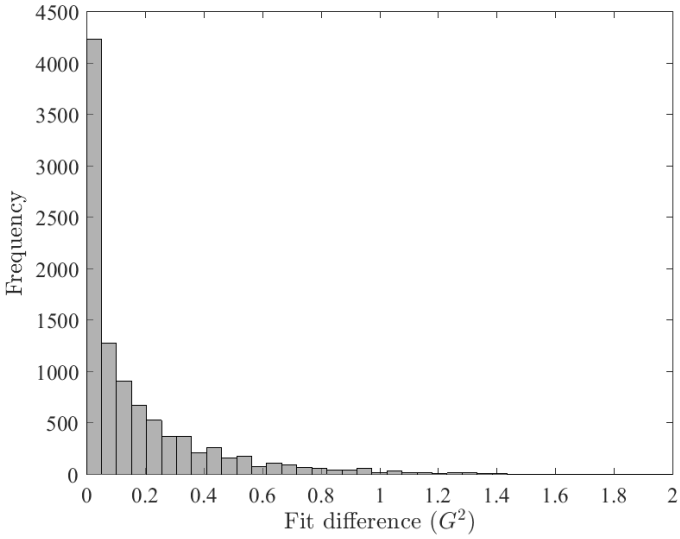

Fig. 6.5 Output from staCMRFITBN. Histogram of bootstrap fit values for the difference in fit between the monotonic and partial order models for participant 1 in the `dfie` data set.

The observed p-value is 0.8207. It can be concluded that there is no evidence to reject the hypothesis that the population means have the same order across the conditions. Figure 6.5 shows the resulting histogram of fit values (contained in the output variable, `fits`).

Chapter 7
More Examples

7.1 Introduction

In this chapter we illustrate the application of the STACMR software package to data derived from some additional experimental designs.

7.2 Fully repeated measures designs

In a fully repeated measures design all the experimental factors are manipulated within-participant. The repeated measures may also include the set of dependent variables as 'levels' of a nominated independent variable (as with the levels of 'faces' and 'houses' in a face-inversion experiment). As mentioned previously, this plays no role in the analyses conducted using the STACMR package. Even if the observations are correlated across dependent variables, they are treated by the software as independent.

A non-significant result

We use the results from the study by Nakabayashi et al. (2012) discussed in Chapter 3 to illustrate the application of STACMR to a fully within-participants design. Each participant in this study completed a set of memory tasks that resulted from crossing the familiarity of the to-be-recalled item (two levels) with three different ways of studying the items. These six conditions are thus fully repeated. The data from this study are in cell array format in the file `naka.mat`. Thus:

```
>> load naka
```

```
>> disp(naka)
    [30x6 double]    [30x6 double]
```

The fact that the data are fully repeated is reflected in the size of the cell array, which is 1 (the number of between-participant independent variables) by 2 (the number of dependent variables). Each cell contains observations from 30 participants in each of 6 conditions.

The data can be visualized using `staPLOT`

```
>> staPLOT(naka, 'groups', {1:3, 4:6}, ...
  'labels',{'Familiar', 'Unfamiliar'}, 'axislabels', ...
  '{Faces (A^\prime)', 'Houses (A^\prime)'});
```

which produces Figure 7.1

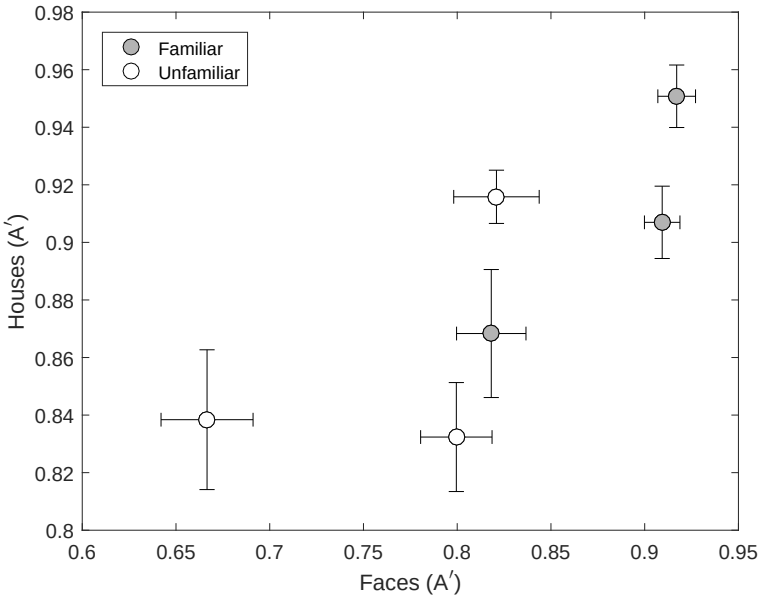

Fig. 7.1 A state-trace plot of the results from Nakabayashi et al. (2012).

The predictions of the best fitting monotonic regression model can be found using `staCMR` without, in this case, specifying a partial order:

```
>> [x, f, s] = staCMR(naka);
>> disp([x{:}])
    0.9093    0.9131
    0.8182    0.8653
    0.9171    0.9505
    0.7995    0.8340
    0.6666    0.8340
```

```
    0.8209    0.9131

>> disp(f)
    0.4027

>> disp(s.shrinkage)
    0.5469    0.2358
```

The best fitting model is quite close to the data, as can be seen by plotting the predicted values:

```
>> staPLOT(naka, 'groups', {1:3 4:6}, 'labels', ...
 {'Familiar', 'Unfamiliar'}, 'axislabels', ...
 {'Faces (A^\prime)','Houses (A^\prime)'}, 'pred', x);
```

which produces Figure 7.2

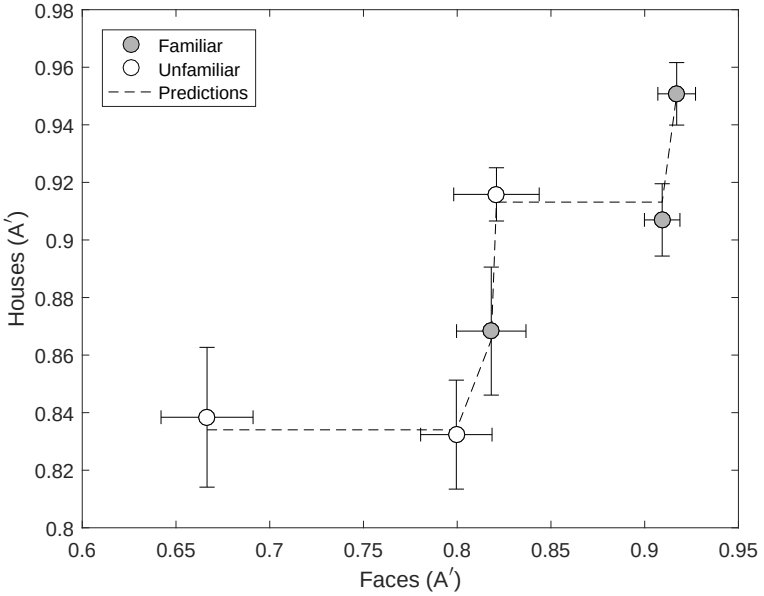

Fig. 7.2 The results of the Nakabayashi et al. (2012) study as a state trace plot, along with the best fitting monotonic model shown as a dashed line.

The test of the monotonic model is conducted by the following command:

```
>> [p, datafit, fits] = staCMRFIT(naka, 'n', 10000);
>> disp([p datafit])
    0.6107    0.4027
```

which again does not allow rejection of the null hypothesis that the monotonic model is adequate. The distribution of the test statistic is shown in Figure 7.3.

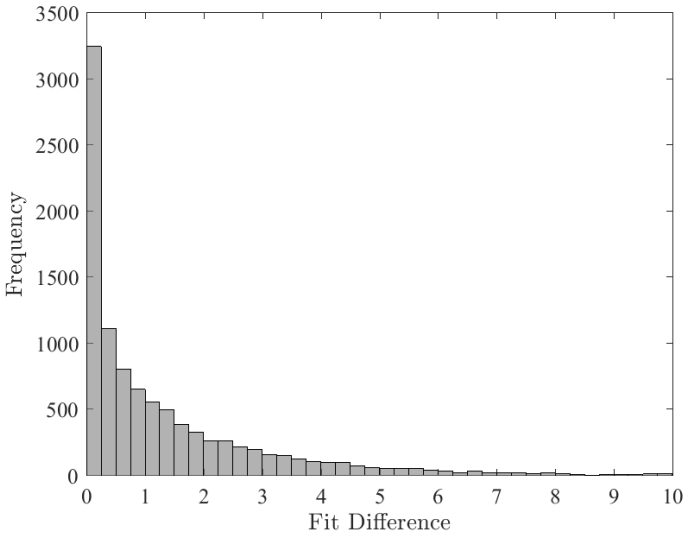

Fig. 7.3 The distribution of the fit statistic for the monotonic model applied to the Nakabayashi et al. (2012) data.

A significant result

Although, as noted earlier, the monotonic model can be quite flexible, the next case shows that this model is not immune to rejection. The data are derived from a study by Singmann and Klauer (2011) who asked participants to evaluate conditional (if...then) arguments under two different instructions. Under *deduction* instructions, participants were asked to judge 'how necessary' is the conclusion of an argument given its premises. Under *induction* instructions, participants were asked to judge 'how plausible' is the conclusion given the premises. In their Experiment 2, there were four kinds of argument – the valid *modus ponens* and *modus tollens*, and the invalid *asserting the consequent* and *denying the antecedent* – combined with three kinds of content called prological, counterlogical, and neutral, yielding a total of 12 conditions. The STACMR package includes the data file, `SK2011Exp2.mat`, that contains a data structure in cell array form in

the variable sk. In addition, it also contains the following variables, used in staPLOT, defined as follows:

```
g = {[1 5 9],[2 6 10],[3 7 11],[4 8 12]};
lab = {'MP','MT','AC','DA'};
limits = {[30 100],[30 100]};
axislab = {'Deductive','Inductive'};
```

First, load the data and fit the monotonic model, without a partial order, to the data:

```
>>load SK2011Exp2
>>x = staCMR(sk);
```

Then we can plot the results using the variables defined above. That is,

```
>> staPLOT(sk,'p',x,'g',g,'lab',lab,'axislimits',...
limits,'axislab',axislab);
```

which produces Figure 7.4.

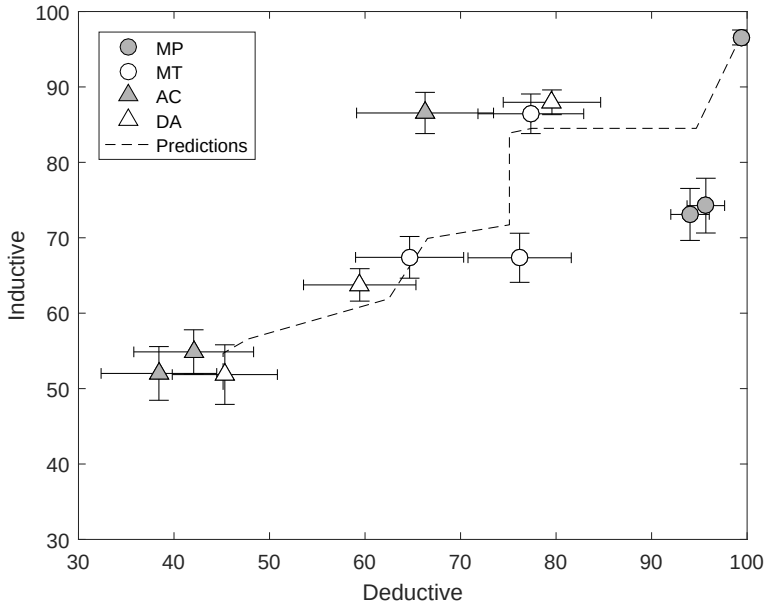

Fig. 7.4 The results from Experiment 2 of Singmann and Klauer (2011) as a state trace plot, along with the best fitting monotonic model shown as a dashed line.

The monotonic model leaves a significant residual, as the following analysis shows:

```
>> [p, datafit, fits] = staCMRFIT (sk, 'n', 10000);
```

```
>> disp([p datafit])
    0.0054   19.5975
```

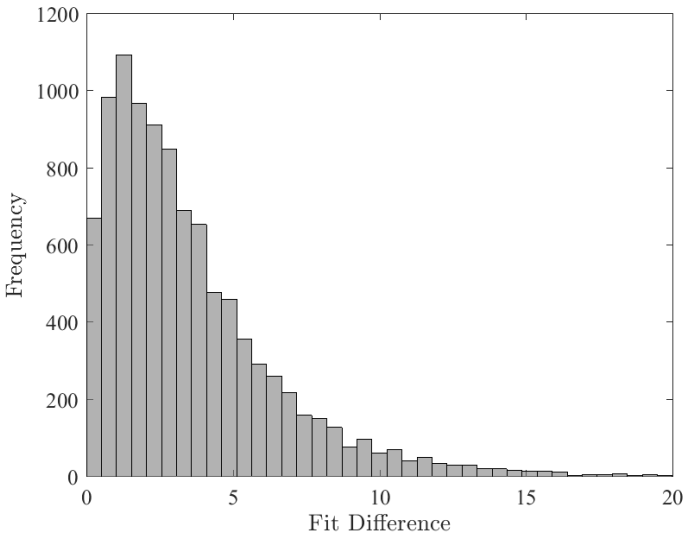

Fig. 7.5 The distribution of the fit of the monotonic model applied to the Singmann and Klauer (2011) data.

The empirical sampling distribution of the fit of the monotonic model for these data is shown in Figure 7.5. This model can clearly be rejected at $\alpha = .05$ and it can be concluded that there is more than one latent variable mediating the effects of the independent variables on the two dependent variables across the conditions of the experiment.

7.3 Fully randomized design

As mentioned previously, there is no conceptual problem in using more than two outcome measures to look for the influence of multiple latent variables. One example of this kind of experiment is Experiment 1 of Lewandowsky, Yang, Newell and Kalish (2012) which investigated the same kind of category learning processes that was of interest to Dunn et al. (2012), discussed in Chapter 5. Lewandowsky et al. (2012) were concerned with the effect of individual differences in working memory capacity on the rate at which people could learn categories with different structures. They measured the performance of participants as they learned six different categorization tasks.

These tasks differed in several ways, the most important being that three were *information integration* tasks and three were *rule based* tasks. The three versions of each type of task differed in ways that were expected to influence their relative difficulty, so there was an easy, medium and hard version of each. Participants completed a total of six blocks of training and were each scored on the four-part working memory capacity battery developed by Lewandowsky, Oberauer, Yang and Ecker (2010). Based on these scores, participants were divided into high, medium, and low working memory terciles at the time of data analysis. Here we consider performance of just the high and low working memory groups.

The experiment conducted by Lewandowsky et al. (2012) provides a useful example for two reasons; it shows the power of the STACMR package in managing an experiment with six dependent variables, and it shows the limitations of the package as the number of conditions increases. We return to this latter point shortly.

The data from the experiment are provided in the file, `LYNK.mat`. The variable `d` contains the subset of the data that concerns this example. We begin by loading these data:

```
>> load LYNK
```

These data consist of observations under 12 conditions (corresponding to the six training blocks for each of the two working memory groups) on each of six outcome variables. The state-trace therefore corresponds to a set of 12 points embedded in a six-dimensional DV space. Because of the difficulty in visualizing a 6-dimensional space, it is of more use to plot each pairwise combination of dependent variables. This is shown in Figure 7.6.

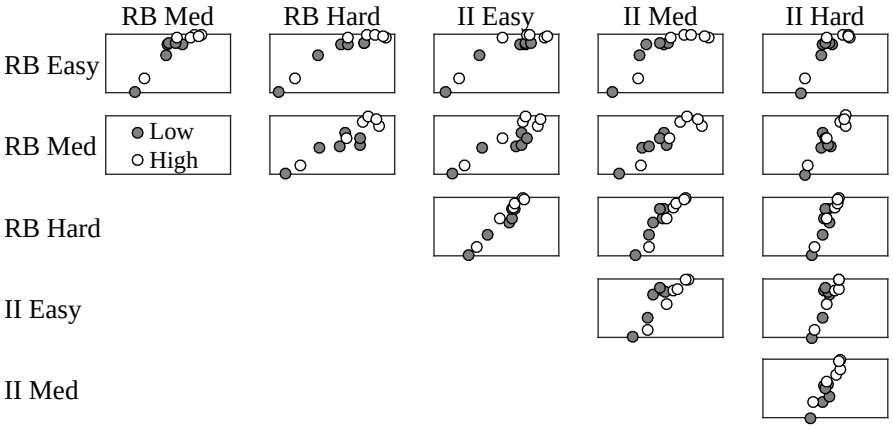

Fig. 7.6 The matrix of state-trace plots for each pairwise combination of the six outcome variables in Lewandowsky et al. (2012).

The first step in the analysis of the LYNK data set is to investigate the fit of an appropriate partial order model. This model is defined by the expectation that performance will increase from blocks 1 to 6 for both low working memory group (conditions 1 to 6) and for the high working memory group (conditions 7 to 12). Because there is no reason to suppose one group should perform better than the other, no partial order is placed on the groups themselves. The partial order model is tested as follows:

```
>> E = {1:6, 7:12};
>> [p1, datafit1, fits1] = staMRFIT(d, 'p', E, 'n',...
10000);
>> disp([p1 datafit1])
    0.0730   19.3268
```

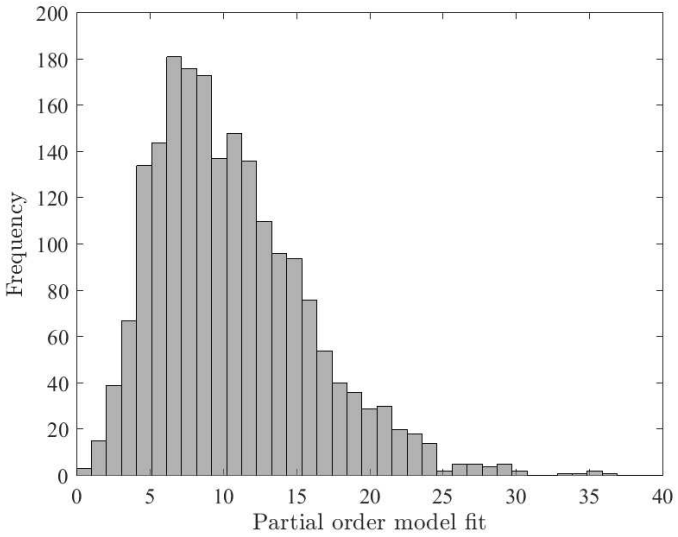

Fig. 7.7 The distribution of the fit of the partial order model applied to data from Lewandowsky et al. (2012).

Because there are six outcomes rather than two in this example, the algorithm takes longer to find an optimal solution simply because there are more violations of the partial order with six variables than with two. Be prepared to wait. With 10,000 samples, this function took approximately one minute on a standard laptop. Compared to the distribution of bootstrap fit values shown in Figure 7.7, the observed value of 19.33 is relatively high although it does not reach statistical significance, $p_{est.} = 0.073$.

The next step is to determine the difference in fit between the monotonic model and the partial order model. Thus:

```
>> [p, datafit, fits] = staCMRFIT (d, 'p', E, 'n',...
2000);
>> disp([p datafit])
    0.9950    4.6413
```

Notice that the number of samples here is lower than in other examples. The high p value makes this possible, as precision is not critical. The reason for the smaller number of samples is that the computation time for each sample is just under 0.1 seconds, so the sample in Figure 7.8 took about two minutes on a desktop to draw. Readers may wish to test the speed of their own computers by reproducing the example.

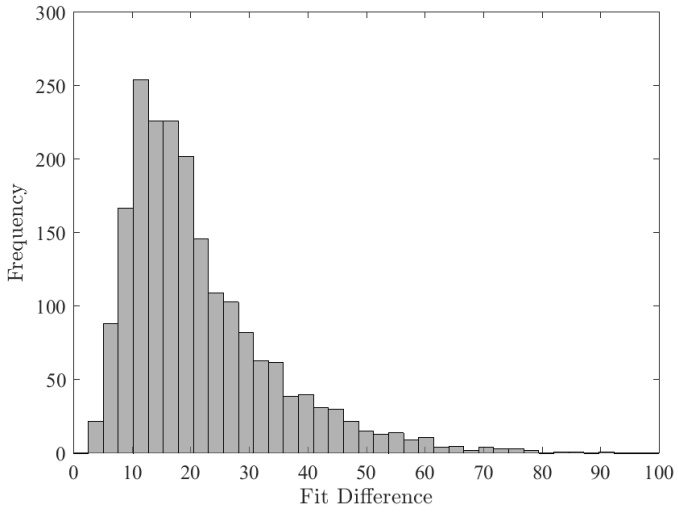

Fig. 7.8 The distribution of the difference in fit between the monotonic and partial order models applied to data from Lewandowsky et al. (2012).

The selection of the two extreme working memory capacity groups was explained as an example of the limitations of STACMR. If all three groups are included in the analysis then there are 18 data points measured on six outcomes. Although this is only a 50% increase, because the time taken by the CMR algorithm increases exponentially, this relatively modest increase results in a 30-fold increase in computation time. The result is that the two minutes it took to conduct 2,000 samples on two groups would take over an hour for three groups. Because of the size of the problem there are also memory considerations that may exceed the capacity of some computers. The moral is that large data sets should be approached with caution.

7.4 On near degeneracy

We conclude this section by touching on an important aspect of data analysis using STA; the concept of *degeneracy*. We say that a vector, x, is degenerate if all its components are equal. That is, if $x_i = x_j$ for all $i, j \in \{1, \ldots, k\}$. Extending this idea to a set of observed values, X, associated with measurement error, we say that X is *near degenerate* no difference between any pair of components is statistically significant. This idea is illustrated in Figure 7.9.

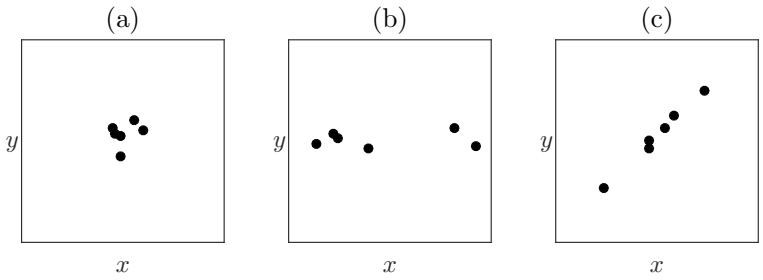

Fig. 7.9 Illustration of near degeneracy of dependent variables. (a) Both dependent variables, x and y, are nearly degenerate - the differences between conditions on both variables are small and none is statistically significant. (b) Dependent variable, y, but not dependent variable x is nearly degenerate – at least one of the differences between conditions on x is significant but none of the differences between conditions on y is significant. (c) Neither dependent variable is near degenerate – at least one of the differences between conditions on both x and y is significant.

Figure 7.9(a) illustrates a situation in which each of two dependent variables is near degenerate. In other words, although there are small differences between the means on both outcome variables (here labelled x and y), none of these is statistically significant[1]. The problem this presents is that because a degenerate vector is consistent with all possible partial orders, the fit of any partial order model to a near degenerate set of observations will be close to zero and unlikely to be significant. The same applies to the fit of the monotonic model.

Figure 7.9(b) illustrates a situation in which one of the dependent variables (y in this case) is near degenerate while the other dependent variable (x in this case) is not. The result is that while a partial order places a real constraint on x, it places little or no constraint on y. As a result, the difference in fit between the monotonic model and the partial order model is unlikely to be

[1] For clarity, associated error bars have been suppressed.

statistically significant. Furthermore, if there is no partial order model then the fit of the monotonic model is also unlikely to be statistically significant[2].

Finally, Figure 7.9(c) illustrates a situation corresponding to the one that we have implicitly assumed involving no near degeneracy in either dependent variable. In this instance, the data happen to be monotonically ordered but this need not be the case and, given the lack of near degeneracy, it is an open question whether the difference in fit between the monotonic and partial order models is statistically significant.

As noted above, near degeneracy undermines the logic of applying STA to a data set. When data are near degenerate, nothing meaningful can be concluded from the fact that the data conform to the monotonic model because it is hardly possible that it could be otherwise. It is therefore important first to check that neither dependent variable is near degenerate. This is simply done through the appropriate one-way analysis of variance.

[2] The astute reader will have noted that this corresponds to a single dissociation, demonstrating its lack of informativeness.

Part III
Further topics

Chapter 8
Bayesian approaches

8.1 Introduction

Our approach to state-trace analysis has two main components[1]. The first is a method to solve the problem of finding the best conjoint total order consistent with a specified partial order on two or more variables. Conjoint monotonic regression is the solution to this problem, and it provides both a best-fitting set of predictions, and a statistic that quantifies the degree to which the observed data deviate from the predicted monotonicity. The second component is a null-hypothesis significance test (NHST) based on an empirical sampling distribution of this statistic. As with all frequentist NHSTs, this second component has important limitations (Wasserstein & Lazar, 2016). One of our reasons for developing such a test is to foster its adoption by the active research community which, at the present time, remain committed to the use of NHSTs. While we are confident that the empirical test we have developed is meaningful when used appropriately, it does not obviate the value of a Bayesian form of state-trace analysis.

Bayesian data analysis can be used either for hypothesis testing or for estimation. While Bayesian state-trace analysis could take either of these two forms, the only Bayesian method currently available is the hypothesis testing method of Prince et al. (2012a), as extended by Davis-Stober, Morey, Gretton & Heathcote (2016). We discuss this method briefly here, in order to highlight possible alternative hypothesis testing and estimation approaches.

[1] The reader will note that we avoid formal mathematical presentation in this chapter. We do so as an indication of the preliminary stages of our theorizing here.

8.2 Hypothesis testing

The general approach of Prince et al. is to compare models that have different constraints on the orders of the dependent variables. Their logic is similar to that of our frequentist test, where we assess the difference in fit between a partial order model, which requires each dependent variable to have a total order consistent with a specified partial order, and a monotonic model, which requires both depdndent variables to have a common total order consistent with the same partial order. We explained, in Chapter 4 that likelihood ratio tests, which would allow a frequentist approach to nested model comparison, are unavailable for STA because the relative degrees of freedom of the models cannot be established. This problem is avoided in the Bayesian approach, which allows for the assessment of the evidence for both the partial and total order models.

From a Bayesian perspective, a model is composed of a likelihood and a prior, which together encode the data analyst's belief about how the data came to be – that is, they define a data generating process. Bayesian inference proceeds by computing either the posterior probability of the parameters of this process given the data (for Bayesian estimation) or the prior probability of the data integrated over the parameters (for Bayesian hypothesis testing). To apply Bayesian analysis to STA, a model must parameterize both the input mapping and the output mapping. But, of course, one source of the appeal of STA is that it allows inference about the number of latent variables in the absence of just such a parametric model. The generative model Prince et al. use has two parts. First, the prior probability of the dependent variables (which are the success rates θ for each experimental condition) are uniform over all allowable total orders. Second, the likelihood of the data given θ follow the standard binomial distribution (and so also depend on the number of trials N for each condition). This minimal model enforces the order constraints of STA only in its selection (in the prior) of the allowable total orders. So, for example, their 'encompassing prior' model is specified by assigning equal probability to every possible total order. Their 'trace model', which is equivalent to our partial order model, assigns zero probability in the prior to total orders that are not linear extensions of the partial order. And, finally, their monotonic model, which is equivalent to ours, places equal probability on all shared total orders that are linear extensions of the partial order and zero probability on all other orders.

Specifying a prior over possible orders makes it possible to quantify the relative evidence for both a null and alternative hypothesis. The various models are nested, so Prince et al. are able to take advantage of some special properties of order-restricted models in computing their Bayes factors. The comparison between the likelihood of the data given a model with broader priors relative to a model with more constrained priors is usually very sensitive to the particulars of the prior distribution that defines each model. However, Prince et al. claim that their method is insensitive to the

particulars of these distributions so long as every allowable order has some mass in the prior (which, after all, is what 'allowable' means). To achieve this insensitivity it is necessary to use a special likelihood, based only on the probability masses. Prince et al. thus define a likelihood for the purposes of hypothesis testing that is one when the parameters θ are consistent with a given partial order (one that is more constrained than the prior order that generated them), and zero otherwise. Their Bayes factor is then just the ratio of proportion of the joint θ posterior distribution that is consistent with the partial order to the proportion of the prior that is.

There are some limitations to this approach. The first is that, while the definition of the marginal likelihood as the proportion of the probability mass of the distribution of model parameters in a given region allows the result to be insensitive to changes in the shape of the prior distribution, it has the disadvantage of treating all violations of the allowable orders the same. That is, only the evidentiary value of an experiment can be assessed, as there are no parametric definition of 'monotonicity' to be estimated. In contrast, the CMR approach provides a way of measuring the degree to which a given set of θ values deviate from a common total order – one that can be computed for both dichotomous and metric dependent variables alike. The second is that the method only applies to dichotomous data modelled as binomial. It is unclear if the method can be readily extended to metric data for mixed designs. Finally, the decision derived from a Bayes factor is necessarily uninformative about the distribution of the posterior over the model parameters. That is, accepting or rejecting the monotonic model tell us nothing about the extent to which the data violate the allowable orders.

We would like to suggest that there are alternative Bayesian approaches that might merit future development. One would be an estimation approach, based on using CMR to measure the credible non-monotonicity of the data. The other would represent a more radical re-consideration of state-trace analysis, by parameterizing the monotonic model in a way that allows both estimation and model comparison, but also would allow interpolation between observed points. One way to pursue this second approach would be to replace the uniform prior approach of Prince et al. with a form of Gaussian process regression.

8.3 Estimation based on unrestricted priors

The sensitivity of Bayes factors to priors is a well known problem. In the case of STA, there is no pre-theoretic way to justify any particular allocation of prior probabilities directly to the set of all possible orders. Prince, et al. deal with this problem by taking as a likelihood measure the simple presence or absence of any unallowable order in the estimated cell rates. One way to conceive of this likelihood is from the perspective of counting the number

of pairwise violations of the order constraints[2]. The likelihood used in the Prince, et al. Bayes factor is just a count on whether or not at least one of the pairwise differences of estimated cell rates is unallowable. For metric data, the same measure could be taken over the cell means. A more fine-grained measure of the consistency of sample parameters with the monotonic order would allow inferences to be made about how credible the variation was. As a direct extension, one could simply count the number of pairwise violations. This value is a random variable that could be estimated in the posterior of the cell rates or means[3]. To the extent that the data violate monotonicity, the credible values of this variable should be large. Of course, this is a rather crude measure of monotonicity, but it is one that might be useful when the number of conditions is very large.

Our suggestion is that the CMR could be considered as simply a measurement of non-monotonicity, and used as a measurement on estimated (or sampled) data. The idea would be to use a generic generalized linear model to estimate the cell statistics, whether they are means or rates. This would both imply a prior on the CMR metric and allow estimation of a posterior value. Like any estimation approach, this would depend only weakly on the distribution of the prior on the linear model's parameters. The high density interval of the posterior CMR test statistic ought to be a good indication of how far the data are from a monotonic relationship, and so how credibly non-monotonic the data are.

8.4 Bayes factors from the parametric marginal likelihood

Bayesian data analysis differs from frequentist analysis in that the Bayesian specifies a generative model. This requires a definition of both a parametric likelihood function for the data, and a set of priors over the parameters of that function. Defining a generative model for state-trace analysis that begins with an explicit prior on monotonicity is problematic, as a central claim of the analysis is that it is a non-parametric approach. The dependent variables also need only be monotonically related to the latent variables, which themselves can be any function of the independent variables. A specification of any given relationship between latent variables and dependent variables, or between independent variables and latent variables, necessarily introduces theoretical constructs that can be contentious, and that can compromise the generality of the state-trace approach.

Gaussian processes afford a way to minimally constrain the shape of the monotonic model, and so come close to meeting the non-parametric status

[2] We discussed the pairwise order interpretation of monotonicity in Chapter 2

[3] See, for example, the work of Stephens, Dunn & Hayes (in press)

that state-trace analysis enjoys. One benefit of adopting a parametric specification of the IV-to-DV mapping is that, as with any regression approach, it becomes possible to use the resulting model for prediction as well as for description. There are a variety of ways one might incorporate Gaussian process regression into an analysis of a state-trace design, and full explication will have to await systematic development. In essence, the notion would be to relate the values of one dependent variable to the values of all dependent variables through a covariance function that is only minimally constraining of the dependent values.

The principle of minimal constraint is adopted in order to conform to the ideal of a non-parametric STA. In the case of Gaussian processes, it means choosing a kernel that results in functions that are just smooth enough to allow inference through estimation of their parameters, and no smoother. The kernel can nonetheless be made directional, so that its length-scale component can be taken to represent the degree of monotonicity, increasing when the local predictions are in the same direction as the global predictions. Conversely, its noise component will reflect in part the degree of non-monotonicity, and will grow to the extent that local predictions in one region of the state space are not similar in other regions. Common MCMC methods will produce posterior estimates for both of these parameters. One could then interrogate the posterior from an estimation perspective, to see how large the length-scale is, for example, or the magnitude of the ratio of length- to noise-scale. One could also use the posterior to form Savage-Dickey test, assuming a landmark value for the null can be identified.

To summarize, the CMR algorithm may be of use in some future Bayesian statistical approaches. There is considerable room for innovation on this front, however, and the remarks in this chapter merely begin to set the stage for what we hope are continued developments in Bayesian state-trace analysis.

Chapter 9
Final comments

9.1 Introduction

STA addresses the following question: Are the effects of one or more independent variables on two or more dependent variables mediated by one or more than one latent variable? However, the general structure, introduced in Chapter 1 conceives of the possibility that any number of latent variables may mediate the effects of a set of independent variables on a set of dependent variables. This leads to the question of whether it is possible to determine the number of latent variables involved, whether one, two, three or more. Because the dimensionality of the state-trace cannot exceed the dimensionality of DV space, this question can be phrased as follows: Given n dependent variables, how many latent variables, from one and n, are required to account for the joint effects of a set of independent variables?

The difficulty associated with assessing the dimensionality of the state-trace was noted in Chapter 2. The same problem arises in higher dimensions. Without constraining the output mapping in some way, it is not possible to determine the dimensionality of the state-trace. This is the approach taken by *signed difference analysis* which we briefly outline below.

9.2 Generalizing STA: Signed difference analysis

Signed difference analysis (SDA; Dunn & James, 2003; Dunn & Anderson, submitted) generalizes STA to models with more than one latent variable. It does so by restricting the output mapping to a class of functions. In terms of the general structure introduced in Section 1.2, an output mapping is decomposed into two component mappings called the *structural mapping* and the *measurement mapping*. The structural mapping specifies how latent variables combine for each dependent variable. The measurement mapping is

a monotonic mapping that maps each combination of latent variables onto the corresponding dependent variable.

Formally, let $x = \{x_1, \ldots, x_n\}$ be a set of n dependent variables and let $u = \{u_1, \ldots, u_m\}$ be a set of m latent variables. Then, as described in Section 1.2, the output mapping is defined as a vector of functions, $g = (g_1, \ldots, g_n)$, such that $x = g(u)$. In STA, it is proposed that each $x = g(u_1)$, where g_i is a monotonic function of the single latent variable, u_1. In SDA, it is proposed that each $x_i = h_i \circ g'_i(u)$, where g'_i is a multivariate function of the latent variables, u, and h_i is a univariate monotonic function of $g'_i(u)$. Then $g' = (g'_1, \ldots, g'_n)$ is called the structural mapping and $h = (h_1, \ldots, h_n)$ is called the measurement mapping and $h \circ g'$ is called the *model* of x.

Although, in principle, the structural mapping may take any form, most interest lies in the case in which each g'_i is a linear function of the latent variables. That is,

$$g'_i(u) = \sum_{j=1}^{m} a_{ij} u_j \tag{9.1}$$

where a_{ij} is a constant associated with dependent variable, x_i, and latent variable, u_j. Let $a_i = (a_{i1}, \ldots, a_{im})$. Then the matrix,

$$A = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

is called the *model matrix* of the linear structural mapping, g'. As an example, consider the output mapping based on signal detection theory given earlier in Equation (1.1):

$$\begin{aligned} FAR &= \Phi(-c) \\ HR &= \Phi(d' - c) \end{aligned}$$

Here, the measurement mapping is $h = (\Phi(\cdot), \Phi(\cdot))$, the structural mapping is $g' = (-c, d' - c)$, and the corresponding model matrix is,

$$A = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}$$

It is apparent that STA is a special case of Equation (9.1) where, $g'_i(u) = a_{i1} u_1$. Because of the monotonic measurement mapping, without loss of generality, each a_{i1} can be set to one. Then $g'(u) = u_1$ and $x = h \circ g'(u) = h(u)$, equivalent to the output mapping proposed under STA (with h replacing g above). The corresponding model matrix is therefore,

$$A = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

The insight behind SDA is that a linear structural mapping places constraints on the set of dependent variables in terms of their *signed differences*. Let $x_{(j)} = (x_{1(j)}, \ldots, x_{n(j)})$ be the vector of the values of n dependent variables $x_1, \ldots, x_n$, under condition j. Then $\text{sign}(x_{(j)} - x_{(k)})$ is the *signed difference vector* of x between conditions j and k. A signed difference vector is a vector of *signs* such as, $(+, -, 0, +, \ldots)$. Dunn and James (2003) showed that if $x = h \circ g'(u)$ for a given linear g', then the set of signed difference vectors of x is completely determined by the linear subspace defined by the model matrix of g'. This subspace partitions the set of all possible sign vectors into a *permitted* set, P, and a *forbidden* set, $\bar{P}$ (Dunn & Anderson, submitted) and leads to the following, in principle, test of a model:

Signed difference test

Let $x = \{x_1, \ldots, x_n\}$ be a set of dependent variables and let $x_{(j)} = \{x_{1(j)}, \ldots, x_{n(j)}\}$, be the set of values of x observed under condition j. Let $X = \{x_{(1)}, \ldots, x_{(k)}\}$. Let $h \circ g'$ be a model, M, of x for a monotonic measurement mapping, h, and a given structural mapping, g'. Let P be the set of sign vectors permitted under M. If there exists $x_{(j)}, x_{(k)} \in X$ such that $\text{sign}(x_{(j)} - x_{(k)}) \notin P$ then M is not a model of x.

The signed difference test for one latent variable is equivalent to STA. In this case, if there are $n = 2$ dependent variables then the one-dimensional model proposed by STA partitions the set of 9 possible sign vectors into the permitted set,

$$P = \{(0,0), (0,+), (0,-), (+,0), (-,0), (+,+), (-,-)\}$$

and the forbidden set,

$$\bar{P} = \{(+,-), (-,+)\}$$

It is apparent that the sign vectors in P are consistent with the monotonicity constraints given in Equation (4.5).

Recent work by Stephens et al. (in press) has extended the conjoint monotonic regression problem given by Equation (4.5) to fit a linear structural model to data according to SDA. They used a version of this extended algorithm to fit a set of models to data obtained from a large number of studies of human reasoning. These data consisted of four dependent variables, defined by the endorsement rates of the validity of deductively valid and invalid arguments under two different argument evaluation instructions, observed over a range of different conditions. The models differed in the number of latent variables (from one to four) and in the forms of their structural mappings (although all were linear). Following their analysis, Stephens et al. were able to reject all the proposed models except two – a testable model with three latent variables and a saturated (hence, untestable) model with four latent variables.

9.3 Syntactic vs. semantic model tests

A feature of STA is that it provides a test of a single latent variable model without attempting to characterize this variable in any way. For example, the single latent variable signal detection model discussed in Section 1.3 was motivated by an account in which the latent variable in question was identified as a response criterion. But such an interpretation plays no part in the test of this model using STA (or SDA for that matter). We therefore draw a distinction between the syntactic (or formal) properties of a model and its semantic (or substantive) properties.

In the present context, the syntactic properties of a model relate to the number of latent variables it has and, if there is more than one, how these variables combine under each dependent variable. The semantic properties of a model relate how these latent variables are interpreted and, in so doing, how each relates to features of the external world – namely, how each latent variable is affected by different independent variables. For example, in the signal detection model given by Equation (1.1), the parameters c and d' are assigned meanings (response criterion and discriminability, respectively) within the framework of the model with the expectation that they will be appropriately affected by relevant independent variables. For example, if this semantic interpretation is correct then c should be affected by the kinds of independent variables likely to affect a response criterion, such as changes in payoffs or the probability of a target. Similarly, d' should be affected by the kinds of independent variables likely to affect a discriminability, such as signal-to-noise ratio.

STA is indifferent to the semantic properties of a model. This means that it does not distinguish between formally equivalent models even if they have different substantive interpretations. For example, consider the following two signal detection models:

Model 1:

$$FAR = F(-c)$$
$$HR = F(1 - c)$$

Model 2:

$$FAR = F(d'/2)$$
$$HR = F(d')$$

Model 1 states that the hit rate (HR) and false alarm rate (FAR) are affected only by changes in the response criterion. Model 2 states that they are affected only by changes in discriminability. From the point of view of STA, these two models are indistinguishable because they have equivalent syntactic properties – they are both single latent variable or one-dimensional

models. On the other hand, it is relatively straightforward to distinguish between these models based on their substantive properties.

Why then are we interested in testing the syntactic properties of a model? Because such a test is logically prior to a test of its semantic properties. If it can be shown, using STA, that a one-dimensional model is inconsistent with the data then further investigation, to determine which substantive single latent variable model (e.g., Model 1 or Model 2) is more viable, is simply irrelevant. Here lies the power of STA. By testing the syntactic properties shared by an entire class of models, it is able to falsify all possible exemplars and demonstrate the necessity of some (unspecified) member of another (more complex) class of models.

The distinction between the syntactic and semantic properties of a model has implications for the interpretation of the dimensionality of the state-trace. Throughout, we have used the term 'latent variable' simply as a label for components of the general structure introduced in Section 1.2. However, terminology is more critical when we attempt to ascribe meaning to the dimensionality of the state-trace. In Section 1.1, we motivated STA with reference to the aim of psychological research as identifying (and characterizing) different cognitive 'mechanisms', 'processes' or 'processing systems'. Bearing in mind that none of these terms is well-defined, theorists are nevertheless often tempted to use such terms to characterize the latent structure revealed by STA. This is acceptable, in our view, as long as this ascription of meaning is kept distinct from what STA itself is able to achieve. STA reveals the dimensionality of the state-trace, not the existence of two or more mental processes (mechanisms, systems, etc.). If the dimensionality is greater than one then, in terms of the the general structure introduced in Chapter 1, it can be inferred that more than one latent variable mediates the effects of the set of independent variables on the set of dependent variables. However, the interpretation and characterization of these latent variables lies outside of STA *per se*.

9.4 From hypothesis to theory

As noted in Chapter 1, STA is a procedure to determine whether one or more than one latent variable mediates the effects of one or more independent variables on two or more dependent variables. We can unpack this aim in order to distinguish between the test of a specific hypothesis and the test of a broader theory.

Throughout this volume, we have focused on the question of testing the hypothesis that the dimensionality of the state-trace is equal to one. That is whether, across selected levels of a given set of independent variables, the dimensionality of the observed state-trace is greater than one. However, theoretical interest usually extends beyond this fairly narrow question. In

terms of the theoretical debates that motivate much of our discussion, interest lies in whether a set of dependent variables depend upon one or more than one latent variable in an absolute sense. For example, the dual-process model, developed by Yonelinas (1994), proposes that recognition memory depends upon two latent variables called 'recollection' and 'familiarity'. Although, in general, both components are viewed as involved in recognition memory, the theory allows that under some circumstances – across the levels of some independent variables – only one or other component may be affected. In other words, this theory can accommodate the finding that a state-trace, in any one instance, may be one-dimensional. Indeed, such an occurrence may be informative concerning the nature of the hypothesized latent variables. Alternatively, a one-dimensional state-trace may also be ascribed to functional dependence of the input mapping (see Section 1.3). Again, this too may be informative about the characteristics of the proposed latent variables.

These considerations lead us to the view that a 'single latent variable' theory cannot be proved in principle. Even if the hypothesis that an observed state-trace is one-dimensional is accepted (as would be the case using the Bayesian methods outlined in Chapter 8), this in itself does not refute any, more general, theory involving more than one latent variable. On the other hand, if the hypothesis that an observed state-trace is two-dimensional is accepted then this *does* refute any and all one-dimensional theories. This logic conforms ineluctably to classical Popperian falsificationism. While the simpler theory can be refuted, the more complex theory never can[1].

It is important to note that this inherent asymmetry in theory testing does not easily admit to a formal analysis. Mechanistic Bayesian updating fails because the same experimental situation is never repeated. In fact, the goal is usually to vary experiments in such a way as to explore the extent of LV space in order to determine its dimensionality. A single latent variable theory asserts that the *entirety* of LV space is one-dimensional which, in order to be confirmed, dictates an exhaustive search. A simple analogy is provided by the classical contrast between a theory that asserts that all swans are white with the alternative that some swans are not. If a non-white swan were reliably observed then the first theory is refuted. But no observation of the colours of swans can refute the second theory. Nevertheless, if over the passage of time, many searches for non-white swans are made and none are found, confidence in the more complex theory is likely to decline although it will never exclude the possibility of a non-white swan being found in the next place that is searched (Australia, for example). Much hinges on the nature of the search and whether all possible locations are sampled appropriately. This is analogous to single and multiple latent variable theories in STA. While the single latent variable theory is refuted by a two-dimensional state-trace, no number of one-dimensional state-traces can refute the multiple latent variable

[1] Although in SDA, a two-dimensional model may be rejected in favor of a three-dimensional model and so on.

theory. Yet, a chain of such observations may erode confidence in this theory with the proviso that much depends on the nature of the search.

The solution to theory asymmetry can only be an appeal to parsimony. If all the evidence (gathered so far) is equally consistent with both a simple, single latent variable model and a more complex, multiple latent variable model, then, despite the fact that the complex model cannot be refuted, the simpler model is still to be preferred (c.f., Russell's teapot, Russell, 1952). Further, notwithstanding the objection of a selective search, this preference may increase as the results of future studies, designed to reveal evidence of a two-dimensional state-trace, fail to do so. It is ever thus.

Appendix

The following list is a summary of the STACMR functions in Matlab (equivalent functions are available in R).

`stats = staSTATS (data, shrink)`
Returns descriptive statistics of continuous data

`staPLOT (data, Name, Value)`
Generates a state-trace plot for continuous data

`[pred, fval, shrinkage] = staMR (data, partial, shrink)`
Conducts monotonic regression according to partial order and covariance shrinkage option for continuous data

`[pred, fval, shrinkage] = staCMR (data, partial, shrink)`
Conducts conjoint monotonic regression according to partial order and covariance shrinkage option for continuous data

`[pval, fval, fits] = staMRFIT (data, Name, Value)`
Calculates empirical p-value of fit of partial order model to continuous data

`[pval, fval, fits] = staCMRFIT (data, Name, Value)`
Calculates empirical p-value of difference in fits of monotonic model and partial order model to continuous data

`stats = staSTATSBN (data, shrink)`
Returns descriptive statistics of binary data

`staPLOTBN (data, Name, Value)`
Generates a state-trace plot for binary data

`[pred, fval, shrinkage] = staMRBN (data, partial, shrink)`
Conducts monotonic regression according to partial order and covariance shrinkage option for binary data

`[pred, fval, shrinkage] = staCMRBN (data, partial, shrink)`
Conducts conjoint monotonic regression according to partial order and covariance shrinkage option for binary data

`[pval, fval, fits] = staMRFITBN (data, Name, Value)`
Calculates empirical p-value of fit of partial order model to binary data

`[pval, fval, fits] = staCMRFITBN (data, Name, Value)`
Calculates empirical p-value of difference in fits of monotonic model and partial order model to binary data

References

Ashby, F. G. (2014). Is state-trace analysis an appropriate tool for assessing the number of cognitive systems? *Psychonomic Bulletin & Review*, *21*(4), 935–946.

Ashby, F. G., Alfonso-Reese, L. A., Turken, A. U., & Waldron, E. M. (1998). A neuropsychological theory of multiple systems in category learning. *Psychological Review*, *105*(3), 442–481.

Ashby, F. G., & Maddox, W. T. (2005). Human category learning. *Annual Review of Psychology*, *56*, 149–178.

Bamber, D. (1979). State-trace analysis: A method of testing simple theories of causation. *Journal of Mathematical Psychology*, *19*(2), 137–181.

Batchelder, W. H., & Riefer, D. M. (1990). Multinomial processing models of source monitoring. *Psychological Review*, *97*(4), 548–564.

Beattie, J. A., Blaisdell, B. E., Kaye, J., Gerry, H. T., & Johnson, C. A. (1941). An experimental study of the absolute temperature scale VIII. The thermal expansion and compressibility of vitreous silica and the thermal dilation of mercury. *Proceedings of the American Academy of Arts and Sciences*, *74*(11), 371–388.

Block, H., Qian, S., & Sampson, A. (1994). Structure algorithms for partially ordered isotonic regression. *Journal of Computational and Graphical Statistics*, *3*, 285–300.

Brainerd, C. J., & Reyna, V. F. (2010). Recollective and nonrecollective recall. *Journal of Memory and Language*, *63*(3), 425–445.

Burdakov, O. P., Dunn, J. C., & Kalish, M. L. (2012). An approach to solving decomposable optimization problems with coupling constraints. In *21st International Symposium on Mathematical Programming* (p. 271). Berlin, Germany.

Burdakov, O. P., Sysoev, O., Grimvall, A., & Hussian, M. (2006). An $O(n^2)$ algorithm for isotonic regression. In G. Di Pillo, & M. Roma (Eds.), *Large-scale nonlinear optimization* (pp. 25–33). Berlin: Springer-Verlag.

Busey, T., Tunnicliff, J., Loftus, G., & Loftus, E. (2000). Accounts of the confidence-accuracy relation in recognition memory. *Psychonomic Bulletin & Review*, *7*(1), 26–48.

Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. (2006). *Measurement error in nonlinear models: A modern perspective*. (2nd ed.). Boca Raton, FL: Chapman & Hall/CRC.

Chang, H. (2004). *Inventing temperature: Measurement and scientific progress*. New York: Oxford University Press.

Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. (2001). Drc: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*, *108*(1), 204–256.

Craik, F. I. M., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior*, *11*(6), 671–684.

Davis-Stober, C. P., Morey, R. D., Gretton, M., & Heathcote, A. (2016). Bayes factors for state-trace analysis. *Journal Of Mathematical Psychology*, *72*, 116–129.

DeCarlo, L. T. (2002). Signal detection theory with finite mixture distributions: Theoretical developments with applications to recognition memory. *Psychological Review*, *109*(4), 710–721.

Dunn, J. C. (2008). The dimensionality of the Remember-Know task: A state-trace analysis. *Psychological Review*, *115*(2), 426–446.

Dunn, J. C., & Anderson, L. (submitted). Signed Difference Analysis: Testing for structure under monotonicity. *Journal of Mathematical Psychology*, .

Dunn, J. C., & James, R. N. (2003). Signed difference analysis: Theory and application. *Journal of Mathematical Psychology*, *47*(4), 389–416.

Dunn, J. C., Kalish, M. L., & Newell, B. R. (2014). State-trace analysis can be an appropriate tool for assessing the number of cognitive systems: A reply to Ashby (2014). *Psychonomic Bulletin & Review*, *27*(4), 947–954.

Dunn, J. C., & Kirsner, K. (1988). Discovering functionally independent mental processes: The principle of reversed association. *Psychological Review*, *95*(1), 91–101.

Dunn, J. C., & Kirsner, K. (2003). What can we infer from double dissociations? *Cortex*, *39*, 1–7.

Dunn, J. C., Newell, B. R., & Kalish, M. L. (2012). The effect of feedback delay and feedback type on perceptual category learning: The limits of multiple systems. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *38*(4), 840–859.

Evans, J. S. B. T. (2003). In two minds: Dual-process accounts of reasoning. *Trends in Cognitive Sciences*, *7*(10), 454–459.

Goel, V., Buchel, C., Frith, C., & Dolan, R. J. (2000). Dissociation of mechanisms underlying syllogistic reasoning. *NeuroImage*, *12*(5), 504–514.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, *15*(1), 20–25.

Heit, E., & Rotello, C. M. (2010). Relations between inductive reasoning and deductive reasoning. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *36*(3), 805–812.

Jang, Y., & Nelson, T. O. (2005). How many dimensions underlie judgments of learning and recall? Evidence from state-trace methodology. *Journal of Experimental Psychology: General*, *134*(3), 308–326.

Kahana, M. J., & Wingfield, A. (2000). A functional relation between learning and organization in free recall. *Psychonomic Bulletin & Review*, *7*(3), 516–521.

Kalish, M. L., Dunn, J. C., Burdakov, O. P., & Sysoev, O. (2016). A statistical test of the equality of latent orders. *Journal of Mathematical Psychology*, *70*, 1–11.

Kruskal, J. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, *29*(2), 115–129.

Laguesse, R., Dormal, G., Biervoye, A., Kuefner, D., & Rossion, B. (2012). Extensive visual training in adulthood significantly reduces the face inversion effect. *Journal of Vision*, *12*(10), 1–13.

Laming, D. (1973). *Mathematical psychology*. New York: Academic Press.

Ledoit, O., & Wolf, M. (2004). Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management*, *30*(4), 110–119.

de Leeuw, J., Hornik, K., & Mair, P. (2009). Isotone optimization in R: Pool-adjacent-violators algorithm, (PAVA) and active set methods. *Journal of Statistical Software*, *32*(5), 1–24.

Lewandowsky, S., Oberauer, K., Yang, L.-X., & Ecker, U. K. H. (2010). A working memory test battery for Matlab. *Behavior Research Methods*, *42*(2), 571–585.

Lewandowsky, S., Yang, L.-X., Newell, B. R., & Kalish, M. L. (2012). Working memory does not dissociate between different perceptual categorization tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *38*(4), 881–904.

Loftus, G. R. (1978). On interpretation of interactions. *Memory and Cognition*, *6*(3), 312–319.

Loftus, G. R., & Bamber, D. (1990). Learning-forgetting independence, unidimensional memory models, and feature models: Comment on Bogartz, 1990. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *16*(5), 916–926.

Loftus, G. R., & Irwin, D. E. (1998). On the relations among different measures of visible and informational persistence. *Cognitive Psychology*, *35*(2), 135–199.

Loftus, G. R., Oberg, M. A., & Dillon, A. M. (2004). Linear theory, dimensional theory, and the face-inversion effect. *Psychological Review*, *111*(4), 835–863.

Loomis, L. H., & Sternberg, S. (1990). *Advanced calculus*. Boston: Jones and Bartlett.

Luss, R., Rosset, S., & Shahar, M. (2010). Decomposing isotonic regression for efficiently solving large problems. In J. Lafferty (Ed.), *Advances in Neural Information Processing 23 (NIPS 2010)* (pp. 1513–1521). Curran Associates Inc.

Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A user's guide*. Mahwah, N.J.: Erlbaum.

Maxwell, W. L., & Muchstadt, J. A. (1983). Establishing consistent and realistic order intervals in production-distribution systems. *Operations Research*, *33*, 1316–1341.

Nakabayashi, K., Burton, A. M., Brandimonte, M. A., & Lloyd-Jones, T. J. (2012). Dissociating positive and negative influences of verbal processing on the recognition of pictures of faces and objects. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *38*(2), 376–390.

Newell, B. R., & Dunn, J. C. (2008). Dimensions in data: Testing psychological models using state-trace analysis. *Trends in Cognitive Sciences*, *12*(8), 285–290.

Newell, B. R., Dunn, J. C., & Kalish, M. (2011). Systems of category learning: Fact or fantasy? In B. H. Ross (Ed.), *Psychology of Learning and Motivation. Volume 54* (pp. 167–215). San Diego, CA: Academic Press.

Parks, C. M., & Yonelinas, A. P. (2007). Moving beyond pure signal-detection models: Comment on Wixted, 2007. *Psychological Review*, *114*(1), 188–202.

Pollack, I., & Norman, D. A. (1964). A nonparametric analysis of recognition experiments. *Psychonomic Science*, *1*, 125–126.

Pratte, M. S., & Rouder, J. N. (2012). Assessing the dissociability of recollection and familiarity in recognition memory. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *38*(6), 1591–1607.

Price, C. J. (2010). The anatomy of language: A review of 100 fMRI studies published in 2009. *Annals of the New York Academy of Sciences*, *1191*(1), 62–88.

Prince, M., Brown, S., & Heathcote, A. (2012a). The design and analysis of state-trace experiments. *Psychological Methods*, *17*(1), 78–99.

Prince, M., Hawkins, G., Love, J., & Heathcote, A. (2012b). An R package for state-trace analysis. *Behavior Research Methods*, *44*(3), 644–655.

Rajah, M. N., & D'Esposito, M. (2005). Region-specific changes in prefrontal function with age: A review of PET and fMRI studies on working and episodic memory. *Brain*, *128*(9), 1964–1983.

Reif, F. (1965). *Fundamentals of statistical and thermal physics*. New York: McGraw-Hill.

Robertson, T., Wright, F. T., & Dykstra, R. L. (1988). *Order restricted statistical inference*. Chichester, UK: John Wiley & Sons.

Roundy, R. (1986). A 98% effective lot-sizing rule for a multiproduct multistage production/inventory system. *Mathematics of Operations Research*, *11*, 699–727.

Russell, B. (1952). Is there a God? In R. G. Slater (Ed.), *The collected papers of Bertrand Russell, Vol. 11: Last philosophical testament, 1943-68* (pp. 547–548). Routledge.

Russell, B. (1956). The philosophy of logical atomism. In R. C. Marsh (Ed.), *Logic and knowledge* (pp. 175–282). London: Allen & Unwin.

Shallice, T. (1988). *From neuropsychology to mental structure*. Cambridge, UK: Cambridge University Press.

Shiffrin, R. M., & Steyvers, M. (1997). A model for recognition memory: REM—retrieving effectively from memory. *Psychonomic Bulletin & Review*, *4*(2), 145–166.

Singmann, H., & Klauer, K. C. (2011). Deductive and inductive conditional inferences: Two modes of reasoning. *Thinking & Reasoning*, *17*(3), 247–281.

Stephens, R. G., Dunn, J. C., & Hayes, B. K. (in press). Are there two processes in reasoning? The dimensionality of inductive and deductive inferences. *Psychological Review*, .

Stout, Q. F. (2013). Isotonic regression via partitioning. *Algorithmica*, *66*, 93–12.

Swets, J. A., Tanner, W. P., & Birdsall, T. G. (1961). Decision processes in perception. *Psychological Review*, *68*(5), 301–340.

Valentine, T. (1988). Upside-down faces: A review of the effect of inversion upon face. *British Journal of Psychology*, *79*(4), 471–491.

Verde, M. F., & Rotello, C. M. (2007). Memory strength and the decision process in recognition memory. *Memory & Cognition*, *35*(2), 254–262.

Vilberg, K. L., & Rugg, M. D. (2008). Memory retrieval and the parietal cortex: A review of evidence from a dual-process perspective. *Neuropsychologia*, *46*(7), 1787–1799.

Wagenmakers, E.-J., Krypotos, A.-M., Criss, A. H., & Iverson, G. J. (2012). On the interpretation of removable interactions: A survey of the field 33 years after Loftus. *Memory & Cognition*, *40*(2), 145–160.

Wagenmakers, E.-J., Ratcliff, R., Gomez, P., & Iverson, G. J. (2004). Assessing model mimicry using the parametric bootstrap. *Journal of Mathematical Psychology*, *48*(1), 28–50.

Wasserstein, R. L., & Lazar, N. A. (2016). The ASA's statement on p-values: Context, process, and purpose. *The American Statistician*, *70*(2), 129–133.

Wixted, J. T., & Mickes, L. (2010). A continuous dual-process model of remember/know judgments. *Psychological Review*, *117*(4), 1025–1054.

Yin, R. K. (1969). Looking at upside-down faces. *Journal of Experimental Psychology*, *81*(1), 141–145.

Yonelinas, A. P. (1994). Receiver-operating characteristics in recognition memory: Evidence for a dual-process model. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *20*(6), 1341–1354.

Yonelinas, A. P. (2002). The nature of recollection and familiarity: A review of 30 years of research. *Journal of Memory and Language*, *46*(3), 441–517.