Computational Approaches to Cognition and Perception

Brandon M. Turner
Birte U. Forstmann
Mark Steyvers

Joint Models of Neural and Behavioral Data

Springer

Computational Approaches to Cognition and Perception

Editor-in-chief
Amy H. Criss, Department of Psychology, Syracuse University, Syracuse, New York, USA

Computational Approaches to Cognition and Perception is a series that aims to publish books that represent comprehensive, up-to-date overviews of specific research and developments as it applies to cognitive and theoretical psychology. The series as a whole provides a rich foundation, with an emphasis on computational methods and their application to various fields of psychology. Works exploring decision-making, problem solving, learning, memory, and language are of particular interest. Submitted works will be considered as well as solicited manuscripts, with all be subject to external peer review.

Books in this series serve as must-have resources for Upper-level undergraduate and graduate students of cognitive psychology, theoretical psychology, and mathematical psychology. Books in this series will also be useful supplementary material for doctoral students and post-docs, and researchers in academic settings.

More information about this series at http://www.springer.com/series/15340

Brandon M. Turner • Birte U. Forstmann
Mark Steyvers

Joint Models of Neural and Behavioral Data

Springer

Brandon M. Turner
Department of Psychology
The Ohio State University
Columbus, OH, USA

Birte U. Forstmann
Cognitive Science Center
University of Amsterdam
Amsterdam, The Netherlands

Mark Steyvers
Department of Cognitive Sciences
University of California, Irvine
Irvine, CA, USA

ISSN 2510-1889 ISSN 2510-1897 (electronic)
Computational Approaches to Cognition and Perception
ISBN 978-3-030-03687-4 ISBN 978-3-030-03688-1 (eBook)
https://doi.org/10.1007/978-3-030-03688-1

Library of Congress Control Number: 2018964405

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Kat, the one known quantity amidst the randomness.

BMT

Foreword

Psychologists know well that understanding and predicting human behavior are incredibly difficult. Likewise, neuroscientists are acutely aware of how difficult it is to explain how behavior is implemented by neural mechanisms. One tantalizing possibility is that tackling both of these challenges simultaneously may be more tractable than addressing each separately at its own level of analysis. Although this ambitious and integrative path may seem fantastical, if constraints and relationships exist across measures, then this enterprise has hope.

To make this hope a reality requires powerful and transparent methods to link measures across levels of analysis in a way that can support inference and model comparison. The authors provide just such a recipe in this book, walking the reader through the motivation, background, math, and code and how to interpret results step-by-step. The solution is joint modeling, which simultaneously analyzes multiple brain and behavioral measures within a shared formal framework, allowing their relationships to inform model fitting and inference.

With joint modeling, one can potentially predict behavior in a task better by incorporating brain measures. For example, trial-by-trial fluctuations in EEG signals could indicate how prepared a person is to respond, which could capture variance in response times. Likewise, multiple brain measures could be incorporated to take advantage of the relative strengths of each. For example, fMRI has good spatial resolution but poor temporal resolution, whereas EEG has the mirror pattern of strengths and weaknesses. Joint modeling can pool multiple measures, whether they are behavioral or neural, to improve estimates. When data are missing for a trial, joint modeling can impute the missing value.

In this book, the practicalities of how to do joint modeling are covered, such as how to evaluate whether joint modeling is adding anything on top of considering measures in isolation. The authors start with simple illustrative examples, which include accompanying code, to build the reader's intuitions and ability to formulate their own models. To appreciate what is possible with joint modeling, a chapter is devoted to considering published examples. Finally, potential solutions to future challenges, such as scaling the approach to more complex problems that involve relating numerous model parameters and brain regions, are discussed. The

relationship of joint modeling to alternative approaches concerned with bridging levels of analysis is also covered.

As the field transitions toward considering ever-richer and larger datasets, mastering analysis techniques such as joint modeling will become increasingly important. The authors, who are pioneers and leaders in this area of research, are ideally positioned to guide the reader on this journey. Of course, the journey will be easiest for those with familiarity with basic concepts from graphical models, model fitting and evaluation, and Bayesian methods, but it is also worth the ride for others who are willing to work through these preliminaries using the provided code snippets. Mastering the concepts in this book should be rewarding because this knowledge will provide the means for one to build their own models that link the brain and behavior. Only a decade or so ago, bridging these levels of analysis seemed farfetched, but now, with techniques such as joint modeling, it is within reach. Hopefully, readers of this book can add to this integrative and growing area of research.

London, UK Bradley C. Love
June 2018

Acknowledgments

This work was supported by a National Science Foundation Integrative Strategies for Understanding Neural and Cognitive Systems Collaborative Research Grant (1533500 and 1533661) and by an ERC Starting Grant from the European Research Council.

Contents

About the Authors

Brandon Turner is an Assistant Professor in the Psychology Department at The Ohio State University. His research interests include dynamic models of cognition and perceptual decision-making, efficient methods for performing likelihood-free and likelihood-informed Bayesian inference, and unifying behavioral and neural explanations of cognition. His current focus is on understanding how external factors, such as the environment, and internal factors, such as working memory, interact to shape an observer's perception of the world and ultimately how this perception drives their decisions. Dr. Turner is an Editorial Board Member for the journal *Decision*.

Birte U. Forstmann is a Professor for Cognitive Neurosciences at the University of Amsterdam. She earned her PhD in 2006 at the Max Planck Institute for Human Cognitive and Brain Sciences in Leipzig, Germany. After completing her postdoc in 2008 at the University of Amsterdam, she became tenured Research Fellow at the Cognitive Science Center Amsterdam with the focus of model-based cognitive neurosciences. Since then, she has contributed to a range of topics in cognitive neuroscience, experimental psychology, mathematical psychology, and, lately, also in quantitative neuroanatomy. Despite its diversity, her work is motivated by a single strong conviction, namely, that behavioral data and brain measurements need to make contact with psychological theory via concrete mathematical models of latent cognitive processes.

Mark Steyvers is a Professor in the Department of Cognitive Sciences at the University of California, Irvine. He also holds joint faculty appointments with the Department of Computer Science and the Department of Psychology and Social Behavior at the University of California, Irvine. Dr. Steyvers is Associate Editor for *Computational Brain & Behavior* and Consulting Editor for *Psychological Review* and an Editorial Board Member for *Behavior Research Methods*.

Chapter 1
Motivation

The evolution of technology for measuring brain signals, such as electroencephalography (EEG) and functional magnetic resonance imaging (fMRI), has provided exciting new opportunities for studying mental processes. Today, scientists interested in studying cognition are faced with many options for relating experimentally-derived variables to the dynamics underlying a cognitive process of interest. Figure 1.1 nicely illustrates how many different disciplines can be used independently to understand a cognitive process of interest. These disciplines share the common goal of drawing conclusions about cognitive processes, but each branch has a different vantage point: experimental psychologists focus on behavioral data, mathematical psychologists focus on formal models, and cognitive neuroscientists focus on brain measurements.

While conceptually the presence of these new "modalities" of cognitive measures could have immediately spawned an interesting new integrative discipline, the emergence of such a field has been slow relative to the rapid advancements made in these new technologies. Until a little over a decade ago, much of our understanding of cognition had been advanced by two dominant but virtually non-interacting groups. The largest group, cognitive neuroscientists, relies on models to understand patterns of neural activity brought forth by the new technologies. Like experimental psychologists, the models and methods used by cognitive neuroscientists are typically data-mining techniques, and this approach often disregards the computational mechanisms that might detail a cognitive process. The other group, mathematical psychologists, is strongly motivated by *theoretical* accounts of cognitive processes, and instantiates these theories by developing formal mathematical models of cognition. The models often detail a system of computations and equations intended to characterize the processes assumed to take place in the brain. As a formal test of their theory, mathematical psychologists usually rely on their model's ability to fit and predict behavioral data relative to the model's complexity.

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_1

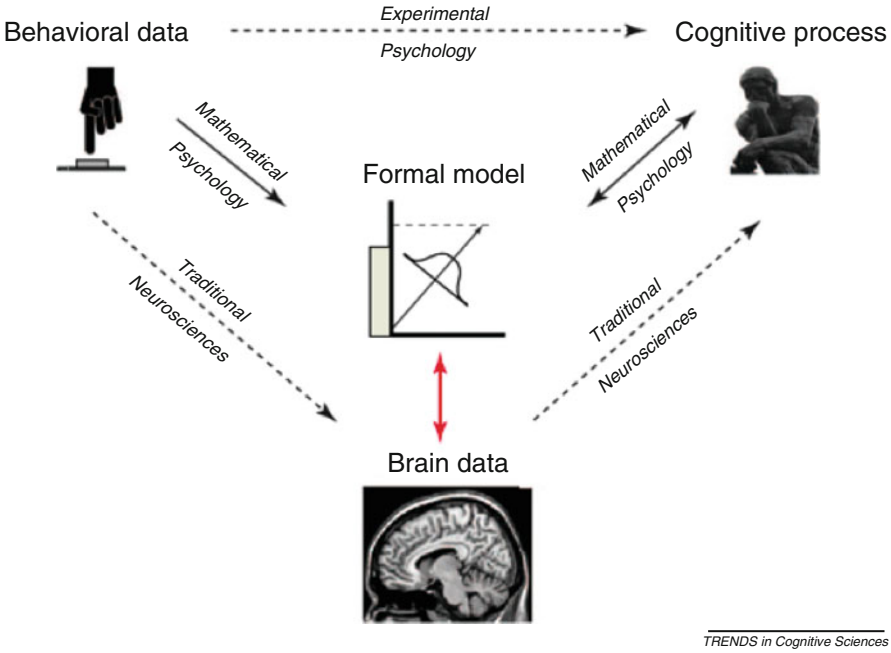


Fig. 1.1 The "model-in-the-middle" approach unifies three different disciplines for understanding the mind through behavioral data. (Taken with permission from Forstmann et al. [1])

A recent trend in cognitive science is to blend the theoretical and mechanistic accounts provided by models in the field of mathematical psychology with the high-dimensional data brought forth by modern measures of cognition [1]. For example, in Fig. 1.1, Forstmann et al. [1] advocated for the use of *reciprocal* relationships between the latent processes assumed by cognitive models and analyses of brain data (i.e., indicated by the red arrow). While conceptually, blending these two fields may seem like the ideal approach, as this book will discuss, it is often not straightforward to impose such a relationship [2, 3] as there are many theoretical, philosophical, and methodological hurdles any researcher must overcome. Yet, the pursuit continues because the payoff is far too enticing to detour some researchers: the notion that agreed upon theoretical and computational mechanisms supporting decision making could be substantiated in the one organ housing mental operations presents a unique opportunity for major advancements in the field of cognitive science.

1.1 Neural Data Can Inform Cognitive Theory

One of the most powerful examples of how computational theories can be advanced by brain data comes from Hanes and Schall [4]. Their goal was to identify how variability in response times from perceptual decisions arise. At the time, the

simplest mathematical models explained variability in response times in one of two ways: the first way posited that the rate of evidence accumulation – referred to as the drift rate – changed from one trial to the next, whereas the second way posited that the amount of evidence – referred to as the threshold – required to make a decision varied from one trial to the next. The left panel of Fig. 1.2 illustrates the two models, where the variable drift rate model is shown in the top panel, and the variable threshold model is shown in the bottom panel. The top panel shows how the rate at which evidence accumulates can be altered from one trial to the next to produce different response times because the accumulation process terminates at different times. The three colored lines are generated with

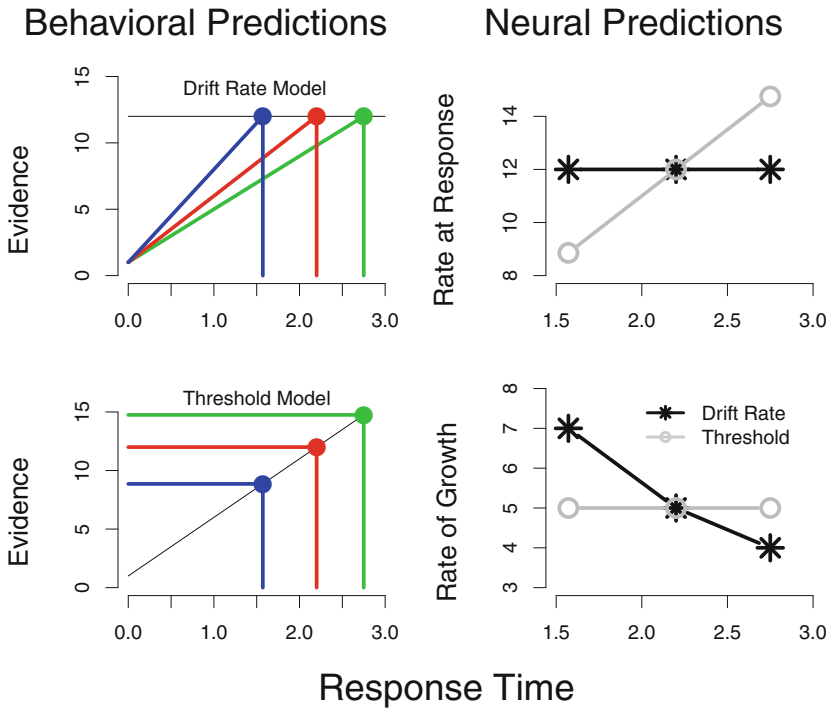


Fig. 1.2 Two explanations of how response time variability manifests. The left panels show three examples (blue, red, and green) of how response times arise as a function of either (top) the rate of evidence accumulation or (bottom) the threshold amount of evidence varies from one trial to the next. For the drift rate model, the three colored lines are generated with different drift rates (i.e., slopes) that accumulate until a fixed threshold (black line). For the threshold model, a single drift rate (black line) accumulates to different threshold amounts of information (colored lines). The right panels illustrate predictions for two statistics (panels) of the neural data from the variable drift rate model (black lines) and the variable threshold model (gray lines). Predictions for the firing rate at the time a decision is made are shown in the top panel (i.e., the "threshold" prediction), whereas predictions for the growth of the firing rate as a function of time is shown in the bottom panel. (Figure adapted from Hanes and Schall [4])

different drift rates (i.e., slopes) that accumulate until a fixed threshold (black line). By contrast, the bottom panel shows how the exact same response times can be explained by assuming a constant rate of evidence accumulation but requiring three different levels of evidence prior to making a response. Here, a single drift rate (black line) accumulates to different threshold amounts of information (colored lines). Either of these two mechanisms can be used to generate response time distributions that closely match empirical data. Unfortunately, as Fig. 1.2 implies, the two mechanisms can also match one another, making the task of ruling one model out reliability on the basis of behavioral data alone impossible. In fact, Dzhafarov [5] showed that under some unrestrictive assumptions, the two sources of variability are mathematically indistinguishable. Hence, these two computational mechanisms describing choice response time were at a perfect theoretical stalemate, where the behavioral data from the task could neither support nor refute the two proposed mechanisms for explaining variability.

Hanes and Schall [4] considered whether other measures of mental activity besides response times could be used to distinguish these two important theories. In their work, they had recorded from neurons in the frontal eye field (FEF), an area that had been implicated as a likely candidate for the biological realization of the evidence accumulation abstraction assumed by mathematical models. Recording from single units is essentially the holy grail of neuroimaging, as the data come directly from neurons – the biological building block of the brain – and have excellent temporal and spatial resolution. Hanes and Schall [4] supposed that if the activity from the neurons they recorded from in the FEF did map to evidence accumulation, then the statistical properties of the firing patterns in the neurons could be used to differentiate the two competing theories of response time variability. Specifically, they focused on (1) the rate of "spikes" (i.e., when a neuron fires) observed within a moving time window, and (2) the total spiking rate of that neuron at the time a decision is made. Assuming these two statistics correspond to the rate of evidence accumulation and the threshold, respectively, the two models make different predictions for what these statistics should look like as a function of decision time. For example, the right panel of Fig. 1.2 shows predictions for the two statistics from each of the models, where black lines correspond to the variable drift rate model, and the gray lines correspond to the variable threshold model (gray lines). The variable drift rate model (black lines) predicts that increases in response times are a consequence of slower spike rates (i.e., the bottom panel), and not changes in the threshold, which should remain constant (i.e., the top panel). On the other hand, the variable threshold model (gray lines) predicts that no change in the growth rate is necessary and so it should remain constant as a function of time (i.e., bottom panel), but when a response time increases, it is captured by an increase in the response threshold for that trial (i.e., top panel). Because the models make strikingly different predictions about the statistics that should be observed in the neuronal firing rate, we need only examine the firing rates from a perceptual experiment to determine which mechanism is most likely implemented in the brain.

Hanes and Schall [4] first binned trials according to the decision time observed. They then computed the two statistics – the "growth" of the firing rate and the level of the firing rate at the time a decision was made – and plotted these statistics as a function of the (binned) decision times. They found that while the threshold firing rate remained stable across time, the growth of the firing rate declined linearly with increases in the response times. Together, these results provided strong evidence that the variability in response times is most likely a consequence of fluctuations in the rate of evidence accumulation rather than changes in the levels of evidence needed to make a decision.

The work of Hanes and Schall [4] effectively demonstrates how researchers can incorporate both theoretical mechanisms and data from multiple measures of mental activity to better understand the processes underlying cognition. From the mathematical psychologist's perspective, no definitive evidence could be gleaned from empirical data regarding the type of mechanism best articulating variability in response times without the help of the neural data. From the cognitive neuroscientist's perspective, one could argue that the appropriate statistics for analyzing the neural data may not have been devised, and more importantly, the statistics may never have been interpreted from a mechanistic perspective without the language of the mathematical model. Hence, while both perspectives provide a unique view of the underlying cognitive process, combining the two approaches provided compelling evidence that neither approach could have achieved alone.

Beyond the conceptual benefits of considering multiple measures of mental activity, there are also major practical benefits in having a wealth of extra information that is provided by neural data. By any measure, the amount of information in behavioral data is limited. In a typical behavioral experiment, the types of measures we obtain are not much more than choices and response times. Literally, the data from a typical behavioral experiment can usually be summarized in a few thousands bytes. On the other hand, data from an experiment where neural measures are obtained can take up to a few billion bytes per subject. Considering this, neural data provide us with a unique opportunity to develop richer models of cognition that are simply impossible with behavioral data alone.

1.2 Statistical Reciprocity Through Joint Models

Key successes such as Hanes and Schall [4] have inspired a wave of researchers to combine neural and behavioral measures in an integrative fashion. The importance of solving the integration problem has spawned several entirely new statistical modeling approaches developed through collaborations between mathematical psychologists and cognitive neuroscientists, collectively forming a new field often referred to as "model-based cognitive neuroscience" [1, 6–19]. This field uses formal cognitive models as tools to isolate and quantify the cognitive processes of interest in order for them to be associated to brain measurements more effectively. However, the field is not limited to a particular modality of neural information;

indeed the field is diverse in its use of brain measurements such as single-unit electrophysiology, magneto/electroencephalography (MEG, EEG), and functional magnetic resonance imaging (fMRI) to address questions about formal models that cannot be addressed from within the models themselves. Similarly, the field is not limited by the types of cognitive models that can be applied to data.

Figure 1.1 illustrates how the field of model-based cognitive neuroscience fits within the extant fields for understanding cognitive processes. The so called "model-in-the-middle" approach [1, 6, 20] attempts to unify these separate disciplines by using formal models as the pivotal element in bridging behavioral data and brain measurements. The endeavor central to this field is that cognitive models and brain measures should be used reciprocally to enhance our understanding of cognitive processes. The mechanisms describing the (latent) cognitive processes are put forth by the mathematical models, whereas the manifestation of the cognitive processes are to be inferred from the neural measures.

While we discuss many other approaches for creating reciprocity in Chap. 6, the purpose of this book is to elaborate on a particular style of enforcing reciprocity through what we call "joint modeling." The models are referred to as "joint" for two reasons. First, each approach explicitly specifies a statistical constraint between the measures, making it a complete model. Second, these models consider the joint distribution of both neural and behavioral measures, making them "joint" models. This distinction is important because it separates joint models from other types of approaches (see Chap. 6). Three types of joint models are illustrated in Fig. 1.3 via graphical diagrams, where observed variables (e.g., N and B) are shown as filled square nodes, and parameters are shown as empty circles. Paths between the nodes in the graph indicate dependency among the nodes, where an arrow pointing from one node to another indicates a "parent-to-child" ancestry [21]. In other words, the node being pointed at depends on the node from which the arrow originates. Although the three types of joint models can be illustrated with similar

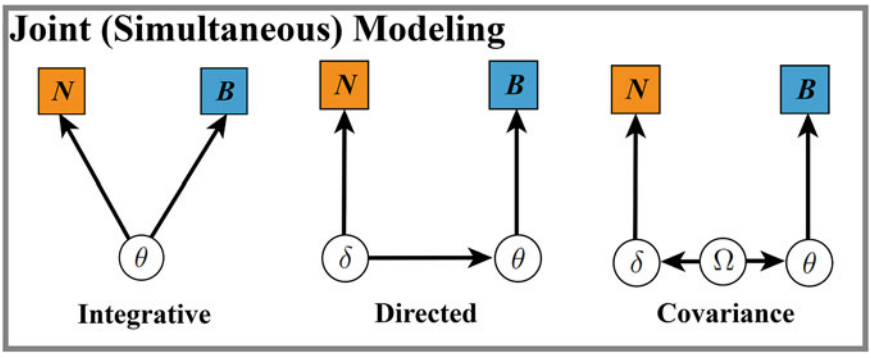


Fig. 1.3 An illustration of the three joint modeling approaches for linking neural and behavioral data. N represents the neural data, B represents the behavioral data, and θ, δ, and Ω represent model parameters

graphical diagrams, the structures introduce different constraints, which have major implications for a joint model's complexity relative to the observed data. We now discuss each of the three classes of joint models in Fig. 1.3.

1.2.1 Integrative Approach

In the Integrative approach, a single cognitive model is developed to predict both neural and behavioral measures simultaneously. This approach, illustrated in the left panel of Fig. 1.3, uses a single set of parameters θ to explain the neural N and behavioral B data together, indicated by connections from θ to N and θ to B. In some cases, Integrative models can also use a set of modulators to transform an internal state of a model into a prediction about the precise functional form of the neural measures. For example, different modulators would be necessary to make predictions for a BOLD response in an fMRI study versus predictions for an evoked response potential in an EEG study.

Some recent applications of ACT-R have aimed for this Integrative approach. ACT-R assumes the presence of distinct cognitive modules that are recruited sequentially during a task. The recruitment of these modules across the time course of the task can be represented as a vector of binary outcomes, such that a 1 indicates that a module is being used, and a 0 indicates it is not being used. This vector naturally lends itself to convolution with the canonical HRF in the same way as experimental design variables (i.e., called the design matrix). The result of the convolution is a model-generated BOLD signal that can be compared to empirical data. In this way, the ACT-R model can actually be used in both exploratory and confirmatory research. When used for exploration, the model-generated BOLD signal is regressed against the data in a voxel-by-voxel fashion through the general linear model [22, 23]. From this analysis, clusters of voxels typically emerge, and these clusters are taken to represent brain areas where the modules are physically executed. This explorative analysis more closely resembles the Latent Input Approach, and has the added benefit of avoiding strong commitments to where a mechanism in the model is carried out in the brain.

The ACT-R model can also be used in a confirmatory fashion, making it a fully Integrative model [24–27]. To do this, Anderson and colleagues first used an approach to identify which brain areas should become active during the recruitment of different modules within a task [24, 26, 28]. Next, Anderson and colleagues used the voxels discovered in their exploratory analysis to specify the locus of brain activation in an out-of-sample prediction for neural data from the ACT-R model. Although global, whole-brain predictions could be made within this framework, the strict assumption of localized module activity in the brain constitutes a confirmatory Integrative approach, where predictions for neural activity – as well as behavioral data – can be quantitatively evaluated. We refer the reader to Borst and Anderson [29] for a tutorial on exploratory and confirmatory analyses with the ACT-R model.

Integrative joint models are arguably the most difficult to develop as they require strong commitments to not only the underlying cognitive process, but also where this process is executed in the brain. Beyond theoretical issues, one technical hurdle in using the Integrative approach is in the description of random variables with different temporal properties. For example, neurophysiological measures are typically observed on a moment-by-moment basis, detailing activation in the brain throughout the trial. By contrast, behavioral data are typically observed only at the end of a trial, such as in any number of perceptual decision making tasks. Hence, to instantiate a cognitive theory via the Integrative approach, we need a moment-by-moment prediction of neural data, and a trial-by-trial prediction of the behavioral data, usually assumed to be the result of a series of unobservable (i.e., latent) processes. Given the unique structure of Integrative approaches, properly fitting them to data is a difficult task, often involving sophisticated techniques such as Hidden Markov Models [30, 31], or Bayesian change point analyses [32].

1.2.2 Directed Approach

One way of sidestepping the technical hurdles in the Integrative approach is to rely on modulators of certain parameters to predict the neural signal. We refer to these models as "Directed" models to make a connection to the Direct Input models described above. The middle panel illustrates one example of the Directed approach. On the neural side, the parameters δ describe how the neural data N come about, usually through some statistical model. However, the parameters δ also detail how the behavioral model parameters θ are specified through some linking function $\mathscr{M}$, such that

$$\theta = \mathscr{M}(\delta).$$

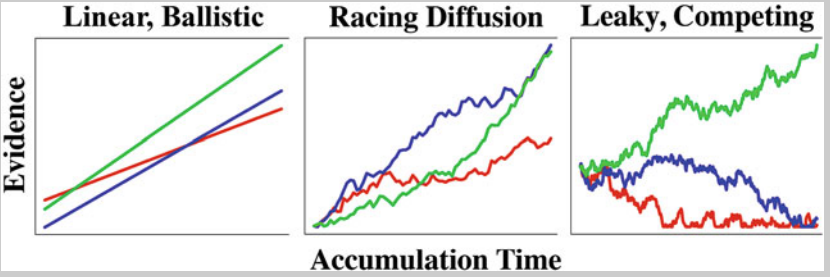
Usually, the linking function $\mathscr{M}$ will also consist of a set of modulating variables to allow flexibility in the mapping from δ to θ (e.g., if they are on different scales). Once specified, the parameters θ describe the behavioral data B through a cognitive model.

One example of Directed joint models come from recent applications of the Diffusion Decision model to neural data [33–35]. (See Box 1.2.2 for a description of the LBA model, as well as other models of choice response time.) For example, Nunez et al. [33] used EEG data on a perceptual decision making experiment as a proxy for attention. They controlled the rate of flickering stimuli presented to subjects to match the sampling rate of their EEG data, a measure known as the steady-state visual evoked potential. Importantly, Nunez et al. [33] showed that individual differences in attention or noise suppression was indicative of the choice behavior, specifically it resulted in faster responses with higher accuracy. In a particularly novel application, Frank et al. [34] showed how models of reinforcement learning could be fused with the DDM to gain insight into activity in the subthalamic nucleus

(STN). In their study, Frank et al. [34] used simultaneous EEG and fMRI measures as a covariate in the estimation of single-trial parameters. Specifically, they used pre-defined regions of interest including the presupplementary motor area, STN, and a general measure of mid-frontal EEG theta power to constrain trial-to-trial fluctuations in response threshold, and BOLD activity in the caudate to constrain trial-to-trial fluctuations in evidence accumulation. Their work is important because it establishes concrete links between STN and pre-SMA communication as a function of varying reward structure, as well as a model that uses fluctuations in decision conflict (as measured by multimodal activity in the dorsomedial frontal cortex) to adjust response threshold from trial-to-trial. These models specify that certain components of the model, such as the drift rate or response threshold, are completely determined by the neural data and a set of modulating parameters. Instead of estimating the model parameters in the typical fashion, the modulating parameters are the ones being estimated across trials. Due to the deterministic mapping, models with this type of structure are unidentifiable without neural data, presenting both advantages and disadvantages to the types of problems to which the model can be applied.

Popular Models of Choice Response Time

There are now many models that are flexible enough to capture important trends in choice response time data. Three popular models are the Linear Ballistic Accumulator (LBA; [36]) model, the Diffusion Decision Model (DDM; [37]), and the Leaky, Competing Accumulator (LCA; [38]) model. These models make a number of different processing assumptions, and the figure below illustrates a few of these important differences. One can view the models as having similar architectures, but with increasing degrees of complexity (arranged in increasing order from left to right).



Linear Ballistic Accumulator Model: The left panel shows a graphical representation of the LBA model for two-choice data. Each response option is represented as a single accumulator (i.e., the red, blue, and green lines). Following the presentation of a stimulus, evidence ballistically accumulates

(continued)

for each alternative until one of the alternatives reaches the threshold (top line). The model assumes some initial amount of evidence is present for each response option, and this amount is randomly distributed across trials. The rate of evidence accumulation itself is also randomly distributed across trials, but has a mean that is fixed allowing one option to be chosen systematically over other options. The accumulation process in the model is linear, and each alternative accumulates information independently, meaning that the state of one accumulator does not depend on any others.

Diffusion Decision Model: The middle panel illustrates a racing diffusion process [39], which is a more general case of the DDM. In the racing diffusion process, evidence for each alternative accumulates independently, as in the LBA. However, the DDM assumes that evidence accumulates in a perfectly anti-correlated fashion, meaning that evidence for one alternative is evidence against the other alternative. This feature of the DDM makes it difficult to apply directly to multi-alternative choice. The DDM adds to the LBA an assumption about within-trial variability in the accumulation process. The middle panel illustrates this stochastic process by the wavy paths through evidence space as a function of time.

Leaky Competing Accumulator Model: The LCA model was developed as a neurologically plausible way to describe the dynamics of response competition. Within the LCA, several nonlinearities complicate the accumulation process. Most importantly, the accumulators compete with one another in a way that is state-dependent: as one accumulator gathers more evidence, it can inhibit other accumulators, causing their rate of accumulation to slow and even become negative. In the illustration above, this competitive dynamic can be seen by inspecting the interaction of the accumulators, where the green accumulator dominates first the red accumulator, and later the blue accumulator. Like the DDM, the LCA assumes within-trial variability. Traditional applications of the LCA do not usually assume between-trial variability in the drift rate, and only occasionally assume between-trial variability in starting point. The LCA model also assumes that the accumulation of evidence is "leaky", meaning that some information is lost during the integration of sensory information.

While the middle panel of Fig. 1.3 shows that the parameter δ specifies the parameters θ, the opposite direction of influence could obviously be made such that the parameters θ could specify the parameters δ. As one example, van Ravenzwaaij et al. [40] applied a Directed joint model to data from a mental rotation task. Here, they used the Linear Ballistic Accumulator (LBA) model [36] to describe the behavioral data, and used the drift rate parameter in the LBA model, combined with some modulating parameters, to describe the mean of an EEG signal through time. Here, because the modulating variables transform the drift rate parameters to describe the mean parameter of a normal distribution, we consider their approach

a Directed approach, although it has some similarities to the Integrative approach above.

The distinction between Integrative and Directed joint models is a tricky one, but this distinction can be better realized by considering the differences in the model architecture between say, the ACT-R model and the LBA model. The ACT-R framework requires a "pipeline" of module activation over time. One issue is the development of the modules themselves, but supposing we had these modules in hand, inferring when a module has become active is a difficult problem. Considering only behavioral data, identifying module activation would be nearly impossible without some form of behavioral measure over time. However, the inclusion of neural data and sophisticated statistical techniques can isolate neural signatures that can be used to articulate what module activation should look like in the neural data [29–31]. Unfortunately, these accomplishments seem to be specific to the models using an ACT-R like architecture [41–43] because it is unclear how one would objectively decompose other cognitive models into a discrete set of modules while preserving their key theoretical and convenient properties. For example, the LBA model has enjoyed widespread success due to its parsimony and elegant mathematical tractability. Breaking the LBA model down into its constituent parts could compromise this tractability in such a way that estimation of the model's parameters would be nontrivial [44]. Hence, it is clear that not every cognitive model can easily be transformed and prepared for an analysis using the Integrative Approach. The approach presented in [40] represents a commendable attempt to constrain the LBA model further with EEG data through the use of modulating variables, rather than decomposing the structure of the model itself.

1.2.3 Covariance Approach

The final approach we discuss is Covariance approach [12, 18, 45, 46]. The Covariance approach is conceptually similar to the Directed approach as it describes the joint distribution of the behavioral model parameters θ and the neural model parameters δ through a statistical constraint. However, unlike the Directed approach, the Covariance approach assumes an overarching distribution to describe the patterns present in the joint distribution of (θ, δ) across the levels to which the parameters (θ, δ) are applied. For example, if (θ, δ) describe the effects at the trial level, the overarching distribution linking θ to δ would describe how these parameters were related in say, an experimental block or a given subject. This connection is important because it allows the information contained in the neural data N to affect the information we learn about the behavioral model parameters θ, and vice versa.

The right panel of Fig. 1.3 illustrates the overarching connection between the parameters θ and δ through the parameters Ω. In this illustration, the connection enforced by Ω is clearly abstract; one must make a specific assumption about how θ and δ should coexist in their explanation of the underlying cognitive process. In

other words, we must specify an expression for how θ and δ are related through the linking function $\mathscr{M}$ with parameters Ω:

$$(\theta, \delta) \sim \mathscr{M}(\Omega).$$

As an example, one simple linking function used in practice has been the multivariate normal distribution where Ω consists of the hyper mean vector and the hyper variance-covariance matrix.

The main advantage the Covariance approach has over the Directed approach is in the treatment of the model parameters θ and δ. Unlike the Directed approach, the Covariance approach maintains that parameters like θ are latent, and not transformations of neural parameters δ or aspects of the neural data N. Hence, while having neural data N does enforce a constraint on θ [45], the neural data N are not essential to identifying the parameter θ. For example, suppose we observed an extreme outlier for some neural measure N. In the Directed approach, we are tied to the assumption about how the parameter δ, inferred from the data N, maps to a prediction about the behavioral data B. However, because the neural measure N is an outlier, it might cause a mapping to B that is completely senseless. On the other hand, Covariance models specify the random behavior of N and outliers observed in N would simply inflate the variability terms in the overarching parameters Ω. As a result, the predictions about the behavioral data B would be mitigated by the rest of the behavioral data. A similar type of phenomenon exists when comparing the accuracy of the parameters estimated by a hierarchical Bayesian model to those estimated by maximum likelihood estimation [47–49].

1.3 Organization

This book will focus on using joint models to fulfill the reciprocity advocated for in Forstmann et al. [1] (Fig. 1.1, red arrow). We begin in Chap. 2 with a tutorial on the three types of joint models discussed above. Here, we focus on concrete, albeit simplistic, applications of joint models, providing both equations and computer code for simulating and fitting your own models to data. In Chap. 3, we focus on an important consideration in evaluating the generalizability of joint models, by discussing out-of-sample tests that can be performed, while also providing examples and code that follow from Chap. 2. In Chap. 4, we highlight a few interesting applications of joint models that have appeared in the literature. In Chap. 5, we discuss a few new directions for joint models that solve some of the existing limitations. In Chap. 6 we discuss some other important approaches for modeling neural and behavioral measures of cognition. Finally, in Chap. 7 we provide some conclusions and closing remarks.

Chapter 2
A Tutorial on Joint Modeling

2.1 The Generative Model

Our tutorial focuses on an illustrative example in the area of episodic memory. One classic experiment used to study episodic memory is the recognition memory paradigm, where subjects are first given a list of words or items and instructed to commit these items to memory. The process of storing specific memories of the presented list is known as encoding, and the time period for encoding is experimentally defined as the study phase. Following the study phase is the test phase where another list of items are presented to the subject sequentially, and the subject is instructed to categorize each item as belonging to the previously studied list – an "old" response – or not – a "new" response. By mixing the types of items presented to subjects at test, we can examine how well the items from the study list were encoded into memory by calculating the number of times a response of "old" was elicited when an item from the original study list was presented to the subject.

Although we can tabulate the proportions of "old" and "new" responses for each item type, it is difficult to understand the process involved in either encoding or retrieval because both of these processes are not directly observable – that is, they are latent. As a consequence, our ability to articulate how well each specific item is stored in memory is somewhat limited, as the behavioral data can only take one of two values, and we observe exactly one response per test item. Could neural data potentially guide our inference of item encoding? Our goal in this tutorial is to use hypothetical information from the brain to enhance a simple cognitive model of trial-by-trial item encoding. Motivated by several studies of episodic memory, we assume that some area of the brain is related to the formation of episodic memories, such that greater activation is related to a higher probability of forming a memory of that item. Although this is a hypothetical example, some plausible brain areas that have been shown to exhibit patterns consistent with the concept of encoding are located in the medial temporal lobe such as perirhinal cortex [51], and hippocampus

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_2

[51, 52]. As an illustrative example, activation of these areas might signal better memories of the study items, and could potentially be used to better articulate how "new" and "old" responses are elicited at test.

For the purposes of our tutorial, we first describe the joint model from a generative perspective by illustrating how hypothetical data can be generated from the model under fixed settings of the parameter values. The application we present below is intended to be simple so as to keep the equations and code necessary to fit the model to data concise. The data we consider are univariate, meaning that both the behavioral data and neural data can be captured by a single parameter. Figure 2.1 shows a graphical diagram for the joint model used in this tutorial chapter. For the joint model considered here, there are three main components of the model: the neural submodel, the behavioral submodel, and the linking function. The left side of Fig. 2.1 corresponds to the neural submodel because it describes how the neural data $N_{i,t}$ are related to the neural model parameters δ_i. The right side of Fig. 2.1 corresponds to the behavioral submodel because it describes how the behavioral data B_i are related to the cognitive model parameters θ_i. The middle part of the Fig. 2.1 describes the linking function by using the hyperparameters ϕ and Σ to connect the two submodel parameters θ and δ. We now turn to a detailed description of each subcomponent of the model. Where appropriate, we refer back to Fig. 2.1 to facilitate the description of each subcomponent of the model as it relates to the full model.

2.1.1 Neural Submodel

Suppose in our example that we obtain blood oxygenated level dependent (BOLD) responses from the brain on every trial i, and we obtain these measures at five points in time t. Suppose that the scanning times consist of the set $T = \{0, 1, 2, 3, 4\}$,

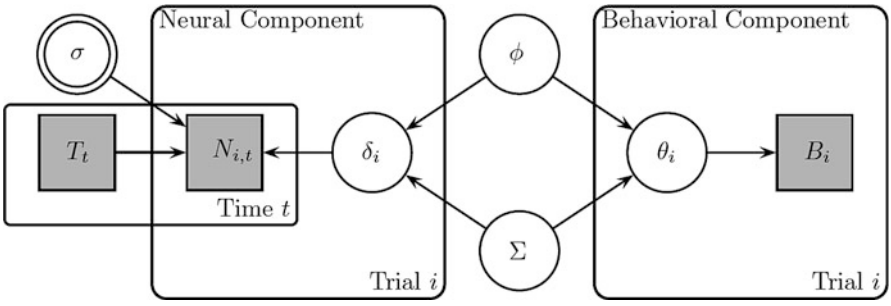


Fig. 2.1 Graphical diagram for the joint model presented in this chapter. Each node represents a variable in the model, where gray nodes correspond to observed variables, white nodes correspond to latent variables, and double-bordered nodes correspond to deterministic nodes (that are not estimated). Paths indicate direct relationships between the variables and plates indicate replications across dimensions (e.g., trials or time)

which might correspond to seconds since a study item was presented. Letting $N_{i,t}$ denote the neural data at time t on trial i as in Fig. 2.1, $N_{i,t}$ might describe the degree of activation of the brain region of interest (ROI) over time. Of course, there are many complex models of brain function that could be used to describe patterns of activation through time or interconnectivity among regions of interest. However, as this chapter will show, one can start with a very simplified model of activation, with the hope that more realistic models can be constructed from the preliminary results. Our goal for this book is to demonstrate how joint modeling can be done even with these simple models, with the hope of facilitating understanding.

While there are many ways to describe brain activation over time, suppose we choose a linear ramping function of the form

$$N_{i,t} = T_t \delta_i, \tag{2.1}$$

where δ_i is the ramping rate parameter on Trial i. The left panel of Fig. 2.2 depicts the linear ramping function for three hypothetical values of δ, where larger values of δ cause the ROI activation to grow at a faster rate. In this illustrative example, the parameter δ might describe the latent neural activation of a key brain area that relates to memory encoding, such as the hippocampus or medial temporal lobe. The main purpose of the example is to see how the likelihood of recognizing a test item is related to the level of neural activation δ observed during study.

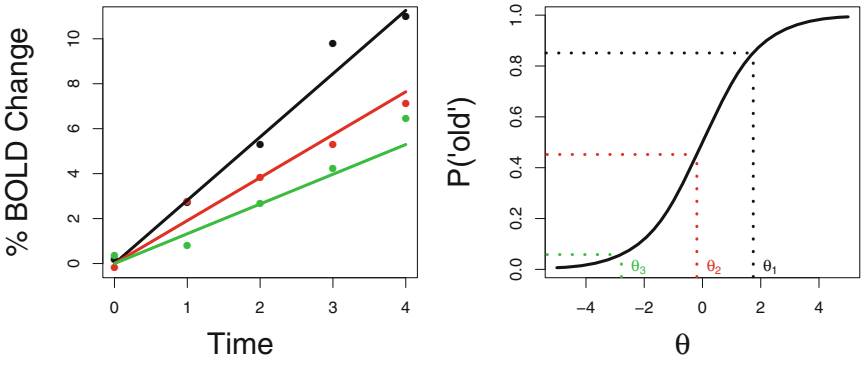


Fig. 2.2 Three realizations of data simulated from the model. Once the behavioral model parameters θ and the neural activation parameters δ have been simulated, they can be used to generate predictions for the observed behavioral (B) and neural (N) data, respectively. The left panel shows three ramping functions predicted by the model (lines) along with three random draws obtained by simulating the model (dots). The right panel shows the corresponding values for the behavioral model parameters (θ; x-axis) that are converted into probabilities of memory formation (y-axis) according to a logistic model. As a general rule, larger ramping functions produce larger probabilities of memory formation because the parameters of the two submodels (i.e., θ and δ) are positively correlated (i.e., $\rho = 0.6$)

While Equation 2.1 describes the mean predicted BOLD response over time, it is unlikely that in practice the observed data will follow such a simple pattern. To simulate hypothetical data, we assume that the observed BOLD responses $N_{i,t}$ are perturbed by some random observation error ϵ, such that

$$N_{i,t} = T_t \delta_i + \epsilon_i. \tag{2.2}$$

We can further assume that the observational errors ϵ_i are independent and identically distributed according to a normal distribution:

$$\epsilon_i \sim \mathcal{N}(0, \sigma),$$

where $\mathcal{N}(0, \sigma)$ denotes a normal distribution with mean zero and standard deviation σ. Equivalently, we simply can write

$$N_{i,t} \sim \mathcal{N}(T_t \delta_i, \sigma). \tag{2.3}$$

For nontrivial values of σ, we can expect the observed variables $N_{i,t}$ to depart from the model's predicted BOLD response of $T_t \delta_i$. For example, when neural data from the model, the left side of Fig. 2.2 shows random realizations of $N_{i,t}$ as dots along with the predicted BOLD response from the model as lines with coordinating colors. As we describe below, in this example we set $\sigma = 0.5$ which is small relative to the range of BOLD responses. As a consequence, the simulated data (dots) are tightly aligned with the predicted BOLD response for each of the three settings of δ.

2.1.2 Behavioral Submodel

Although a variety of theories have been put forth to describe the process of encoding and how it relates to the process of retrieving the memory at test [53–55], a simple, illustrative model might suppose that for the ith test item, the degree of "familiarity" with that item is represented by a parameter θ_i. The process of encoding items during the study period amplifies the familiarity with that item, and so θ_i for studied items will tend to be larger than that of unstudied items.

To convert the item familiarities θ_i to a probability of responding "old" (i.e., the probability of remembering that the item was on the previously studied list), we assume a logistic function that maps θ_i into p("old"), such that

$$\text{p("old")} \mid \text{Item } i) = \left(\text{logit}^{-1}(\theta_i)\right).$$

The logit function is a convenient function for mapping variables who have support ranging from $(-\infty, \infty)$ to variables with bounded probability support ranging from [0, 1]. The logit function is

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right),$$

and the inverse logit function is

$$\text{logit}^{-1}(x) = \left(\frac{1}{1+\exp(-x)}\right).$$

To simulate from the joint model as well as fit it to data, we need to define both of these functions. In R, we write the following to declare these functions for later use:

```
# need both logit and logit^{-1} functions
logit <- function(x)log(x/(1-x))
invlogit <- function(x){1/(1+exp(-x))}
```

To see the effects of the logit transformation on the probability of "old" response, consider three random values of θ as shown in the right panel of Fig. 2.1. In this particular simulation, θ_1 was a large value corresponding to a large ramping rate δ_1 in the left panel of Fig. 2.1. The value θ_2 was a intermediate value and θ_3 was the smallest draw. In general, the larger the rate of the ramping function from the neural submodel, the larger the familiarity parameter θ, because they were simulated from hyperparameters that forced a strong positive correlation between θ and δ. The logistic function is shown as the black line in the right panel of Fig. 2.1. The right panel illustrates how values of θ (x-axis) get mapped to probabilities of responding "old" (i.e., y-axis) by marking their paths with dashed lines.

To connect the probability of an "old" response to the observed behavioral variable B_i in Fig. 2.1, we assume that each B_i is a Bernoulli random deviate drawn with probability p("old" | Item i), such that

$$B_i \sim \text{Bernoulli}\left(\text{p}(\text{"old"} \mid \text{Item } i)\right). \tag{2.4}$$

Although our behavioral model is a simple one, it bears some resemblance to the high-threshold model proposed by [56], but does not have the additional guessing parameter in the decision rule. While the behavioral submodel proposed here is simple enough for illustration purposes, it is unlikely to fit data well, nor is it mechanistic in that it does not describe why θ varies from one item to the next.

2.1.3 *Linking the Submodels*

The final component of the model is the specification of how the submodel parameters (θ and δ) covary. There are many ways to specify this function, and indeed, this choice determines which type of joint model is being used (see Chap. 1).

For example, if you believed that there was a one-to-one correspondence between a brain region and a model parameter, you could use a directed approach that regressed neural activity onto the behavioral model parameter. If you believed a many-to-one correspondence, you might use a weighted average of the activations in the regions of interest to produce a single behavioral model parameter. However, if you were interested in the patterns of coactivation in the brain and how they relate to the behavioral model, you might want to specify a very generic linking function, such as the multivariate Gaussian linking function we have often used [12].

Figure 2.1 illustrates the dependence between θ and δ through a path diagram. Importantly, θ and δ are not directly connected in this particular joint model, which implies that they are conditionally independent given the hyperparameters. However, because all parameters in the diagram comprise a single model, the parameters θ and δ are not *marginally* independent. As we will see later in this chapter, the dependency between these parameters can be used to constrain the parameter estimates, and this constraint can lead to better predictions about unobserved data.

In our original presentation of joint modeling [12], we specified that θ and δ should have some linking function $\mathscr{M}$, dictated by a set of hyperparameters Ω, such that

$$(\theta, \delta) \sim \mathscr{M}(\Omega). \tag{2.5}$$

This specification is openly generic to allow for a variety of different linking functions to constrain the estimates of θ and δ [2, 3, 57]. However, to fit the model formally, we must make a specific assumption about the form of $\mathscr{M}$ and the role that the parameters Ω play. In several studies [12, 18, 46], we have assumed $\mathscr{M}$ to be a multivariate normal distribution with mean vector ϕ and variance covariance matrix Σ, where $\Omega = \{\phi, \Sigma\}$. Formally, we write

$$(\theta, \delta) \sim \mathscr{N}_p(\phi, \Sigma), \tag{2.6}$$

where $\mathscr{N}_p(a, b)$ denotes a multivariate normal distribution of dimension p (i.e., the length of θ plus the length of δ) with mean a and variance covariance matrix b. Box 2.1.3 describes some interesting properties of the multivariate normal distribution in detail. The multivariate normal assumption has proven useful for a number of reasons. First, the mean vector ϕ conveniently characterizes the central tendency of each parameter θ and δ. Second, the variance covariance matrix Σ expresses the degree of relatedness between each pairwise combination of θ and δ, which allows us to assess how well our model relates to brain activity across a set of ROIs. Third, as we will discuss below, with the appropriate prior selection for ϕ and Σ, we can establish a conjugate relationship between the prior and posterior, which facilitates efficient estimation of the model parameters [18].

The Multivariate Normal Distribution

To illustrate some of the conveniences of the multivariate normal distribution as well as how the "multivariate" part of the model departs from the standard normal distribution, consider the following partition:

$$\left[\frac{\theta}{\delta}\right] \sim \mathcal{N}_p\left(\phi = \left[\frac{\phi_1}{\phi_2}\right], \Sigma = \left[\begin{array}{c|c} \Sigma_{1,1} & \Sigma_{1,2} \\ \hline \Sigma_{2,1} & \Sigma_{2,2} \end{array}\right]\right)$$

Here, ϕ_1 is a $(p_1 \times 1)$ matrix, and ϕ_2 is a $(p_2 \times 1)$ matrix, where $p_1 + p_2 = p$. Partitioning the parameters in this way allows us to identify the mean and variance components of the submodel parameter sets. Given the properties of the multivariate normal distribution, there are three important facts worth highlighting [58, 59].

1. The marginal distributions of θ and δ are multivariate normal. Specifically, $\theta \sim \mathcal{N}_{p_1}(\phi_1, \Sigma_{1,1})$ and $\delta \sim \mathcal{N}_{p_2}(\phi_2, \Sigma_{2,2})$.
2. The conditional distributions of θ and δ are multivariate normal. Specifically, the conditional distribution of θ given that δ is some specific value, say $\delta = \delta^*$, is

$$\theta \mid \delta = \delta^* \sim \mathcal{N}_{p_1}\left(\phi_{1|2}, \Sigma_{1,1|2}\right),$$

   where

$$\begin{aligned} \phi_{1|2} &= \phi_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(\delta^* - \phi_2) \\ \Sigma_{1,1|2} &= \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1}. \end{aligned} \tag{2.7}$$

3. A zero covariance matrix $\Sigma_{1,2}$ implies that θ and δ are independent. Thus, θ and δ are independently distributed if and only if $\Sigma_{1,2} = \mathbf{0}$, where $\mathbf{0}$ indicates a $(p_1 \times p_2)$ matrix of zeros.

Together, Facts 1 and 2 provide some interesting (and perhaps surprising) information about exactly how neural measures constrain behavioral submodel parameters. Equation 2.7 shows that the mean of the conditional distribution of θ depends on the value of δ^*, whereas the variance of the conditional distribution of θ does not. However, both the mean and variance depend on the covariance matrix $\Sigma_{1,2}$. In general, we can think of the multivariate normal distribution as consisting of multiple normal distributions, with the added complexity of specifying how these normal distributions are correlated (i.e., elements within the matrix $\Sigma_{1,2}$).

The properties of the hyperparameters will depend on how the lower-level parameters θ and δ are used. For example, θ and δ could represent subject-specific parameters meaning that Ω would describe the distribution of the model parameters between subjects in the group. By contrast, θ and δ could also represent trial-specific parameters meaning that Ω would be a set of condition- or subject-specific parameters. Regardless of the characterization of the model parameters, the hyper mean vector ϕ can be divided into the set of mean parameters for the neural submodel (δ_μ) and the behavioral submodel (θ_μ), such that $\phi = \{\delta_\mu, \theta_\mu\}$. Similarly, the variance-covariance matrix Σ can be partitioned as

$$\Sigma = \left[\begin{array}{c|c} \delta_\sigma^2 & \rho\delta_\sigma\theta_\sigma \\ \hline (\rho\delta_\sigma\theta_\sigma)^T & \theta_\sigma^2 \end{array}\right] \tag{2.8}$$

to reflect that it consists of matrices that characterize various dispersions of the model parameters. Note that the variance-covariance matrix $\rho\delta_\sigma\theta_\sigma$ uses the parameter matrix ρ to model the correlation between submodel parameters. Specifying the model in this way allows us to directly infer the degree to which behavioral submodel parameters are related to which neural submodel parameters. To reduce the number of model parameters, we can also constrain elements of this partition to be equal to zero. For example, if we were uninterested in correlations that might exist from one parameter in the behavioral model to another, we could impose a constraint on θ_σ^2 to make the off-diagonal elements equal to zero. Or, if we had a specific brain-to-mechanism hypothesis we wanted to investigate, we could selectively estimate specific elements of ρ [46]. Such constraints are particularly useful when the intention of one's research is confirmatory rather than exploratory [57].

2.1.4 Simulating Data

With the model's functional form specified, we can now generate predictions from the model, given a set of specific model parameters. This is an important step as we often will want to evaluate how well the model fits to data, either before we collect experimental data or after parameter estimates have been obtained. It is also considered good modeling practice to first establish that one can accurately recover the model parameters by fitting the model to its own simulated data, a task we will perform later in this chapter.

In this section we provide R code that can be used to generate data from the joint model described above. For this model, both θ and δ are single parameters characterizing the behavioral and neural data, respectively. This implies that ϕ is a vector of length two, and Σ is a (2×2) diagonal matrix. To generate data at the level of the observed variables in Fig. 2.1, we begin by first picking values for ϕ and Σ in order to produce a matrix (θ, δ). In our example, we assume that this joint

model characterizes the data for a single subject, and so the rows of the matrix (θ, δ) correspond to individual trials. As such, we must also specify the number of trials $n = 500$. To initiate ϕ, Σ, and the matrix (θ, δ), we can run the following block of code:

```
# install a package for the MVN distribution
install.packages("mvtnorm")

# set up model specification
n <- 500      # total number of trials

# establish the hyperparameters
sig1 <- .5    # std. dev. of single-trial BOLD responses
sig2 <- 1     # std. dev. of item memory strength (logit scale)
rho <- .6     # cor b/n brain activation and memory strength

# set up hyper variance-covariance matrix Sigma
sigma <- matrix(c(sig1^2,         # element [1,1]
                  sig1*sig2*rho,  # element [1,2]
                  sig1*sig2*rho,  # element [2,1]
                  sig2^2          # element [2,2]
                   ),2,2,byrow=TRUE)

# set up hyper mean vector phi
phi <- c(2,0)

# simulate single-trial delta and theta matrix
DeltaTheta <- rmvnorm(n,phi,sigma)
```

The first step is to install the "mvtnorm" package in Lines 1–2 so that we can sample parameters from the multivariate normal distribution. Lines 7–10 correspond to the individual elements of Σ, as described in Equation 2.8. Lines 19–20 establish the hyper mean vector ϕ, and lines 22–23 generate the matrix of single-trial parameters (θ_i, δ_i). Hence, the variable "DeltaTheta" in the code is a matrix of size $(n \times 2)$.

With the δ and θ nodes from Fig. 2.1 generated, we can now simulate the observed neural and behavioral data nodes N and B, respectively. To do this, we run the following block of code:

```
# generate observed variable nodes
ts <- seq(0,4,1)    # scan times
sig <- .5           # the std. dev. of BOLD responses

# declare some storage objects
N <- matrix(NA,n,length(ts))
B <- numeric(n)

# loop over trials
for(i in 1:n){
  # N is a normal deviate with mean controlled by delta
  N[i,]=rnorm(length(ts),DeltaTheta[i,1]*ts,sig)
  # B is a Bernoulli deviate with probability controlled by
    theta
  B[i]=rbinom(1,1,invlogit(DeltaTheta[i,2]))
}
```

In the code, we first define a set of time points at which the neural measures are collected. In our running example, these time points correspond to the times when the ROI was scanned, and they correspond to the neural measures $N_{i,t}$ at each time t. We then define the variable "sig", which corresponds to the standard deviation of the BOLD signal over time. Next, lines 5–7 define storage objects "N" and "B", which correspond to the observed variables N and B in Fig. 2.1. Finally, we generate the observed data by randomly sampling N and B in lines 12 and 14, which correspond to Equations 2.2 and 2.4, respectively. Together, the blocks of code presented above were used to generate the data illustrated in Fig. 2.2.

2.2 Inferring the Model

While generating data from the joint model is an important part of model development and testing, the more practical modeling exercise is performing inference on the structure present in brain-behavior relationships. In model-based cognitive neuroscience, inferring brain-behavior relationships relies on assessing the distribution of parameter estimates after a model has been fit to data. As such, the quality of the parameter estimates is an important concern, and so reliable methodology should be invoked whenever possible.

In this section, we describe how to fit a joint model to data. Consistent with our running example, we will fit the same joint model presented above to data that was generated in the first section. Such a modeling exercise is considered good practice [60] because if we can show that parameter estimates can be properly recovered when they are known in simulation studies, it provides some assurance when the model is applied to real data (when parameters are unknown).

In parallel with the first section, this section details the equations necessary for fitting the model to data by considering each subcomponent of the model separately. Once each subcomponent has been described, we discuss how JAGS can be used to fit the joint model to simulated data. Readers who are not interested in the technical details of the model may safely skip to the prior specification in Sect. 2.2.3.1, which is essential for following the JAGS implementation. We close the inference section with a description of how to assess parameter recovery.

2.2.1 Equations for the Neural Submodel

For the neural submodel, Equation 2.2 describes the random distribution of the observed neural data N as a function of the model parameters δ. The distribution of neural data is normal in form as a consequence of the assumption about the observational errors ϵ. The notation in Equation 2.2 describes how the neural data N can be simulated from the model with parameters δ, but the equation that characterizes the probability of observing a particular $N_{i,t}$ given the parameter δ_i is written

$$p(N_{i,t}|\delta_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{\left[N_{i,t} - T_t\delta_i\right]^2}{2\sigma^2}\right). \tag{2.9}$$

To perform inference, we require a mathematical statement about how well each parameter δ relates to the data that were observed. In other words, we require the inverse of the conditional statement in Equation 2.9, which is known as the likelihood function $\mathscr{L}(\delta|N)$. To define the likelihood function, we take the product of the densities in Equation 2.9 evaluated at each data point $N_{i,t}$:

$$\begin{aligned}\mathscr{L}(\delta|N) &= \prod_t \prod_i p(N_{i,t}|\delta_i) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{n^*t^*} \prod_t \prod_i \exp\left(\frac{\left[N_{i,t} - T_t\delta_i\right]^2}{2\sigma^2}\right),\end{aligned} \tag{2.10}$$

where n^* and t^* denote the number of trials and time points, respectively. Evaluating this function for a set of δs and a matrix of neural data N will provide an assessment of how "likely" the single trial parameters δ are to have produced the data N.

2.2.2 *Equations for the Behavioral Submodel*

Similar to the neural submodel, we require a statement about how the model parameters θ relate to the observed behavioral data B in order to form the likelihood function. Recall that the behavioral model relates the model parameters θ to the probability of an "old" response through the Bernoulli function in Equation 2.4. While the notation in Equation 2.4 describes how the behavioral data might be simulated, the probability distribution $p(B_i|\theta_i)$ characterizing the behavioral data conditional on the model parameters θ_i is

$$p(B_i|\theta_i) = \left(\text{logit}^{-1}(\theta_i)\right)^{B_i} \left(1 - \text{logit}^{-1}(\theta_i)\right)^{1-B_i}.$$

As in the neural submodel above, we simply invert the conditional relationship between B and θ to form the likelihood function:

$$\begin{aligned}\mathscr{L}(\theta|B) &= \prod_i p(B_i|\theta_i) \\ &= \prod_i \left(\text{logit}^{-1}(\theta_i)\right)^{B_i} \left(1 - \text{logit}^{-1}(\theta_i)\right)^{1-B_i}.\end{aligned} \tag{2.11}$$

2.2.3 Equations for the Linking Structure

Following Equation 2.6, the linking function describes how θ and δ can be randomly simulated across the trials in an experiment. Consequently, the probability distribution $p(\theta_i, \delta_i)$ of a particular $z_i = (\theta_i, \delta_i)$ pair is written

$$p(\theta_i, \delta_i | \phi, \Sigma) = \frac{1}{\sqrt{2\pi |\Sigma|}} \exp\left(-\frac{1}{2} [\phi - z_i]^T \Sigma^{-1} [\phi - z_i] \right), \tag{2.12}$$

where $|\Sigma|$ is the determinant of Σ. Hence, Equation 2.12 details the joint distribution of θ and δ so that both θ and δ are informed by their respective streams of data B and N, while being simultaneously constrained by ϕ and Σ. In this way, Equation 2.12 can be thought of as a prior distribution on θ and δ.

2.2.3.1 Priors on Hyperparameters

Because we are fitting our joint model in a hierarchal Bayesian framework, the final step in specifying the model is in the prior specification for the hyperparameters ϕ and Σ. Specifying priors is an important step, as the choice of prior is considered a choice each researcher must make about the constraints on the model's parameters [61–65]. In the previous section we alluded to the importance of the prior specification in facilitating the estimation of the model parameters. One convenient choice is to specify a dependent conjugate prior on $\Omega = (\phi, \Sigma)$, such that

$$p(\Omega) = p(\phi, \Sigma) = p(\phi \mid \Sigma) p(\Sigma).$$

With this dependence structure, we can choose a multivariate normal prior for $p(\phi \mid \Sigma)$ and an inverse Wishart prior on $p(\Sigma)$ of the form

$$\phi \mid \Sigma \sim \mathcal{N}_p(\phi_0, s_0 \Sigma), \text{ and} \tag{2.13}$$

$$\Sigma \sim \mathcal{W}^{-1}(I_0, n_0), \tag{2.14}$$

where $\mathcal{W}^{-1}(a, b)$ denotes the inverse Wishart distribution with dispersion matrix a and degrees of freedom b. This particular specification creates conjugacy such that the conditional distribution of ϕ is multivariate normal and the conditional distribution of Σ is an inverse Wishart distribution.

2.2.4 Fitting The Model

With all subcomponents of the model in place, our goal is to estimate the model parameters. There are two levels of parameters that we care about. First, we would like to estimate the hyperparameters ϕ and Σ for this hypothetical subject. Second,

we would like to estimate each of the single-trial parameters θ and δ. In the Bayesian setting, parameter estimates come in the form of posterior distributions, and when examining multiple parameters at once, these estimates are called "joint posteriors." In fitting the model to data, our goal is to generate samples from the joint posterior distribution of the model parameters conditional on the observed data, which is written

$$p(\theta, \delta, \phi, \Sigma|N, B) \propto \mathscr{L}(\theta|B)\mathscr{L}(\delta|N)p(\theta, \delta|\phi, \Sigma)p(\phi|\Sigma)p(\Sigma),$$

where each function on the right side is given by the equations listed above. In some applications [12, 18, 46], we have used differential evolution with Markov chain Monte Carlo (DE-MCMC) [66, 67] to sample from the joint posterior. In these contexts, we applied full computational models such as the Linear Ballistic Accumulator [36] model or the Diffusion Decision Model [37], whose posteriors have been shown to be highly correlated. The DE-MCMC algorithm is particularly advantageous when the model parameters are highly correlated because the algorithm can automatically tune itself to the shape of the posterior during the sampling period.

While DE-MCMC is useful in the more applied setting, our goal here is to show that some simple joint models can be developed and fit to data by making use of existing Bayesian software packages such as JAGS. JAGS is especially convenient for R or MATLAB users because packages exist in both languages for conveniently interfacing with JAGS. We first describe the JAGS code for fitting the joint model, and then discuss the R code needed to execute the JAGS operations.

Installing JAGS and the rjags Package

To perform the model fitting we present in this chapter, two software packages must be installed. We recommend first installing the JAGS software by visiting the following website and downloading the version of JAGS that corresponds to your operating system:

http://mcmc-jags.sourceforge.net/

Once complete, you must follow the documentation from the JAGS website to install JAGS on your computer. Following the installation, open R and simply type the following commands directly into the R console:

```
> install.packages("rjags")
> require("rjags")
```

These statements will automatically download the rjags package and connect rjags with your version of JAGS installed in the previous step. For more specific instructions about syncing up your version of rjags with JAGS, please visit the following website:

https://cran.r-project.org/web/packages/rjags/index.html

2.2.4.1 JAGS Code

The most central component of the code is the JAGS file needed to perform sampling of the joint posterior distribution. Within this code, we must construct the model by specifying how both the neural and behavioral data are related to the model parameters. The JAGS code is split into two parts: one part defines the likelihood function, whereas the other part describes the priors on the model parameters. To conveniently interface with R, we save the following JAGS code in a separate file, and will later tell R what the name of the file is that specifies the model:

```
1  # JAGS code, file named ``model_general.txt"
2  model {
3      # convert sig to tau for convenience
4      tau <- pow(sig, -2)
5
6      # loop through trials to define likelihood
7      for (i in 1:n){
8          for (t in 1:Nt){
9              # likelihood for neural data
10             N[i,t] ~ dnorm(DeltaTheta[i,1]*ts[t],tau);
11         }
12         # likelihood for behavioral data
13         B[i] ~ dbin(1/(1+exp(-DeltaTheta[i,2])),1);
14     }
15
16     # loop through trials to define prior on (delta, theta)
17     for(i in 1:n){
18         DeltaTheta[i,1:2] ~ dmnorm(phi,Omega);
19     }
20
21     # priors on hyperparameters
22     phi ~ dmnorm(phi0,s0);
23     Omega ~ dwish(I0,n0);
24     # convert Omega to Sigma for convenience
25     Sigma <- inverse(Omega);
26 }
```

In this code, Lines 6–14 correspond to the definition of the likelihood function, Lines 16–19 correspond to the linking function (i.e., the "prior" on θ and δ), and Lines 21–25 correspond to the prior on the model hyperparameters. In particular, Line 10 corresponds to Equation 2.10, Line 13 corresponds to Equation 2.11, Line 18 corresponds to Equation 2.12, and the priors specified in Equations 2.13 and 2.14 are imposed in Lines 22 and 23, respectively. Lines 3–4 convert the variable "sig" into a precision parameter "tau", as JAGS parameterizes the normal distribution in terms of the mean and precision rather than the mean and standard deviation as in R. Similarly, Lines 24–25 convert the variable "Omega" into the variance-covariance matrix "Sigma". These conversions allow us to match up the scale of the parameters used to generate the data in R with the parameters that are estimated by JAGS.

2.2.4.2 R Handler Code

To run the JAGS code within R, we must first have installed the JAGS software and the rjags R package that interfaces with JAGS. Instructions for installing JAGS and the rjags interface are provided in Box 2.2.4. Once JAGS is installed, we can load the rjags package in our R workspace so that we can make use of the functions that control JAGS. Although there are many functionalities of the rjags package, for our illustration, only a few operations are necessary: establishing the model, adapting the sampler, updating the chains, and collecting samples. The following block of code will perform all of these operations in R, assuming that some variables have already been initialized from the implementation of the joint model in the first section:

```
1  # specify the jags model:
2  # locate the JAGS code, pass variables, setup sampler
3  jags <- jags.model('model_general.txt',
4                  data = list('n'=n,
5                              'B'=B,
6                              'N'=N,
7                              'ts'=ts,
8                              'Nt'=length(ts),
9                              'sig'=sig,
10                             'I0'=diag(2),
11                             'n0'=2,
12                             'phi0'=rep(0,2),
13                             'epsilon0'=diag(2)),
14                 n.chains = 4,
15                 n.adapt = 1000)
16
17 # continue adapting the sampler to optimize sampling efficiency
18 adapt(jags, 1000, end.adaptation=TRUE);
19
20 # continue sampling to ensure convergence
21 update(jags, 1000)
22
23 # draw final samples, and monitor important variables
24 out=jags.samples(jags,
25              c('phi', 'Sigma', 'DeltaTheta'),
26              1000)
```

In this code, Lines 3–15 set up the model within R by establishing a set of variables that will be passed to the JAGS code. The first line tells R what the filename of the JAGS code is, the "data" statement in Line 4 are the set of variables that get passed to JAGS, the "n.chains" statement tells JAGS how many chains to use during the sampling period, and "n.adapt" tells JAGS how many iterations of its "adaption" feature to run in initializing the sampler.

Following the initialization of the sampler, we can choose to continue adapting the sampler to further improve the sampling efficiency once posterior samples are acquired. As an example, Lines 17–18 continue the adaptation process and will return either a "TRUE" or "FALSE" printout depending on whether the adaption is

complete or not, respectively. Once the sampler is adapted, we can draw samples from the posterior by using the "update" function. For example, Lines 20–21 perform an update for 1000 iterations for each of the chains. The final step is to generate and collect samples from the posterior, which is performed via the function "jags.samples." Within this function, we can tell JAGS which of the variables are of interest in the estimation. For example, in Line 25 we have specified that the variables "phi", "Sigma" and "DeltaTheta" variables are to be stored as output, but because we did not specify that variables such as "sig" or "Omega" are important in the JAGS code, they will not be returned once the sampling has terminated.

2.2.5 Assessing Parameter Recovery

Because this is a simulation study, we can examine the accuracy of the estimated model parameters by comparing the estimates provided by JAGS to the values used to generate the data in R from the previous section. We can assess accuracy of the model parameters at the hyper- and trial-levels. For the single-trial level, because there are so many parameters, a convenient way to examine the accuracy of these estimates is to first collapse across the chains and iterations to form a single estimate of the central tendency of the posterior distributions, called the "maximum a posteriori" (MAP) estimate. MAP estimates are intended to be the value in the parameter space that is most likely to have generated the data. In practice, the mean statistic of the posterior samples closely resembles the MAP estimate, and is much more convenient to calculate. Hence, while we provide the mean statistic, we refer to it as the MAP out of convenience.

Within R, we can form a MAP estimate by taking the average across both of these dimensions via the following code:

```
# calculate the mean of the posteriors
maps=apply(out$DeltaTheta,c(1,2),mean)
# delta is the first column, theta is the second column
delta=maps[,1]
theta=maps[,2]
```

Running this code will create the variable "delta" and "theta" that correspond to the nodes δ and θ in Fig. 2.1. We can then compare these estimates with the matrix "DeltaTheta" created in the first section when simulating data from the model.

Figure 2.3 shows the estimated model parameters on the y-axis against the true values of the model parameters on the x-axis for δ (right panel) and θ left panel. Within each panel, the correlation between these values is reported. For δ, the correlation between true and estimated model parameters is 0.99, suggesting excellent recovery of the model parameters. By contrast, the left panel of Fig. 2.3 shows that the correlation between the true and estimated model parameters is 0.65, suggesting good, albeit worse, recovery of the θ parameters. One explanation for the

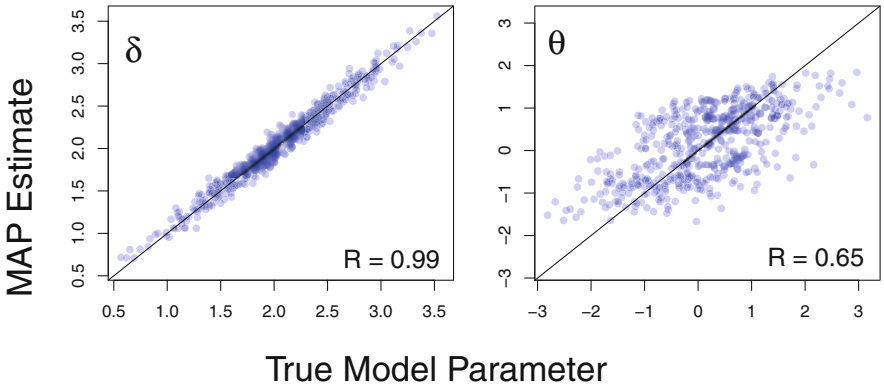


Fig. 2.3 Recovery of the single-trial model parameters. The maximum a posteriori (MAP) estimate (y-axis) is plotted against the true model parameter (x-axis) for the neural model parameters δ (left panel) and the behavioral model parameters θ (right panel). Within each plot, the correlations between parameter estimates and true values of the parameters are reported

differences in the recovery of the model parameters is due to the mismatch in the number of observations for δ compared to θ. Recall that a total of five time points are being explained by each δ_i, whereas only a single data point B_i is being explained by each θ_i. In general, having more data per model parameter will improve the accuracy of the estimated posterior, and as a consequence, the correlation between the MAP estimates and the true model parameters.

We can also assess parameter recovery at the hyper level for the parameters ϕ and Σ. Because there are fewer hyperparameters, we can easily visualize the recovery of the posterior distributions relative to the true model parameters. Figure 2.4 shows the estimated posterior distributions for ϕ (left panel) and Σ (right panel). In each panel, the estimated posterior distribution is illustrated with a violin plot, and the true parameter value is shown as the black "X" symbol. For ϕ, Fig. 2.4 shows that both hyper mean parameters were accurately recovered, where ϕ_1 for the neural data has less variance relative to the parameter estimate ϕ_2. Once again, we attribute this difference to the constraint on δ due to the increased number of data points on the neural side of the model relative to the behavioral side.

The right panel of Fig. 2.4 shows the accuracy of the estimated parameters within the matrix Σ. Here, we have separated out the components of Σ according to the standard deviation of the neural model parameters σ_1 (left), the standard deviation of the behavioral model parameters σ_2 (middle), and the correlation between the single-trial parameters ρ (right). Similar to ϕ, the parameters correspond to the neural subcomponent of the model are better informed and have less variance relative to the behavioral subcomponent of the model. In general, each of the three estimated model parameters within Σ are aligned with the true value.

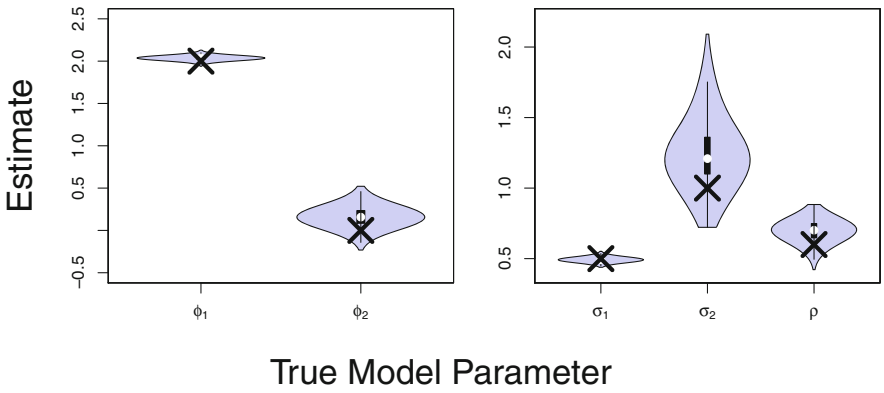


Fig. 2.4 Recovery of the hyperparameters in the joint model. In each panel, the estimated posterior distributions are illustrated with a violin plot and the true value of the model parameters are shown as the black "X." The estimates corresponding to the parameters ϕ are shown in the left panel, whereas the parameters corresponding to the elements within Σ are shown on the right panel

2.3 Other Considerations

The tutorial presented above was intended to illustrate how to generate data from and fit a very simple joint model. While simplicity is certainly useful for gaining a basic understanding, the models presented above are not very practical in that they can only be applied to very simple problems. In this section, we sketch out how to extend the basic models presented in the tutorial for more realistic problems. First, we discuss how priors can be specified for individual elements of Σ to allow for more flexibility in traversing the space from exploratory and confirmatory analyses. Second, we discuss how the JAGS code can be extended to allow for the inclusion of multiple neural covariates. Finally, we show how to use the joint model structure to created a directed model where the behavioral model parameters are determined by the neural activation parameters. While the direction of influence can easily be flipped, this example concretely shows how one type of directed joint model can be simulated and fit to data.

2.3.1 Priors for Specific Elements of Σ

When we have a good deal of information about which parameter in a cognitive model should be related to a neural measure, it may be worthwhile to specify a prior on the individual elements of the variance-covariance matrix Σ rather than on Σ itself. In practical applications, one situation where such prior specifications are advantageous is when we wish to constrain specific elements of Σ to be particular values, such as $\rho = 0$ or $\rho = 1$. When specific elements of Σ are fixed, the inverse Wishart prior on Σ is no longer an appropriate choice as conjugacy cannot be established.

To allow for element-wise constraints on Σ, we need only restructure our specification of the priors in JAGS. As an example, the R handler code presented in the tutorial above can be used in conjunction with the following JAGS code below:

```
# JAGS code, file named ``model_flex.txt"
model {
    # convert sig to tau for convenience
    tau <- pow(sig, -2)

    # loop through trials to define likelihood
    for (i in 1:n){
        for (t in 1:Nt){
            # likelihood for neural data
            N[i,t] ~ dnorm(DeltaTheta[i,1]*ts[t],tau);
        }
        # likelihood for behavioral data
        B[i] ~ dbin(1/(1+exp(-DeltaTheta[i,2])),1);
    }

    # loop through trials to define prior on (delta, theta)
    for(i in 1:n){
        DeltaTheta[i,1:2] ~ dmnorm(phi,Omega);
    }

    # priors on hyperparameters
    phi ~ dmnorm(phi0,s0);

    # calculate priors for individual elements of Sigma
    sig1 ~ dgamma(.01,.01);
    sig2 ~ dgamma(.01,.01);
    rho ~ dunif(-1,1);

    # compute Sigma from individual elements above
    Sigma[1,1] <- sig1^2;
    Sigma[2,1] <- sig1*sig2*rho;
    Sigma[1,2] <- Sigma[2,1];
    Sigma[2,2] <- sig2^2;

    # convert Sigma to Omega
    Omega <- inverse(Sigma);
}
```

In the code, only Lines 29–36 are necessary to "hard code" in the specification of how the individual elements are created within Σ, and Lines 24–27 are where the priors for the individual elements σ_1, σ_2, and ρ are imposed. Fitting this model produces nearly identical parameter estimates as those shown in Figs. 2.3 and 2.4, because there is enough data to negate the differences in the priors across the two models. Of course, when the dimensionality of ϕ and Σ grow, the construction of Σ within JAGS becomes more cumbersome, but nevertheless, it can always be programmed as in Lines 29–36 above.

2.3.2 *Adding Multiple Neural Covariates*

Although the example we presented above was for a single neural covariate, in most practical applications, the number of neural covariates will be much larger than one. For example, [18] modeled a total of 34 neural covariates and an extension presented in [68] modeled as many as 128 neural covariates. Luckily, extending the JAGS code presented above requires only a few modifications, if we assume that all covariates have a similar statistical model and that they are conditionally independent from one another. To do this, we first define a variable "Nroi", which is the number of neural covariates in the model. For example, these covariates could correspond to the number of electrodes in an EEG study, or the number of independent components in an fMRI study. Using a similar R handler code that was presented above, we can use the following JAGS code to fit a joint model with multiple covariates:

```
1  # JAGS code, file named ``model_general_multi.txt"
2  model {
3      # convert sig to tau for convenience
4      tau <- pow(sig, -2)
5
6      # loop through trials to define likelihood
7      for (i in 1:n){
8        for (t in 1:Nt){
9            for(k in 1:Nroi){
10               # likelihood for neural data
11               N[i,t,k] ~ dnorm(DeltaTheta[i,k]*ts[t],tau);
12               }
13         }
14         # likelihood for behavioral data
15         B[i] ~ dbin(1/(1+exp(-DeltaTheta[i,(Nroi+1)])),1);
16     }
17
18     # loop through trials to define prior on (delta, theta)
19     for(i in 1:n){
20         DeltaTheta[i,1:(Nroi+1)] ~ dmnorm(phi,Omega);
21     }
22
23     # priors on hyperparameters
24     phi ~ dmnorm(phi0,s0);
25     Omega ~ dwish(I0, n0);
26     # convert Omega to Sigma for convenience
27     Sigma <- inverse(Omega);
28 }
```

The code above shows an important change on Lines 9–12. Here, an additional "for" loop is necessary to cycle through the set of neural covariates, assuming that the neural data are arranged in an array where the three dimensions correspond to (1) the number of trials, (2) the number of time points, and (3) the number of neural covariates. Furthermore, the matrix "DeltaTheta" now contains all of the δ parameters for each trial and for each ROI.

Other areas of the code must be modified to interface with the changes that have occurred in the dimensionality of the parameter δ. Namely, Lines 15 and 20 both modify the column that is selected within the matrix "DeltaTheta" to go from (1:(Nroi + 1)) instead of from (1:2) as in the previous code. A final change that is necessary to fit the model is in the prior specification in the R handler code. As a result of the increased number of covariates and the increase in the dimensionality of "DeltaTheta", the following variables should be carefully chosen in this multivariate context: "phi0", "s0", "I0", and "n0". If the dimensionality of these variables is not properly aligned with the dimensionality of "DeltaTheta", JAGS will not proceed with the model fitting.

2.3.3 *Directional Influence*

As we discussed in the previous chapters, there are many ways to express the covariation of neural and behavioral data, and these ways comprise the set of models we consider to be "joint" models. Although most of our applications have focused on expressing the relationship between submodel parameters via a multivariate normal, other more restrictive expressions naturally follow from the generic linking function in Equation 2.5 [12].

As an example, maintaining that our behavioral and neural data can still be described via the submodels presented in Equations 2.11 and 2.10, suppose we wish to fit a joint model like the one presented in Fig. 2.5. Here, the hyperparameters ϕ and Σ no longer detail the statistical structure between θ and δ, but instead describe the trial-to-trial fluctuations observed only in δ. For example, we might assume

$$\delta_{i,k} \sim \mathcal{N}_p(\phi, \Sigma),$$

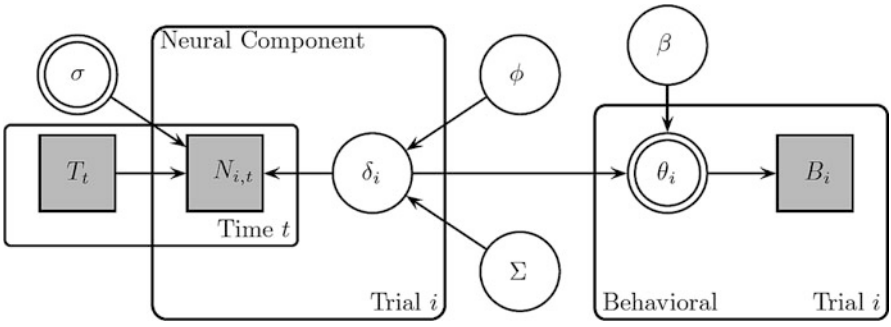


Fig. 2.5 Graphical diagram for a directed joint model. Each node represents a variable in the model, where gray nodes correspond to observed variables, white nodes correspond to latent variables, and double-bordered nodes correspond to deterministic nodes (that are not estimated). Paths indicate direct relationships between the variables and plates indicate replications across dimensions (e.g., trials or time). Note that the plate corresponding to multiple neural measures is not shown

where $\delta_{i,k}$ denotes the neural activation parameter on the ith trial for the kth ROI as described in Sect. 2.3.2. Here, ϕ and Σ describe how the parameters on the ith trial relate to say the jth trial across all ROIs, an model that is more realistic for problems we often face in neuroscience. Note that in Fig. 2.5, the plate for the ROIs is omitted for the purpose of visual simplification.

With an expression for the neural covariates in hand, we can specify how they might be used to constrain the latent parameters θ for the behavioral data B. For example, a simple linear model is

$$\theta_i = \sum_k \delta_{i,k}\beta_k, \tag{2.15}$$

where β_k are regression parameters relating each of the ROIs to the behavioral parameter θ_i. Here, θ_i is completely determined by the regression parameters β and the set of neural covariates δ, so the node corresponding to θ in Fig. 2.5 has a double border to express that it is no longer a freely estimated parameter.

The model illustrated in Fig. 2.5 represents an interesting hybrid model as it shares some similarities to the Integrative models discussed in Chap. 2, but it is also subsumed by the joint model structure we have discussed here and in Chap. 3. The model appears Integrative in that it has a set of model parameters that are no longer conditionally independent from either aspect of the data. However, it also has parameters that serve to mediate the mapping from neural parameters to behavioral parameters, something that is exploited in the more flexible joint models we have used in the past.

2.3.3.1 Model Code

With the model specified, we can program the directed joint model and fit it in JAGS. First, we can use R to generate data from the model, using the following code (in conjunction with the previous code presented above):

```
Nroi <- 2     # total number of ROIs

# declare some storage objects
N=array(NA,c(n,length(ts),Nroi))
B=numeric(n)
theta=numeric(n)

# set up regression parameters
beta <- c(.5,.3) # two ROIs

# loop over trials
for(i in 1:n){
  for(k in 1:Nroi){
  # N is a normal deviate with mean controlled by delta
  N[i,,k]=rnorm(length(ts),Delta[i,k]*ts,sig)
  }
```

```
17   # theta[i] is the single-trial behavioral parameter
18   theta[i] <- Delta[i,]%*%beta
19   # B is a Bernoulli deviate with probability controlled by
       theta
20   B[i]=rbinom(1,1,invlogit(theta[i]))
21 }
```

This code will produce neural data for a set of ROIs, dictated by "Nroi". The regression parameters β are specified in Lines 8–9, and are used to map the neural submodel parameters to the behavioral submodel parameters in Line 18. This code makes use of matrix multiplication to generate the neural data array N, as was discussed in Sect. 2.3.2.

Once the data are generated, we can use nearly identical R handler code presented above to call the following JAGS code, therein fitting the directed joint model to the generated data:

```
# JAGS code, file named ``model_directional.txt"
model {
    # convert sig to tau for convenience
    tau <- pow(sig, -2)

    # loop through trials to define likelihood
    for (i in 1:n){
        for (t in 1:Nt){
         for(k in 1:Nroi){
            # likelihood for neural data
            N[i,t,k] ~ dnorm(Delta[i,k]*ts[t],tau);
            }
      }
        theta[i] <- Delta[i,]%*%beta;
        # likelihood for behavioral data
        B[i] ~ dbin(1/(1+exp(-theta[i])),1);
    }

    # loop through trials to define prior on delta
    for(i in 1:n){
        Delta[i,1:Nroi] ~ dmnorm(phi,Omega);
    }

    # priors on hyperparameters
    phi ~ dmnorm(phi0,s0);
    Omega ~ dwish(I0, n0);
    # convert Omega to Sigma for convenience
    Sigma <- inverse(Omega);
    # prior on regression parameters
    for(k in 1:Nroi){
        beta[k] ~ dnorm(0,.001)
    }
}
```

Here, the JAGS code looks similar to the code for fitting the full joint model presented above, with some important departures. First, Line 14 establishes the variable "theta" for each trial i by first calculating Equation 2.15. Once "theta"

has been calculated, it is passed into the likelihood function on Line 16 in the same way as presented above. Second, Lines 19–22 describe how the single-trial neural submodel parameters δ are modeled according to ϕ and Σ (i.e., "Omega" in the code). Here, a new matrix called "Delta" contains all of the single-trial parameters for δ, and this matrix no longer includes the behavioral submodel parameters θ. This new matrix "Delta" must also be used when calculating the likelihood for the neural data in Line 11. Finally, while the lines specifying the priors for ϕ and Σ can remain unchanged, we must also specify a prior for the regression parameters β. Lines 29–32 specify the priors for any number of regression parameters. Here, we have specified a common prior with mean 0 and precision 0.001, but one could replace these fixed quantities with variables that would be passed to JAGS from R.

2.3.3.2 Recovering the Regression Parameters

After generating hypothetical data using the R code above, we can use the JAGS code to estimate the model parameters. Perhaps most interesting are the estimates for the regression parameters β. Figure 2.6 shows some estimated posterior distributions for β assuming two neural covariates (panels). The left panel shows the estimate for β_1, whereas the right panel shows the estimate for β_2. In both panels, the true value used to generate the data is shown as the red vertical line, corresponding to Line 9 in the R code for the generative model above. Although the priors for the β parameters are shown in Fig. 2.6, because they are so diffuse relative to the posteriors, they are barely visible in the figure (dotted lines along the bottom). Furthermore, the posterior estimates encompass the true value used

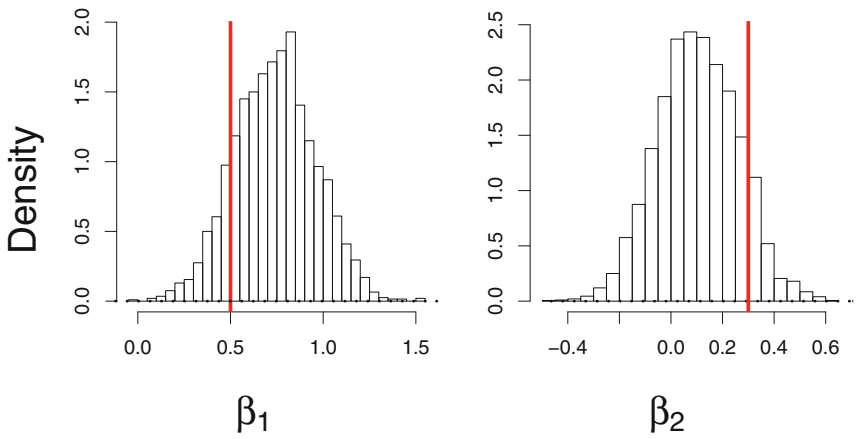


Fig. 2.6 Estimated regression parameters. The left and right panels show histograms of the estimated posterior distributions for β_1 and β_2, corresponding to the first and second ROI respectively. The true value of the parameter used to generate the data is shown as the vertical red line

to generate the data. These two aspects of the posterior estimates suggest that the regression parameters have been accurately recovered.

2.4 Conclusions

In this chapter, we have shown how to perform two important analyses of joint models. First, we showed how to simulate data from a joint model, given some choices about parameter values. Second, we showed how to infer parameter estimates given a set of neural and behavioral data. The models we investigated here were overly simply and likely unhelpful for particular applied settings. However, these models were designed to be simple so as to make the modeling examinations succinct. JAGS is a flexible program that can even be used to build in other density functions that are not immediately available in the stand alone version, such as the diffusion decision model [69]. Such an application was exploited in a larger scale tutorial article on joint modeling by Palestro et al. [70]. We will also discuss an application of the diffusion decision model by Turner et al. [18] with many more covariates than the example presented here.

Chapter 3
Assessing Model Performance with Generalization Tests

3.1 Assessing Model Fit

When evaluating a model's fit to data, perhaps one of the most important considerations is examining how close the model's predictions are relative to the data. However, this is not as straightforward as it might sound. When developing joint models, our goal is to put forth a model that can not only fit data well, but also makes a strong yet accurate commitment to the distribution of data we should see from our experiment [71–73]. The assessment of a model's full credentials involves two important considerations: model fit and model complexity [72–76]. Model fit is perhaps more obvious; it is defined as the likelihood of observing the data from our experiments under the particular constraints assumed within the model. Model complexity, on the other hand, is an assessment of how well we might expect a model to fit data, based on its architecture. Model complexity is defined as the range of hypothetical patterns of data a model can fit: the more complex a model, the better able it is to fit empirical data.

In the domain of model development, the word complexity can sometimes refer to either the flexibility of a model, and can sometimes refer to the ease of implementation [57]. However, these are two different concepts. The ease of implementation is related to concepts such as mathematical tractability and computation time, but it is not related to complexity [77, 78]. A model's complexity is related to the number of parameters, but simply counting up the number of parameters may not be the best approach. For example, simply adding an additional parameter does not necessarily tell us about the new range of hypothetical patterns of data the model can produce. In the context of Bayesian inference, adding one new parameter that has a highly restricted range (e.g., between 0.45 and 0.55) is likely to have very different consequences than adding the same parameter with an

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_3

unrestricted range (e.g., between zero and one). Given these concerns, we must be careful to consider not only the number of parameters, but the constraints on those parameters [62–64], and their uncertainty after fitting the model to data.

One strong candidate for assessing model fit that takes into account parameter uncertainty is the Bayes factor. Suppose we were interested in comparing two model candidates M_q and M_r. We can calculate the Bayes factor $BF_{q,r}$ between them by examining the ratio of each model's marginal likelihood function of the data D given the model M_i. The marginal likelihood integrates out (i.e., it "marginalizes") the contribution of the model parameters when fitting the data, by considering the range of predictions under a fixed data set. Mathematically, the marginal likelihood of the data D given the model M_i is expressed as

$$p(D|M_i) = \int_{\Theta} p(\theta) \prod_t p(D_t|\theta) d\theta, \tag{3.1}$$

where $p(\theta)$ is the prior distribution on the model parameters (i.e., the set of restrictions on each parameter θ), and $p(D|\theta)$ denotes the probability density function, which, after multiplying over the entire set of data becomes the likelihood of observing the set of data D, given the model parameters θ. With the marginal likelihood in place, we need only compare the ratio of marginal likelihoods to obtain the Bayes factor:

$$BF_{q,r} = \frac{p(D|M_q)}{p(D|M_r)}. \tag{3.2}$$

While in theory it should be simple to calculate the Bayes factor, it is sometimes difficult to estimate the marginal likelihood for a given model [79, 80]. This is especially the case when considering cognitive models, as they often make a number of assumptions that are based on complex, stochastic processes.

There are now many new methods for approximating the Bayes factor, which has broadened the use of Bayes factors in the cognitive modeling domain [81]. Most of these methods work by using Monte Carlo operations to estimate the marginal likelihood from the posterior samples that are obtained during the model-fitting process (see Chap. 2). As you might imagine, this is an extremely convenient way of performing the complicated integration shown in Equation 3.1.

3.2 Assessing Generalization

In the joint modeling framework, by jointly considering the available behavioral and neuroimaging data, all model parameters can be inferred simultaneously. There is no need to separate the parameter estimation process into stages. For a hierarchical joint model or a directed joint model, the consequence of simultaneous parameter inference is that the behavioral data can influence the neural model parameters

and the neuroimaging data can influence the behavioral model parameters. This is a desirable property if the joint model captures some aspect of true underlying generative process. In this case, the addition of neural data will be informative about the behavioral model parameters and therefore lead to improved predictions of the behavioral data. Similarly, the addition of behavioral data will be informative about the neural model parameters and therefore lead to improved prediction of the neural data. If, on the other hand, the joint model is severely misspecified, the addition of data from one modality will either have no consequence on the prediction of data in the other modality (when the model learns that no connection exists) or lead to worse prediction (when the wrong connection is learned). Therefore, an important test for joint models is the ability to generalize to new data.

In order to evaluate a joint model or compare among different joint models, there are a variety of generalization tasks that can be designed, each focusing on different modeling goals. Figure 3.1 illustrates one basic generalization tasks in a situation where the data consists of simultaneous neuroimaging and behavioral data from a number of individuals (five in the figure). The first important generalization task tests *whether the addition of neural data can improve behavioral predictions*. For example, imagine a joint model that has been estimated on the basis of the neural and behavioral data for all individuals except for one (individual 5). For the remaining individuals, the test is to make inferences about withheld behavioral data on the basis of the partially observable data for that individual. In case A, the only observable data is a subset of the behavioral data. In case B, the observable data consists of the same subset of behavioral data but also the neural data for that individual. If prediction performance is better in case B relative to A, this would demonstrate that the addition of neural data leads to better behavioral predictions. More generally, it demonstrates that the joint model has learned a useful relationship between the neural and behavioral data.

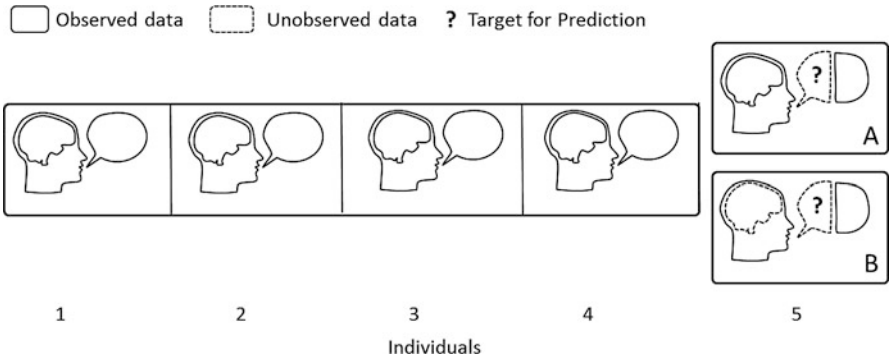


Fig. 3.1 Illustration of generalization task to test whether the addition of neural data can improve behavioral predictions. Neuroimaging data (illustrated by the brain) as well as behavioral data (illustrated by the speech bubble) is collected from a number of individuals (columns). Data is withheld from the model (dashed lines) in order to test the ability of the model to generalize to new behavioral data on the basis of partially observed behavioral data (case A) and additional neural data (case B)

One way to expand this generalization task is to vary the amount of partially observable behavioral data. For example, imagine that the behavioral experiment can continue for an extended period of time and that model parameters for a given individual can be improved by collecting larger quantities of behavioral data. One possibility is that the collection of more behavioral data can compensate for an absence of neural data. In this case, the neural data is informative about the missing behavioral data but the degree to which the neural data is informative diminishes as more behavioral data are observed. In contrast, another possibility is that neural data can improve prediction performance regardless of how much behavioral data is collected from the individual. In this case, the joint model is truly useful as the neural data can be used to provide an independent contribution in the prediction of behavioral data.

The previous generalization task can easily be reversed for joint models in order to test *whether the addition of behavioral data will improve the prediction of neural data*. Again, imagine a case where the joint data from all individuals is observed except for one. We first assess the prediction performance of a baseline model that has to generalize to a subset of neural data when only some subset of neural data has been observed. We then assess the prediction performance of another joint model that also has the partially observed neural data but also the corresponding behavioral data. If the addition of the behavioral data improve the performance of the second model, it would demonstrate that the behavioral data can constrain and inform the neural parameters of a joint model and improve neural predictions.

In more complex data sets, individuals might be involved in multiple behavioral tasks using multiple neuroimaging methods. In this case, joint models not only have to learn the relationship between neural and behavioral data within a given cognitive task but also across cognitive tasks. Figure 3.2 illustrates a number of generalization problems that can assess the ability of a joint model to learn within as well as across tasks. Problems A and B in Fig. 3.2 illustrate two problems analagous to Fig. 3.1. The goal is to predict the behavioral data that is withheld from the model. In problem A, the only observed data is a subset of the behavioral data whereas in problem B, additional neural data is available to inform model parameters.[1] As before, any advantage in prediction performance in case B can be taken as an indication that simultaneously recorded neural data can be informative about behavioral data. However, can the model also benefit by observing data in *different* cognitive tasks? For example, in problem C, all behavioral and neural data for the other cognitive tasks is observed. If generalization performance in problem C is better than in B, this demonstrates that the model has learned important relationships across cognitive tasks that can inform the individual model parameters for a given task. By withholding parts of the data in the other cognitive tasks (e.g., problem D), the exact nature of this cross-task benefit can be pinpointed better. Are the across task benefits solely based on behavioral model parameters or do they also include neural model parameters?

[1]For visual clarity, the two generalization tasks are set up between individuals but in an actual implementation, it is better to contrast the two tasks within an individual.

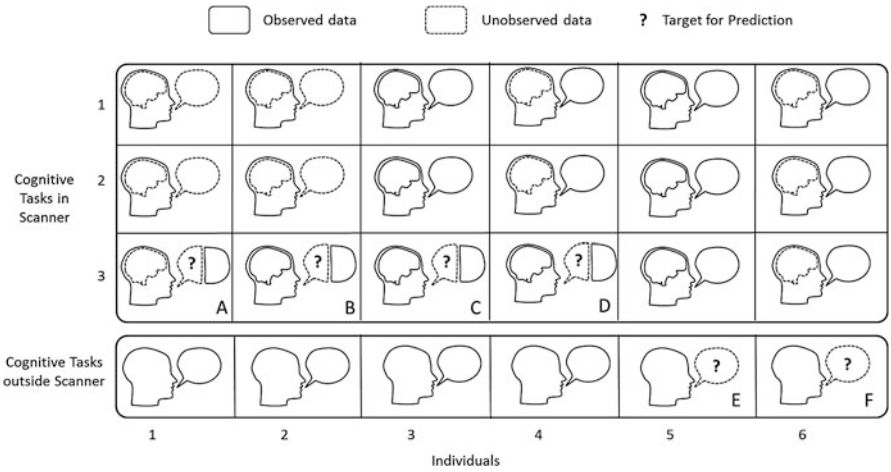


Fig. 3.2 Illustration of more complex generalization tasks involving a number of cognitive tasks (top three rows) performed within concurrent neural measurements, and a behavioral task without any concurrent neural measurements. Data is withheld from the model (dashed lines) in cases A–F in order to test the ability of the model to capture the relationship between neuroimaging and behavioral data

Finally, arguably the most challenging type of generalization problem is to predict cognitive performance on a task that is performed outside the context of a fMRI scanner or EEG recording device (such as an IQ assessment test or a problem-solving task in a real-world situation where simultaneous imaging is not practical). As an example of this type of generalization problem, problem E assesses the ability of the model to predict the behavioral data in a cognitive task when some combination of joint behavioral and neuroimaging data has been observed in other cognitive tasks. Problem F assesses the same generalization tasks when no neural measurements were observed (for that individual) in the other cognitive tasks. Any performance improvement in E over F demonstrates that the joint model can estimate neural model parameters that can generalize across tasks and to situations outside the context of the fMRI scanner or EEG recording device.

Overall, there is not a single generalization task that can be used to demonstrate that one joint model is better than another joint model (or non-joint model). Each model might be designed for a particular purpose. For cognitive modelers, the focal point might be on the behavioral side, in which case it becomes more natural to test the ability of a joint model to generalize to behavioral data. For computational neuroscience modelers, the emphasis might shift toward neural predictions. Regardless of the particular generalization task used, the key demonstration for any joint model is that the combination of behavioral and neural data will improve parameter estimation and subsequent model generalization.

3.3 Performing Generalization Tests

In Chap. 2, we discussed how to simulate data from a joint model, and how to fit a joint model to data. While fitting joint models to data can reveal interesting brain-behavior relationships that could be used to support, refute, or evolve a cognitive theory, as we have stressed so far, an equally compelling test of joint models is in their ability to generalize to new data [74–76, 82]. One standard approach is cross validation. In this method, a model is first fit to some percentage of the full data set, called the "training" set, and is then used to generate predictions about the pattern of data we might expect in a new context, be it on a new trial or for a new subject. The predictions from the model can then be compared to the remain percentage of the full data set, called the "test" set, and the fidelity of the model can be assessed by examining how well the predicted data match the data that were withheld. In this section, we illustrate how JAGS can be used to perform cross-validation tests for joint models, and then compare the performance a joint model's predictions to predictions by a model that considers only behavioral data. All of the associated code assumes that you are already familiar with the examples and code provided in Chap. 2.

3.3.1 Withholding (Behavioral) Data

Within JAGS, withholding data is simple because JAGS essentially treats missing values as latent variables, and will perform inference on these variables along with the model parameters. Hence, the strategy would be to simply replace values with our data vector with the character string "NA" as a way to tell JAGS to perform inference on these variables, effectively hiding these data from the model. To examine this more concretely, we performed a simulation study. First, we generated data from the joint model as described in Chap. 2, and then defined a new variable, called "B1", that overrides some elements of the variable "B" with NA values. In R, the following few lines perform this operation:

```
# define indices that will be withheld from the model
idx=1:4
# declare new variable B1
B1 <- B
# override values of B with missing observations (NA)
B1[idx] <- NA
```

A similar procedure could be done for the neural data by replacing some elements, rows, or columns of the matrix N with NA values.

With B1 declared, we need only control which vector of behavioral data is passed to JAGS when performing the estimation. For example, to fit a joint model to the partial data B1, we need only replace the JAGS model declaration from Chap. 2 with the following statement:

```
# specify the jags model:
# locate the JAGS code, pass variables, setup sampler
jags <- jags.model('model_general.txt',
                   data = list('n'=n,
                               'B'=B1,
                               'N'=N,
                               'ts'=ts,
                               'Nt'=length(ts),
                               'sig'=sig,
                               'I0'=diag(2),
                               'n0'=2,
                               'phi0'=rep(0,2),
                               'epsilon0'=diag(2)),
                   n.chains = 4,
                   n.adapt = 1000)
```

The other lines of code can remain unchanged and the same model will be fit to the new data set. In our analysis below, we compare a joint model fit to the data B to a joint model fit to data B1 to illustrate how the estimates of the single-trial parameters corresponding to the trials determined by the idx variable change when data are either observed or withheld.

3.3.2 A Behavioral-Only Model

For this illustrative example, we wished to show how a joint model's predictions for new data compared to the predictions of a model that ignored the neural covariates, a model we refer to as the "behavioral-only" model. In this model, because we do not have a subcomponent model for the neural data, a number of modifications are required to make it appropriate for fitting the behavioral data hierarchically. As a consequence, a comparison of the behavioral-only model with the joint model may not be straightforward, although we can make the models similar in the hopes that the data overwhelm the contribution of the prior in the estimation of the model parameters.

As in the specification of a joint model in Chap. 2, we first need to specify the model within JAGS, and we can save it as a separate text file that can be called in R. The JAGS code corresponding to the behavioral-only model we used here follows:

```
# JAGS code, file named ``model_behave.txt"
model {
    # convert sig to tau for convenience
    tau <- pow(sig, -2)

    # loop through trials to define likelihood
    for (i in 1:n){
            # likelihood for behavioral data
            B[i] ~ dbin(1/(1+exp(-Theta[i])),1);
    }

```

```
    # loop through trials to define prior on (delta, theta)
    for(i in 1:n){
        Theta[i] ~ dnorm(phi,Omega);
    }

    # priors on hyperparameters
    phi ~ dnorm(phi0,s0);
    Omega ~ dgamma(a0, b0);
}
```

Note that in this code, several lines have changed. In particular, the likelihood portion for the neural data has been removed, and the parameter matrix "Theta" appears rather than "DeltaTheta". Furthermore, the priors in the behavioral model are substantially different. First, the single-trial parameter vector "Theta" is described via a normal distribution rather than a multivariate normal distribution, the prior for the hyper mean parameter "phi" is normal in form, and the prior for the precision parameter "Omega" is gamma in form.

To execute the JAGS code within R, the following R handler code can be used in the same way as described in above:

```
# specify the jags model:
# locate the JAGS code, pass variables, setup sampler
jags <- jags.model('model_behave.txt',
                   data = list('n'=n,
                               'B'=B1,
                               'sig'=sig,
                               'a0'=.01,
                               'b0'=.01,
                               'phi0'=rep(0,1),
                               's0'=1),
                   n.chains = 4,
                   n.adapt = 1000)
```

The other features of the rjags package can be used to ensure that the adaption period is complete and samples have been drawn from the stationary posterior distribution. Note that in Line 5 we have specified that the data to be fit is B1 (i.e., the data with missing observations), but that the data B could also be passed in this line to compare the single-trial estimates.

3.3.3 Assessing Predictive Accuracy

To illustrate the effects of neural covariates and behavioral data, we fit a total of four models. The first two models were fit to the full behavioral data (i.e., the data vector B). The first of these two models was a joint model that used the neural covariates to inform its estimates of the single-trial behavioral model parameters (i.e., θ). The second of these two models was a behavioral-only model that ignored all neural inputs. The second two models were fit to the partially observed behavioral data

(i.e., the data vector B1). Again, both a joint model and a behavioral-only model were fit to the partially observed data so that their estimates could be compared to the models who were fit to the full behavioral data set.

Once each of the four models has been fit to data, we can examine the resulting posterior distributions from each model for the four trials under investigation. We can do this in R by calling either the density or histogram functions, and overlaying them in a single plot. Figure 3.3 shows the estimated posterior distributions for each trial in question along with the posterior estimates for each of the four models: joint model with no missing data (black lines), joint model with missing single-trial data B_i (blue lines), a behavioral-only model with no missing data (gray lines), and a behavioral-only model with missing single-trial data (light blue lines). Within each

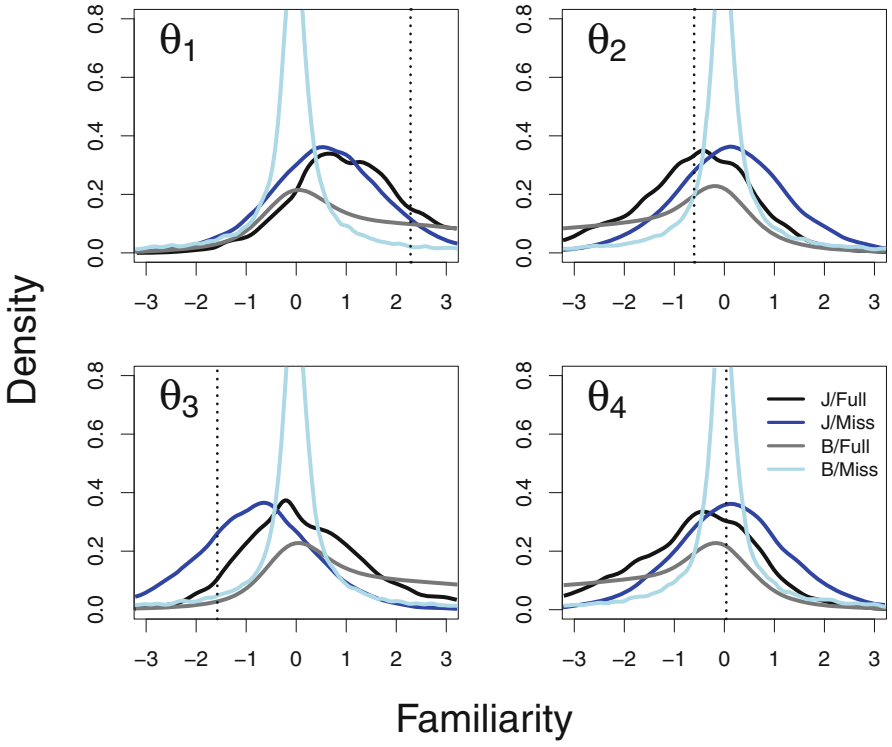


Fig. 3.3 Predicted parameter estimates for four out-of-sample trials from four different models: joint model with no missing data (black lines), joint model with missing single-trial data B_i (blue lines), a behavioral-only model with no missing data (gray lines), and a behavioral-only model with missing single-trial data (light blue lines). The true value of each single-trial parameter is shown as the dashed vertical line. Trials 1 and 3 (left panel) resulted in correct recall performance whereas Trials 2 and 4 (right panel) resulted in incorrect recall performance. Note that the data in these four trials were not used in the inference process, and so the posteriors are pure generalizations of the hyper structure imposed in each model

panel, the true value of each single-trial parameter is shown as the dashed vertical line. In the data we simulated, an "old" response was elicited on Trials 1 and 3 (left panel), whereas a "new" response was elicited on Trials 2 and 4 (right panel).

There are a number of important differences in the posteriors from one model to the next. First, Fig. 3.3 shows that the behavioral-only model with missing observations has the least amount of variance for each posterior. In most of our applications of joint modeling in the past, the typical result is that joint models tend to have less variance relative to behavioral-only models. Here, it seems that the single-trial estimates are only weakly informed by the behavioral data, and as a result, the prior predictive distribution is more narrow in this context. This pattern is well known when using Bayesian statistics, and is called "shrinkage." Essentially, as each parameter is informed by the prior and the observed data, when there is little information in the observed data, the parameter estimates adjust to reflect the information in the prior. Statistically speaking, this is an optimal thing to do, but it does result in conservative parameter estimates that can seem unintuitive. Notice that across the four panels, the prediction for each of the trials from the behavioral-only model with missing data is nearly identical, and for all but Trial 4, it is substantially less accurate relative to the other models. Here, the shrinkage is a bad thing as it would result in poor performance in a cross-validation test. For example, if we compare the joint model with missing data (blue lines) to the behavioral-only model with missing data (light blue lines), Fig. 3.3 shows that the joint model is able to adjust its predictions from one trial to the next, something that the behavioral-only model is unable to do. For these models, this pattern of adjustments from one trial to the next is driven purely by information learned on other trials. For the behavioral-data only model, only the information in the behavioral data constrain these predictions, whereas for the joint model, three sources of information constrain the predictions. First, the joint model uses the information in the behavioral data to generalize to these other trials. Second, the joint model uses the correlation between θ and δ observed across all trials to constrain its predictions. Finally, the joint model uses the neural data on the trials in which the behavioral data are withheld to further enhance its predictions about missing behavioral effects (i.e., θ_i).

Second, θ_1 and θ_2 are best predicted by the joint model with no missing data, θ_3 is best predicted by the joint model with missing data, and θ_4 is best predicted by the behavioral-only model with missing data. In general, we would expect the joint model with no missing data to provide the most accurate predictions, but for these withheld data, occasionally the predictions from other models are more consistent with the true level of familiarity. For example, only a narrow region of the parameter space will be best predicted by the behavioral-only model, but this region happens to be the mean of the single-trial parameters (i.e., ϕ_2). As a consequence, for some trials, the behavioral-only model will perform well relative to the other models, but it will also perform poorly when the level of familiarity is different from the mean, as it will be when σ_θ is large. Finally, when the behavioral data B_3 are inconsistent with the level of familiarity θ_3, the joint model with no missing data pushes its predictive density upward along the axis of familiarity making what appears to be

less accurate predictions for θ_3. However, if these distributions for familiarity where transformed into predictions about the data, the joint model with no missing data would place larger density on the outcome $B_3 = 1$ relative to the joint model with missing data. In effect, the joint model with no missing data will be more accurate in a cross-validation test than when comparing the model on the parameter space.

Third, in comparing both the joint and behavioral-only models with no missing data to those with missing data (i.e., comparing the black to blue lines, and the gray to light blue lines), Fig. 3.3 shows a substantial shift in the posterior estimate along the axis consistent with the behavioral outcome. For example, in Trials 1 and 3 where "old" responses were elicited, both the joint and behavioral-only models predict higher density for larger values of θ when data were not missing relative to models when the data were treated as missing. While this adjustment does enhance the predictions from both models for Trial 1, it deprecates the accuracy in θ space of the models for Trial 3, when an "old" response is highly unlikely given the degree of familiarity θ_3.

3.3.4 Calculating Likelihoods for Missing Data

To determine which model best predicts the withheld data, we must determine the likelihood of the missing data under the model parameters and compare these likelihoods across the four models. The model with the highest likelihood for all four data points is the model that best predicts the missing data. In Fig. 3.3, our task simplifies to determining the density of the posterior at the location of the missing observation under each model. In other words, we must determine the value of the y axis at the dashed vertical line for each model. In R, this can be accomplished by first calculating a kernel density estimate of the posterior samples produced by JAGS, and then interpolating the values to gain an estimate of the density at the desired location. The following two lines of code can be used to produce such an estimate:

```
d1 <- density(out.full$DeltaTheta[j,2,,])
like <- approx(x=d1$x,y=d1$y,xout=DeltaTheta[i,2])
```

The first line produces the kernel density estimate of the model "out.full" on the jth trial, and the second line uses the "approx" function to (linearly) interpolate the grid values used in the density estimation procedure to evaluate the density value at the location of the true parameter value.

While this procedure is useful if the posterior samples have already been calculated, JAGS has the capability of determining the likelihood of the withheld data while simultaneously providing estimates of the desired posterior. Referring back to our example above, assuming we have the variables "B" and "B1", where "B1" is the data vector containing withheld data. If we define the variable "nhold" as the number of withheld observations, we can adjust our base JAGS code to the following, where *both* "B" and "B1" are passed to JAGS:

```
1  # JAGS code, file named ``model_general_prediction.txt"
2  model {
3      # convert sig to tau for convenience
4      tau <- pow(sig, -2)
5
6      # loop through trials to define likelihood
7      for (i in 1:n){
8          for (t in 1:Nt){
9              # likelihood for neural data
10             N[i,t] ~ dnorm(DeltaTheta[i,1]*ts[t],tau);
11         }
12         # likelihood for behavioral data
13         B1[i] ~ dbin(1/(1+exp(-DeltaTheta[i,2])),1);
14     }
15
16     # calculate the likelihood for the withheld data
17     for(i in 1:nhold){
18          like[i] <- dbin(B[i],1/(1+exp(-DeltaTheta[i,2])),1);
19     }
20
21     # loop through trials to define prior on (delta, theta)
22     for(i in 1:n){
23         DeltaTheta[i,1:2] ~ dmnorm(phi,Omega);
24     }
25
26     # priors on hyperparameters
27     phi ~ dmnorm(phi0,s0);
28     Omega ~ dwish(I0,n0);
29     # convert Omega to Sigma for convenience
30     Sigma <- inverse(Omega);
31 }
```

Here, Line 18 shows that the "dbin" function can be used with three arguments to calculate the density of the data "B" on the ith trial, whereas in Line 13 the "dbin" function is used to specify the model. If the variables "B", "B1", and "nhold" are passed properly to JAGS, we can simply monitor the variable "like" when drawing samples from the posterior in JAGS to arrive at a Monte Carlo estimate of the likelihood for the withheld data. Aggregating across iterations and chains provides a single value for each missing data point, which can then be compared across the models of interest.

3.4 Conclusions

In this chapter, we have discussed a few important considerations in the development of any model. While the first section discussed the role of model flexibility in fitting empirical data, the second section focused on the goal of generalization. Assessing a model's ability to generalize is especially important for joint models, where one can test whether having additional covariates such as neural measures are constraining

the model parameters in important ways. For example, a common theme in our research is to fit a joint model alongside a model that only considers behavioral data. These models are not directly comparable through the marginal likelihoods because the structure of data that they consider is not equivalent [18]. However, the models do share a common dimension in that they both consider behavioral data. Because of this common ground, the models can be compared via out-of-sample generalization tests. The degree to which a joint model outperforms a behavioral-only model can then be used to justify the complexity and appropriateness of the linking structure connecting neural and behavioral data [12, 68], as well as the number of possible covariates in the data set [46]. The final section was meant to facilitate methods for performing these generalization tests, so that once you are familiar with how to fit a joint model to data from Chap. 2, you'll be able to appropriately substantiate your linking hypotheses connecting your neural data to your decision model [3].

Chapter 4
Applications

At this point, there have been several successful applications of joint models in the literature. These applications differ along a number of dimensions, two of which are (1) the type of neural measure under consideration, and (2) the type of decision model used to describe the behavioral measures. Table 4.1 organizes several joint modeling applications along these two dimensions. In this chapter we highlight four such applications of joint modeling to different contexts where the integration of neural and behavioral data proved useful in unveiling the underlying cognitive dynamics that were not apparent with behavioral data alone. In the first application, we discuss how structural information in the form of diffusion weighted imaging (DWI) can be used to better constrain a model of evidence accumulation fit to data from a perceptual decision making task with a speed accuracy manipulation. Second, we discuss a recent extension of this approach to functional imaging data obtained using magnetic resonance imaging (fMRI) on a trial-by-trial level. In particular, we discuss how pre-stimulus information can be used to enhance predictions about the speed and accuracy of decision making processes once the stimulus is presented. Third, we discuss how the joint modeling approach can be used to exploit the temporal resolution provided by electroencephalography (EEG) and the spatial resolution provided by fMRI to better constrain a cognitive model. We show that while these two measures provide similar information about the decision making process, the information is not identical within a subject, and as a result, having more neural covariates can enhance model predictions for withheld data. Finally, we discuss how joint modeling can be applied to single-unit recording data from monkeys, highlighting that our approach is not modality specific.

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_4

Table 4.1 Reference table for applications of joint models in the literature. Rows correspond to the type of neural measure under consideration, whereas columns correspond to the type of decision making model used. *LBA* Linear Ballistic Accumulator Model; *DDM* Diffusion Decision Model

	Cognitive model	
Neural measure	LBA	DDM
Structural data	Turner et al. [12] Turner [45]	Turner [45]
Single unit recordings	Cassey et al. [83]	Frank et al. [34]
EEG	Turner et al. [46] Van Ravenzwaaij et al. [40]	Nunez et al. [33] Nunez et al. [35]
fMRI	Turner et al. [46]	Turner et al. [18] Turner et al. [68] Palestro et al. [70]

4.1 Structural Relationships to Decision Making

In our first real-world application, we focused on the problem of relating structural differences between the pre-supplementary motor area (pre-SMA) and the striatum to speeded decision making. The stratal hypothesis central to our investigation asserts that under time pressure, the striatum decreases the activation of the output nuclei of the basal ganglia, thereby releasing the brain from inhibition and facilitating decisions that are fast but error-prone [84, 85]. Previous research had already supported the notion that higher structural or functional connectivity between the striatum and the pre-SMA regions was indicative of an observer's ability to flexibly adjust the level of evidence needed to make a speeded response within the task following the emphasis of the experimenter [86–88].

The data were presented in Forstmann et al. [89] and were produced by 20 young subjects and 14 elderly subjects. The experiment used a moving dots task where subjects were asked to decide whether a cloud of semi-randomly moving dots appeared to move to the left or to the right. Before each decision trial, subjects were instructed to either respond quickly (the speed condition), accurately (the accuracy condition), or at their own pace (the neutral condition). Following the trial, subjects were provided feedback about their performance: in the accuracy condition, subjects were told when their responses were incorrect, whereas in the speed condition, subjects were told when their responses were too slow. These additional feedback mechanisms helped to further emphasize the variable of greatest importance within the task, be it speed or accuracy. Each subject completed 840 trials, equally distributed across the three condition types.

The neural data consisted of the corticostriatal connections between the left and right pre-SMA and the left and right striatum, as measured by structural diffusion-weighted imaging (DWI). From the DWI measures, a probabilistic tractography [90] measure was computed as a proxy for white matter connectivity between different

cortico-subcortical brain regions. The brain regions presented in Fig. 4.2 illustrate the two brain areas from which the tractography measures were derived [86].

To investigate how tractography measures were related to mechanisms of a cognitive model, we fit a joint model to the behavioral and neural data. For the behavioral data, we used the mathematically simple Linear Ballistic Accumulator (LBA) [36] model, which assumes all the conventional mechanisms of perceptual decision making except a within-trial variability component. Despite this, the LBA model has been shown to provide similar interpretations as more sophisticated models [91], and so it was convenient for our purposes. Within the joint model, we related each model mechanism to the tractography measures in the neural submodel through a multivariate normal distribution as described in Chap. 4. While this particular linking hypothesis is perhaps overly flexible, it does allow us to investigate relationships between the model parameters that we might not have expected a priori. As this was the first joint model application, we chose to remain agnostic about the specification of brain-behavior links.

Although the joint model analysis provided a wealth of information about how the model parameters were related to the structural data, perhaps one of the most interesting results was the model's ability to generate predictions about behavioral data on the basis of the structural information alone. To highlight this feature of the model, we performed a simulation study with two conditions worth of withheld data. In the first condition, we removed all observations from two of the 24 subjects: one subject had a high average tract strength value (0.973) and one had a low tract strength value (0.405). In the second condition, we removed only 50% of the data from the same two subjects. We then fit a joint model and a behavioral-only model to the data within each condition. After fitting both models, maximum a posterior (MAP) estimates were calculated from the joint posterior distributions, and these estimates were used to generate a predictive density estimate for the behavioral data. The MAP estimates serve as the best single parameter value for the data that were observed, and as such, the resulting predictive density is the model's best attempt at generalizing to new data.

Figure 5.4 shows the results of the simulation study for the accuracy condition only. The left panel shows the raw data for the low tract strength subject (black lines) and the high tract strength subject (red lines). In each panel, the correct RT distributions are shown on the right whereas the incorrect RT distributions are shown on the left. The right panel shows the posterior predictive distributions for the joint (top) and behavioral-only (bottom) models for the first (i.e., the condition where 100% of the behavioral data were withheld) and second conditions (i.e., the condition where only 50% of the behavioral data were withheld), respectively. For the first condition, the figure shows that the predictions for the behavioral model do not differ across the two subjects because the model has no information that can be used to dissociate one subject from another. On the other hand, the predictions from the joint model show a clear separation between the two subjects as a result of the information learned from the neural data of the two subjects and the relationship learned between the behavioral and neural model parameters from the remaining 22 subjects. In particular, the joint model correctly predicts that the high tract strength

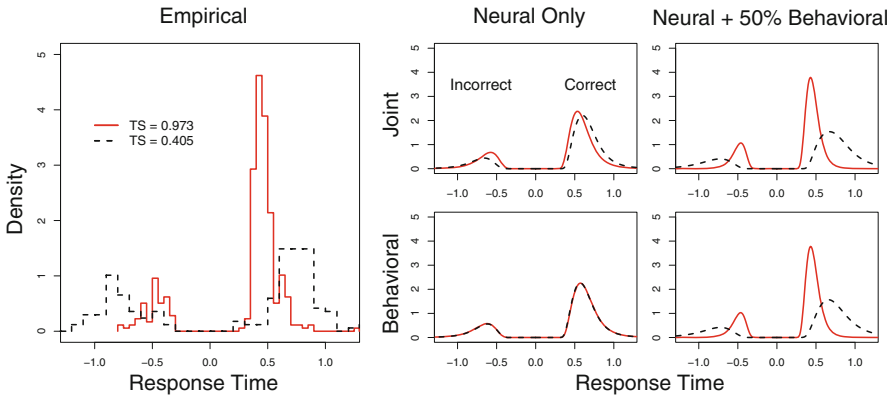


Fig. 4.1 A model prediction comparison for a subject with a high average tract strength measurement (red lines) and a subject with a low average tract strength measurement (black lines). The left panel shows a histogram of the raw data whereas the right panels show the predictions of the joint (top) and the behavioral (bottom) models under two conditions: the left column corresponds to a condition in which all behavioral data are withheld and the right column corresponds to a condition in which only 50% of the behavioral data are withheld. Whereas the joint model uses information in both the prior and the neural data, the behavioral model only uses the prior information. Only the joint model is able to differentiate the predictions for the two subjects when only neural data are available. In each panel, the RT distribution for incorrect responses are shown on the left whereas the RT distribution for correct responses are shown on the right

subject will tend to produce faster and more accurate responses than the low tract strength subject, which was an empirical regularity in these data (Fig. 4.1).

While the neural data for this real-world experiment are sparse, the joint model was able to identify a signal in the data amplified by the hierarchical model. However, due to the sparsity of the neural data, a principled model would predict that as the number of observations from the behavioral data grows, the predictions of the joint and behavioral-only models should converge because the proportion of shared information between the two models increases. The right panel of Fig. 5.4 shows that this is indeed the case: the predictions of the joint and behavioral-only models are nearly identical when 50% of the behavioral data are available. Furthermore, the predictions of both models accurately reflect the patterns of observed data from the two withheld subjects (leftmost panel). This implies that the joint model subsumes the behavioral-only model when enough behavioral data are available, but by virtue of having the neural covariate, the joint model can better generalize to sparse data than can a behavioral-only model.

4.1.1 Greater Constraint on Model Evaluation

While the joint model we applied above was useful in articulating how corticostriatal relationships manifest as mechanisms in a cognitive model, the types of connections that can be appreciated are necessarily constrained by the version of the model under

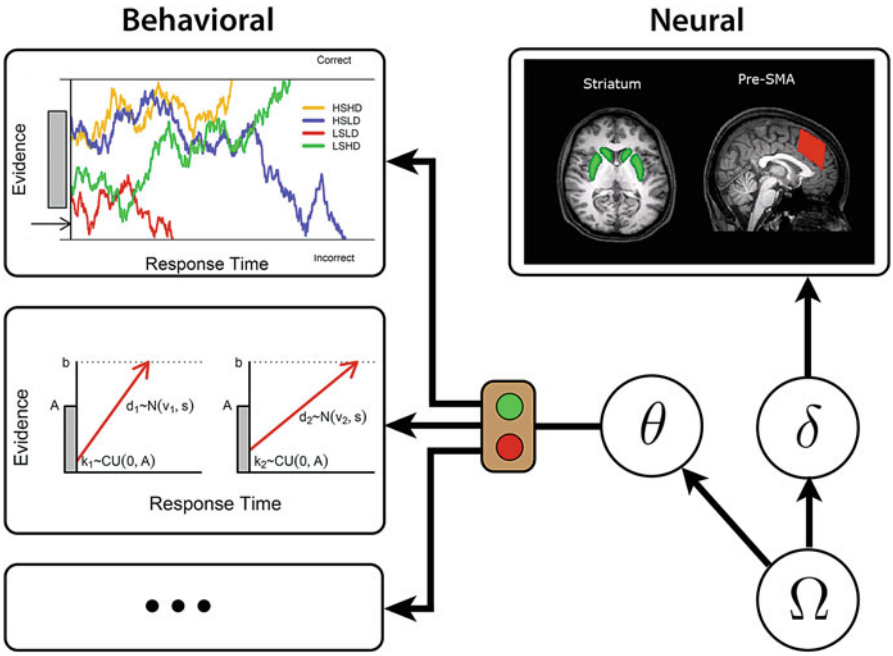


Fig. 4.2 Behavioral submodel interchangeability. Joint models are not committed to any particular submodel of the data, and as a result, one submodel can be substituted into the model fit as a way to evaluate the relative fidelity of brain-behavior relationships. The green/red button box reflects the switch that occurs when mapping a set of neural covariates to one model or another

our employ. Because we chose the LBA model as the behavioral submodel, we are limited by our model's ability to account for the behavioral data, and ultimately how the model fits the behavioral data may dictate the interpretation of which mechanism is most strongly associated with a given brain measure. Hence, this limitation begs the question of whether or not a different cognitive model might reveal a different brain-behavior dynamic.

Fortunately, joint models are agnostic about the submodel specification for a particular data modality. This means that one can choose a submodel on the basis of convenience (e.g., mathematical tractability), theoretical endorsement, or simply personal preference. In other words, if we wished to investigate whether a different theory about the processes underlying a decision could capture the brain-behavior relationship better, we need only to fit each of the models under consideration to the data and use conventional fit statistics to evaluate their relative merits.

Figure 4.2 illustrates the idea of submodel interchangeability. On the left side, one can choose between various submodels of the same behavioral data. As an example, the DDM is pictured on the top row, and the LBA model is pictured on the middle row. As other models come into consideration, they could also be compared to the other models, a feature that is illustrated in Fig. 4.2 as the plate with an ellipses. On the right side, the submodel for the neural data can remain fixed across

models, as there are rarely multiple theories about how the brain measures might arise. Under these assumptions, only the likelihood function $\mathscr{L}(\theta|B)$ needs to be adjusted, and potentially the size of the hyperparameters ϕ and Σ when $\Omega = \{\phi, \Sigma\}$ to accommodate different numbers of model parameters θ. Each unique submodel selection constitutes a new joint model of the behavioral and neural data, and after fitting each joint model to data, model fit and generalizability can be assessed properly.

Turner [45] performed such an analysis comparing a simplified version of the DDM [37, 92] (i.e., one that does not assume any trial-to-trial variability in drift rate, starting point, or nondecision time) to the LBA model [36]. Because the models make slightly different assumptions about the underlying cognitive processes at work in a choice response time experiment, a comparison of the two models on the basis of model fit would provide support for the assumptions made by the better-fitting model. Of course, this examination is highly oversimplified and is in no way a definitive claim of superiority of one model over another. Performing such a task would require many more data sets, a comparison of multiple variants of each model, and potentially the inclusion of many more neural covariates.

After specifying the models in their most "out-of-the-box" form, Turner [45] fit both the simplified DDM and LBA models to the data from Forstmann et al. [89] using DE-MCMC [67]. Some modification of the linking function was necessary to accommodate the parameter dimension mismatch in θ, where the LBA model possessed eight parameters and the simplified DDM possessed seven parameters. After fitting the models to data, Turner examined the resulting posterior distributions of the correlation parameters ρ contained within the variance-covariance matrix Σ. The pattern of correlations between the LBA model parameters and tract strength was similar to the pattern of correlations between the DDM model parameters and tract strength, which might be expected based on previous comparisons of these models for behavioral data alone [91, 93].

The correlations with the threshold parameters are similar to the LBA model's account. Specifically, the correlation between the response threshold parameter and tract strength was negative in the accuracy and neutral conditions, but was essentially zero in the speed condition. The remaining parameters were less clear. There was a slight negative correlation between tract strength and both the starting point and nondecision time parameters, and the drift rates more mostly uncorrelated with tract strength. Together these results suggested that both models associated tract strength with response threshold, something that is consistent with the original findings on the striatum hypothesis [86–89].

Perhaps the most relevant question is how well the two models fit the data. To compare the models, Turner [45] used conventional Bayesian measures of model fit – deviance information criterion (DIC; [94]), and the Bayesian predictive information criterion (BPIC; [95]) – that balance all of the important aspects of model comparison: model complexity, number of parameters, and degree of fit to the data [75, 76, 96]. Turner compared the fits of the two models to both the group data and the individual subject level. In the group analysis, the LBA model provided the best fit to the data, as measured by both the DIC and BPIC measures. In the

individual-level analysis, the LBA model fit the data for 20 of the 24 subjects better than did the simplified version of the DDM used in the analysis.

While the model comparison exercise in Turner [45] was by no means definitive, it hopefully provides some inspiration for performing model comparisons on the basis of both brain and behavioral data. Hence, the model comparison was useful in illustrating that joint models can be used as a basis for testing different theories about how behavioral data arise, similar to other types of brain-behavioral modeling preceding our approach [13, 97].

4.2 Single-Trial Linking to Decision Processes

While initial joint model explorations were useful in linking structural information in the brain to a decision making process, Turner et al. [18] wondered whether the general linking framework could be extended to functional data as well. Indeed, most functional measures such as those obtained by EEG or fMRI consist of a high-dimensional, continuous read out of a brain process on a trial-by-trial level, providing a rich data set for exploring the dynamics of what is ostensibly a cognitive process. Considering this, a natural question to ask is whether trial-by-trial neurophysiological data can be used to contain cognitive models even further than constraints at the subject level [12, 45].

To investigate the types of constraint provided by functional data, Turner et al. [18] extended the preliminary joint modeling framework to capture the trial-to-trial covariation between brain states, as measured by fMRI, and the parameters of a perceptual decision making model. The problem Turner et al. addressed ultimately centered on a lack of information about within-trial accumulation dynamics. In behavioral choice response time experiments, following the presentation of a stimulus, researchers can only observe the eventual choice and response time when considering behavioral measures alone. Traditionally, models of decision making such as the DDM [37] and LBA model [36] assume the presence of trial-to-trial fluctuations in the parameters explaining the decision process. These parameters are assumed to vary from one trial to the next in an (apparently) arbitrary fashion [98, 99], but are constrained to fluctuate according to a *particular* distribution across trials [100]. However, the trial-to-trial variability assumption – known as stationarity – imposed by conventional models is a strong one, and is seldom observed in empirical data [101, 102]. Perhaps more interesting is the wealth of literature on the so-called "default model network" that implicates a gradual waxing and waning of attention directed at the task, suggesting that at least the neurophysiolgical measures of cognition are anything but stationary [103–109].

To reconcile these brain-behavior differences, the goal in Turner et al. [18] was to provide some guidance into the distribution of these trial-to-trial measures by enforcing constraint from trial-to-trial measures of the BOLD response. Although the model made no explicit statement about sequential dependencies in the decision process, binding the model parameters to the neural covariates implicitly enforced a sequential structure by virtue of the neural measures themselves.

While the mathematical details of the extended joint model are described in Turner et al. [18], the gist of the approach is quite simple, and a simplified version can even be specified in JAGS [70]. For the neural submodel, Turner et al. described the degree of activation of a set of brain regions of interest (ROIs) with a set of single-trial β parameters using a method developed in [107]. The neural measures were BOLD responses, which were collected for each of 14 subjects over about 80 trials per subject. The behavioral measures consisted of a standard random dot motion task where subjects were instructed to respond "left" or "right" depending on which direction most of the dots were perceive to be moving toward. For the behavioral submodel, Turner et al. used the DDM to describe the choice response time distributions from the task. Importantly, the model assumed allowed the drift rate and starting point parameters to vary from one trial to the next, along with a set of parameters (e.g., the threshold) which were fixed across trials. Finally, Turner et al. assumed that the single-trial ROI activation functions and the single-trial DDM parameters (i.e., the drift rate and starting point) were distributed according to a multivariate normal distribution.

4.2.1 Generative Analysis

Once the model was fit to the data, Turner et al. [18] performed a number of analyses to interpret the pattern of activation in the brain through the lens of their cognitive model. Several of the analyses were based on an examination of the posterior predictive distribution (PPD) of the model parameters. The PPD serves as a generalization of the information obtained in the empirical data to new, hypothetical data that might have been observed had more trials been obtained in the experiment. The PPD provides a statistically coherent way to simultaneously form a quantification of uncertainty, and establish a "best" estimate for the predicted model parameters, based on the data that were observed. The PPD analysis is particularly advantageous when the data are sparse relative to the number of model parameters, as they were in this application. Because the analysis focused on the trial-to-trial brain-behavior covariation, Turner et al. generated the PPD from the hyperparameters of the multivariate normal distribution. The result was a set of starting point and drift rate parameters for new, hypothetical trials.

Once the PPD was generated, Turner et al. [18] defined regions within the PPD that corresponded to psychologically interpretable constructs, such as the rate of information processing and response efficiency. Defining the regions in this way allowed Turner et al. to glean psychologically meaningful constructs on the basis of the parameters in the model. Perhaps the most relevant analyses to the readers of this book are the drift rate analysis shown in Fig. 4.3. Within the DDM, the rate of stimulus information processing corresponds to the drift rate, and the degree of bias in the choice corresponds to the starting point. Figure 4.3a shows the PPD for drift rates (x-axis) and starting points (y-axis) generated from the model fits. Each colored region in Fig. 4.3a corresponds to a particular type of behavioral response pattern. For this analysis, Turner et al. defined one region corresponding

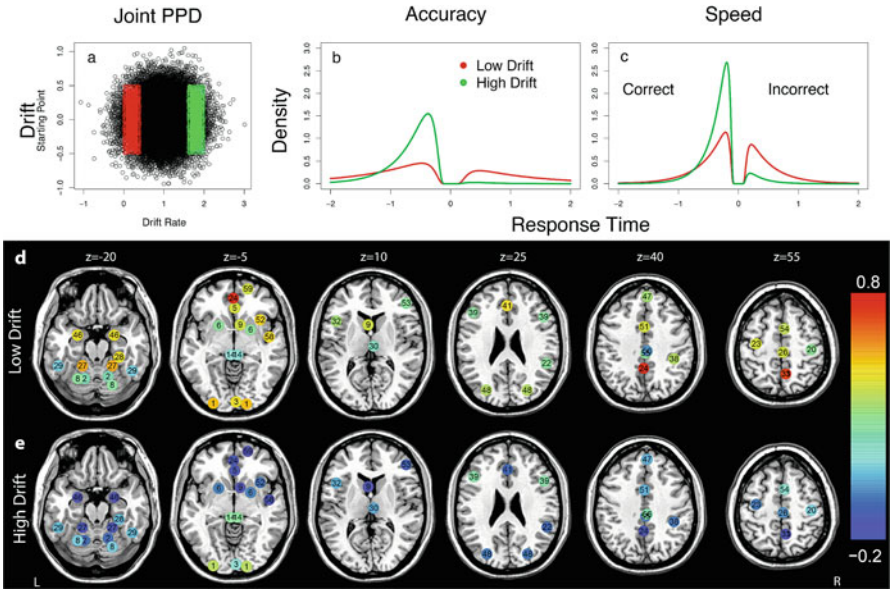


Fig. 4.3 Results of model-based analysis of BOLD data. (**a**) The joint posterior distributions of the drift rate and starting point parameters. Within Panel **a**, two parameter regimes were defined: one corresponding to a slow, error-prone decision process (i.e., the red area), and one corresponding to a fast, accurate decision process (i.e., the green area). (**b** and **c**) Model predictions conditional on the parameter regions in Panel **a** for choice response time distributions for accuracy (**b**) and speed (**c**) emphasis conditions, respectively. (**d** and **e**) Mean predicted BOLD signal as a function of the parameter regimes shown in Panel **a**: Panel **d** corresponds to the slow, error-prone decision process (i.e., the low drift rate region), and Panel **e** corresponds to the fast, accurate decision process (i.e., the high drift rate region). The columns correspond to six axial slices, moving from ventral to dorsal surfaces. Each node corresponds to an ROI, and the node's color represents the degree of activation according to the color key on the right hand side

to slow, error prone decision making (i.e., the red region of the PPD in Fig. 4.3a), and another region corresponding to fast, yet accurate decision making (i.e., the green region of the PPD in Fig. 4.3a).[1] Using the parameters in these regions, one can also generate predictions from the model about the patterns of behavioral data to ensure that the chosen parameter regions are consistent with the definitions made about the type of decisions made. Figure 4.3b, c shows the predicted choice response time distributions for each correspondingly-colored region in the joint posterior distribution under accuracy and speed emphasis instruction, respectively.[2]

[1]Here, the fast, accurate decision process corresponded to a mean value for drift rate of $\xi = 1.8$, with an interval of $\pm$ 0.2, and the slow, error-prone decision process corresponded to a mean value for drift rate of $\xi = 0.2$, with an interval of $\pm$ 0.2.

[2]The figure only shows the model's predictions for stimuli with rightward directional motion because the predictions for leftward motion stimuli were mirror images of the rightward motion stimuli.

In each of the response time distribution plots, distributions corresponding to the correct decision are shown on the left (i.e., negative values), whereas distributions corresponding to the incorrect decision are shown on the right. In each panel, the model's prediction for the probability of a correct response is represented as the density of the correct response time distribution relative to the incorrect response time distribution. These figures show that the high drift rate region produces responses that are both more accurate and faster than the low drift rate region. In fact, for the low drift rate region, the model predicts that responses will have accuracy that is near chance.

Although Fig. 4.3b, c show the model's predictions about behavioral data, in some sense these predictions are expected because Turner et al. [18] chose parameter values to correspond to desired types of behavioral decisions. What is less well understood is how these parameter settings in the model relate to patterns of activity observed in the brain. To investigate this, Turner et al. examined the distribution of predicted BOLD signals for each ROI conditional on the two parameter regions chosen in Fig. 4.3a. Figure 4.3d, e shows the mean predicted BOLD signal for each ROI during low (i.e., the red region in Fig. 4.3a) and high drift rate trials (i.e., the green region in Fig. 4.3a), respectively. Each ROI is represented as a "node" appearing on the nearest axial slice in MNI coordinates, and is labeled according to Table 4.1 in Turner et al. [18]. The mean predicted BOLD signal is color coded according to the key on the right hand side. Using the PPD analysis shown in Fig. 4.3, Turner et al. identified brain regions that were associated with the slower information processing by locating ROIs with high BOLD activity in Fig. 4.3d and low activity in Fig. 4.3e.

4.2.2 Predictive Analysis

Although the generative analyses in the previous section are useful in that they can provide an understanding of how the brain relates to the mechanisms in the cognitive model, another utility of joint modeling is to use the model to make predictions about new data. Turner et al. [18] also examined the joint model's ability to predict withheld data relative to a behavioral-only model in a cross-validation test. To do this, they first created a "test" set by randomly removing 100 trials from the full data set. Next, they refit both the joint model and the behavioral-only model to the remaining "training" data. Once the models had been fit to the training data, they used the resulting posterior distributions to generate predictions for the single-trial drift rate and starting point parameters. As an example, Fig. 4.4a, b show these PPDs for the (single-trial) drift rate parameter ξ and (single-trial) starting point parameter ω, respectively, for the joint model (green histograms) and the behavioral-only model (red histograms) on Test Trial 48, which was randomly chosen.

For Trial 48, while the PPDs for the starting point parameters are nearly identical, the PPDs for the drift rate parameters differ in two important ways. First, the PPDs for the drift rate parameter are more constrained (i.e., have smaller variance)

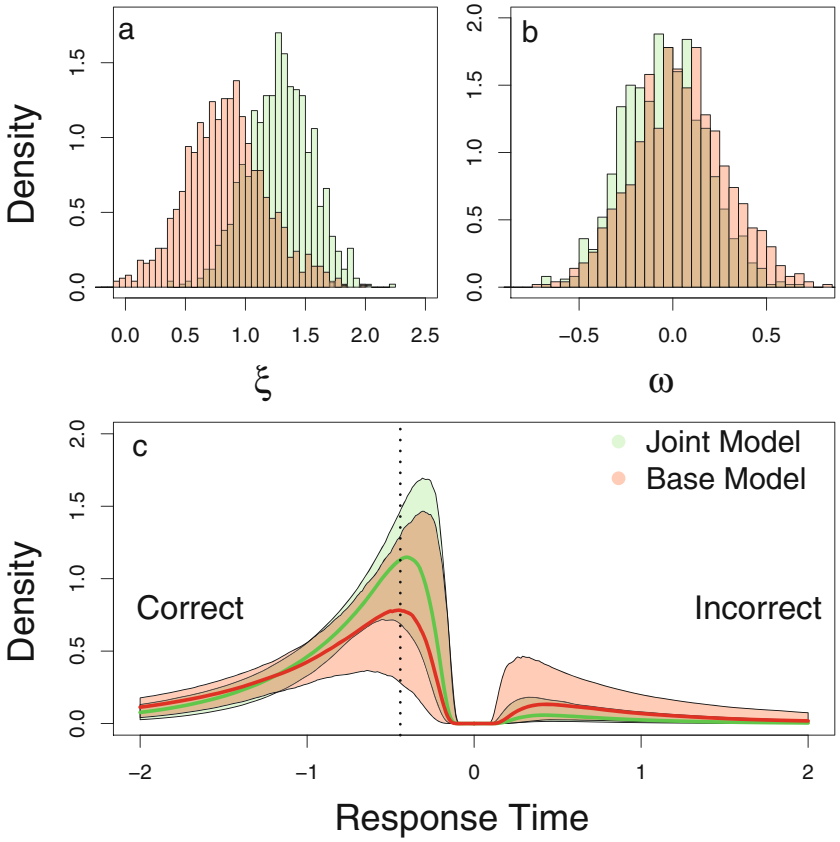


Fig. 4.4 Posterior predictive distributions of model parameters and data variables for Test Trial 48. In all panels, joint model distributions are shown in green whereas behavioral-only distributions are shown in red. The top panels show the posterior predictive distributions for (**a**) the (single-trial) drift rate ξ and (**b**) the (single-trial) starting point ω. Panel **c** shows the PPD for the choice response time distributions for correct (left) and incorrect (right) responses. For each model, 95% credible sets are plotted along with the best prediction illustrated with a solid line of corresponding color. The dashed vertical line represents the test trial data (i.e., a correct response with a response time of 441 ms)

for the joint model relative to the behavioral-only model. Turner [45] showed that the additional constraint on the behavioral model parameters is guaranteed as long as a nonzero correlation exists between *at least one* neural source and a particular latent variable. In Turner et al. [18], several strong correlations between the model parameters and the neural sources were observed, and as a consequence, the information in the neural data better constrain the cognitive model. Second, the means of the PPDs for the drift rate parameter are different between the joint model and the behavioral-only model.

Once the PPDs had been generated for each single-trial model parameter, Turner et al. [18] generated a PPD for the behavioral data. Figure 4.4c shows the PPDs in data space, where green corresponds to the joint model and red corresponds to the behavioral-only model (i.e., corresponding to Fig. 4.4a, b). The 95% credible set for the response time distributions of correct (left) and incorrect (right) responses are shown for each model, along with the median of the PPDs shown in corresponding color. Comparing the two credible sets reveals that the predictions under the joint model are less variable relative to the behavioral-only model. The dashed vertical line represents the data for Test Trial 48, which was a correct response with a response time of 441 ms. To compare the accuracy of the predictions for both models, Turner et al. [18] evaluated the density of the test trial data under the response time curves. If a model places higher density at the location of a particular test trial data point, the predictions are more accurate because the model believes the test data are most likely to occur at this location. Hence, the higher the density around the test trial data, the more accurate the predictions from the model. For Test Trial 48, the joint model places a higher density at this response time than the behavioral-only model, making it the preferred model for this test trial.

Turner et al. [18] repeated the model comparison process illustrated in Fig. 4.4 for the remaining trials in the test set. However, the full analysis also took into account the degree of variability in the predicted densities for each test trial (i.e, the entire shaded region in Fig. 4.4c). Evaluation of the predicted densities from the PPD produced a distribution of 1,000 density values for each model. Turner et al. then compared the distributions across models using a Monte Carlo procedure [110]. Ultimately, the analysis showed strong support for the joint model over the behavioral-only model. Within the test set, 63 of the 100 trials were better predicted by the joint model. In a replication analysis, Turner et al. observed even stronger results when different test sets were generated, where the joint model consistently outperformed the behavioral-only model ranging from 61 to 87 trials out of 100.

4.3 Multi-modal Integration: Combining Behavior, EEG, and fMRI

In another important application of joint modeling, Turner et al. [46] extended the approach to multiple neurophysiological measures. At the time, the field of model-based cognitive neuroscience had established several links between both structural measures such as diffusion-weighted imaging [12, 88], and functional measures such as EEG [33], fMRI [11, 13, 18, 86, 88, 111–114] and magnetoencephalography [115]. These efforts, while elucidating, had focused on relating model parameters to only a single modality of neural measure. However, the limitations of these modalities in their application to understanding cognition are well known. Superficially, measures like EEG provide excellent temporal resolution with a continuous read out of brain activity propagated to the scalp. These measures are well suited

for studying fine-grained details of a decision process where many brain regions interact to perform a given task, but they are limited by the specificity of spatial information they can provide. On the other hand, measures like fMRI provide good spatial resolution, but offer limited information on the temporal properties of the interesting neural signature.

Given the orthogonal nature of the benefits and limitations of EEG and fMRI, Turner et al. [46] wondered whether the joint modeling framework could be used as a way to capitalize on the temporal and spatial resolutions provided by each modality. Of course, many other researchers have pursued this problem before by developing several methods known as "data fusion" techniques. For example, one can use partial least squares correlation [116, 117], independent component analysis [118–127], structural equation modeling [128, 129], multiple regression [130, 131], and canonical correlation analysis [132–134] to fuse functional neuroimaging data. However, the focus of these methods is either source localization [135, 136], or relating behavioral measures (e.g., response times) to brain data [137]. While these methods have produced several theoretically interesting results [135, 137], they are unable to directly inform the latent parameters of cognitive models.

Given the technical hurdles involved in simultaneous EEG/fMRI, Turner et al. [46] investigated an alternative for performing EEG/fMRI integration. Instead of obtaining the neural measures simultaneously, Turner et al. used joint modeling as a way to work around potential artifacts by simply aggregating the effects of interest across experimental contexts. In other words, the models describe effects present in both EEG and fMRI by binding these effects to changes in the behavioral data. If the behavioral data remain consistent from one context to another, the manner in which the behavioral data manifest should be reflected in similar ways across the EEG and fMRI modalities. In this way, the behavioral data serve as a bridge connecting the two modalities to one another, even if the EEG and fMRI data were obtained at different points in time for the same subject, or say EEG data were collected for some subjects and fMRI data were collected for others.

To describe the trivariate model used in Turner et al. [46], we first describe the data from their experiment. Figure 4.6 provides a summary of the data while Fig. 4.5 illustrates the structure of the experiments. Turner et al. [46] used an intertemporal choice task where subjects were presented with two offers on a screen consisting of a monetary value r and a time delay t. The subjects were instructed to choose which of the two offers they preferred. The first offer was always a reference point or "immediate" offer of receiving $r_i = \$10$ now (i.e., $t_i = 0$), or some other "delayed" (r_d, t_d) pair where $r_d > r_i$, but $t_d > t_i$. In order to generate (r_d, t_d) pairs such that the delayed option would be selected with a predictable probability, Turner et al. first used a staircasing task to assess a subject's discounting behavior. Following the results from the task and some modeling assumptions, Turner et al. generated an experimental set of trials such that the probability of picking the delayed option P_d was an independent variable in the task, consisting of the set $P_d = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. However, because both the behavioral and neural data were approximately symmetric about the indifference condition of $P_d = 0.5$, Turner et al. folded the data by recoding the variables to correspond to a "subjectively lower

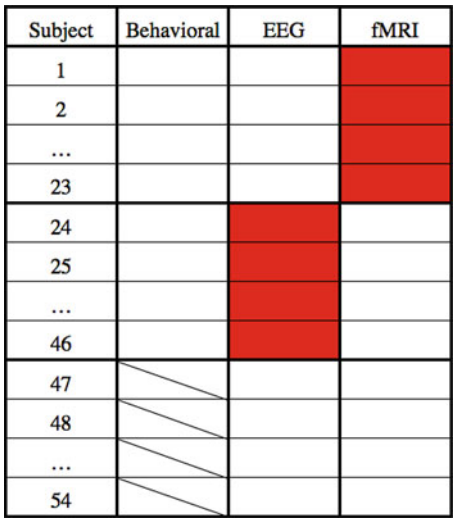


Fig. 4.5 Design matrix for the experimental data. The data consisted of behavior, EEG, and fMRI measures (columns) for different combinations of subjects (rows). The red cells represent observations that were not recorded, and stripped cells represent dual-purpose observations. For these cells, Turner et al. [46] collected behavioral measures during both fMRI and EEG sessions

valued" and a "subjectively higher valued" choice, akin to folding across response type to create "incorrect" and "correct" response choices, respectively. The new variables were labeled $V1$, which consisted of $P_d = \{0.1, 0.9\}$, $V2$, which consisted of $P_d = \{0.3, 0.7\}$, and $V3$, which consisted of $P_d = \{0.5\}$. In this way, the variables $V1$, $V2$, and $V3$ can be thought of as conditions in the experiment of increasing difficulty.

In total, three experiments were run: the first experiment consisted of only behavioral data and EEG measures (i.e., the first 23 rows of Fig. 4.5), the second experiment consisted of only behavior and fMRI measures (i.e., the second 23 rows of Fig. 4.5), and the last experiment was a repeated measures design where subjects provided both behavior and EEG measures in one session and behavior and fMRI measures in the second session. In Fig. 4.5, missing observations are illustrated with red cells, whereas data consisting of multiple measures are illustrated with stripped cells.

Figure 4.6 summarizes the data obtained from the experiment. The first row (i.e., Panels a and b) illustrate the behavioral data, the second row (i.e., Panels c and d) illustrate the EEG data, and the third row (i.e., Panels e and f) illustrate the fMRI data. Panel a shows the response time distributions for each value condition $V1$ (blue), $V2$ (gray), and $V3$ (red) collapsed across the response choice, whereas in Panel b, the response time distributions are separated on the basis of value condition (columns) and choice, where the response times for the subjectively lower valued option (i.e., the "incorrect" response) are shown on the left and the subjectively higher valued option (i.e., the "correct" response) are shown on the right. Panel c shows the mean stimulus-locked event-related potential (ERP) for the three value conditions: V1 (blue line), V2 (black line), and V3 (red line), whereas Panel b shows the corresponding topographic plot of the average EEG signal for the four selected electrodes during the chosen time window shown in Panel c as the dashed vertical

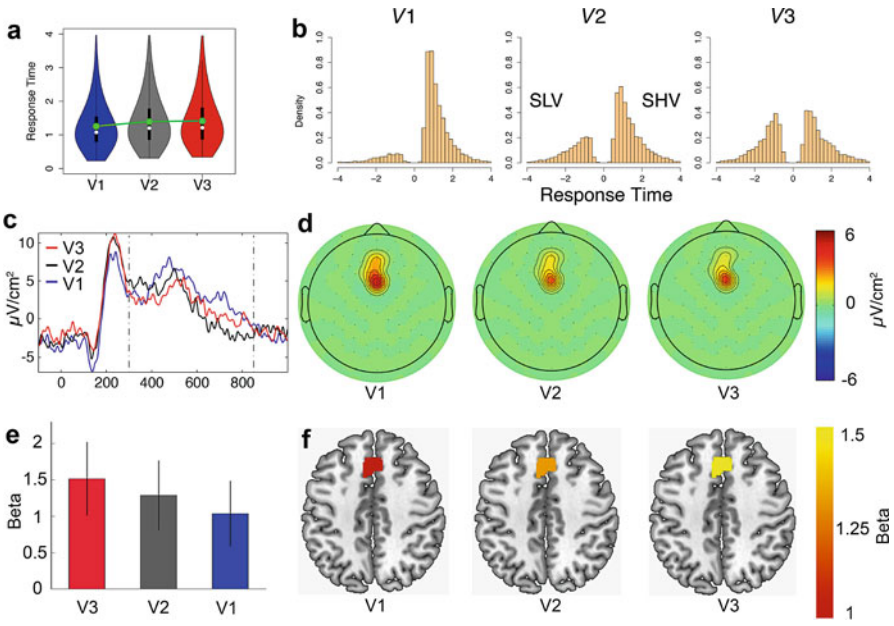


Fig. 4.6 Summary of the data from the experiment. (**a**) Violin plots of the response times collapsed across the response choice for each condition, with the green line/dots representing the mean, and the white dots representing the median. (**b**) The choice response time distributions for the three conditions $V1$ (first column), $V2$ (second column), and $V3$ (third column). The response times corresponding to the subjectively low-valued option (SLV) are shown on the negative x-axis, whereas the response times corresponding to the subjectively high-valued option (SHV) are shown on the positive x-axis. (**c**) Mean stimulus-locked event-related potential (ERP) for the three value conditions: V1 (blue line), V2 (black line), and V3 (red line). The time window used in the analysis was from 300 to 850 ms, which is indicated by the dashed vertical lines. (**d**) Topographic plots of the average EEG signal in the four selected electrodes during the time window shown in (**c**), for each of the three value conditions: V1 (left), V2 (middle), and V3 (right). These trial-by-trial EEG values were used in the modeling applications. (**e**) Average of the single-trial beta estimates for each value condition, based on a GLM analysis of dmFC activity during the decision period. (**f**) Similar information in Panel **e**, but illustrated in the dmFC, and separated across the three value conditions: V1 (left), V2 (middle), and V3 (right). These trial-by-trial β values were used in the modeling applications

lines. Panel e shows a barplot of the single-trial beta estimates separated by value condition, based on a GLM analysis of dmFC activity during the decision period. Panel f provides similar information as Panel e, but shows this information spatially (i.e., within the dmFC) and separated across the three value conditions: V1 (left), V2 (middle), and V3 (right).

To capture all aspects of the data simultaneously, Turner et al. [46] developed what they referred to as a "trivariate" model that enforced a joint constraint on the parameters of a behavioral submodel, EEG submodel, and an fMRI submodel. The trivariate model is illustrated in Fig. 4.7. As in the graphical diagram presented

in Fig. 2.1, different plates in the figure correspond to different modalities of information. Here, the orange plate represents the behavioral data/submodel, the green plate represents the EEG data/submodel, and the blue plate represents the fMRI data/submodel. For the behavioral submodel, Turner et al. assumed the response time distributions shown in Fig. 4.6b arose from the LBA model, where the drift rate parameter was assumed to vary across the levels of V [44, 138]. For the EEG submodel, Turner et al. assumed the degree of activation shown in Fig. 4.6d could be described via a normal distribution with a different mean parameter for each value condition. Similarly, for the fMRI submodel, Turner et al. assumed the degree of activation shown in Fig. 4.6f could also be described via a normal distribution with a different mean parameter for each value condition. To connect the submodels together (i.e., the Ω node in Fig. 4.7), Turner et al. assumed that the drift rate, mean EEG, and mean fMRI model parameters could be described through a multivariate normal distribution across subjects.

Figure 4.7 also illustrates that different types of models can be considered when explaining the manifest variables under consideration. For example, when we wish to explain how all three variables B, E, and F arise, we might create a model such as the trivariate model detailed in Turner et al. [46]. However, we can also examine the extent to which other, simpler models can influence the predictive power of specific effects. For example, if we wanted to know whether having only a single neural covariate such as the EEG data E could perform just as well as having both

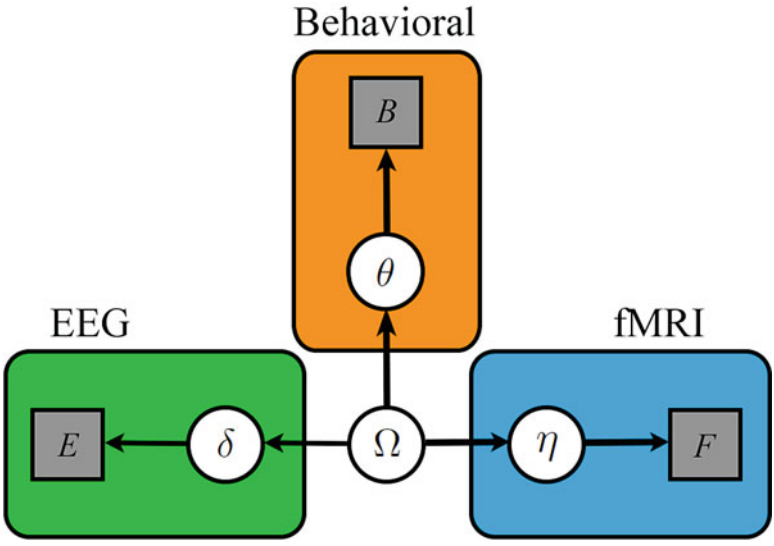


Fig. 4.7 Graphical diagram of a trivariate model from Turner et al. [46]. Observable data are represented as gray boxes, whereas unknown (latent) variables are represented as empty circles. The orange plate represents the behavioral data/model, the green plate represents the EEG data/model, and the blue plate represents the fMRI data/model

E and F, we could fit a "bivariate" model that captured the trends in B and E, but ignores F. Similarly, we could ask whether using the BOLD response F as the neural covariate performed better than say, the EEG covariate E. Considering these questions, Turner et al. tested a total of four models: one model that only included behavioral data (i.e., a behavioral model), one model that included both behavior and EEG data (i.e., a bivariate EEG model), one model that included both behavior and fMRI data (i.e., a bivariate fMRI model), and one model that included all three measures simultaneously (i.e., a trivariate model).

To examine the effects of the covariates, Turner et al. [46] first withheld the behavioral data from one randomly selected subject, Subject 53, and then fit all four models to the remaining data. Of course, as some models ignore some features of the full data set, they are bound to have some consequence on what the models predict for the behavioral effect that would be observed for Subject 53, had the behavioral data been fully observed. Figure 4.8 shows the resulting posterior predictive distributions for the main behavioral effect parameters for Subject 53: $\eta_{53}^{(1)}$ (left panel) and $\eta_{53}^{(2)}$ (right panel). In both panels, predictions from the behavioral-data-only, bivariate EEG, bivariate fMRI, and trivariate models are shown as orange, green, blue and red densities, respectively. Figure 4.8 shows that both $\eta_{53}^{(1)}$ and $\eta_{53}^{(2)}$ are centered on positive values, with $\eta_{53}^{(2)}$ being larger than $\eta_{53}^{(2)}$. Comparing across models, we see that while all models produce posteriors with similar expected values, the dispersion of those posteriors differs substantially. Namely, the behavioral-only model has the widest variance, the bivariate EEG and fMRI models have similar and slightly less variance, and the trivariate model has the smallest variance.

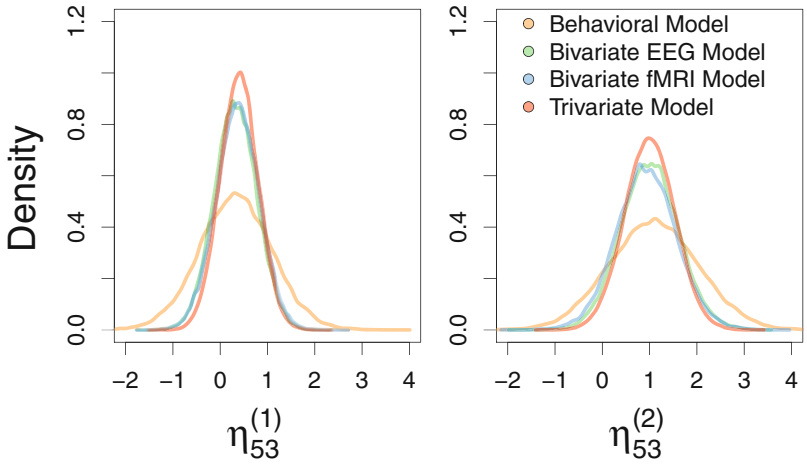


Fig. 4.8 Comparison of the value effect parameters $\eta_{53}^{(1)}$ (left panel), and $\eta_{53}^{(2)}$ (right panel) obtained from the behavioral-data-only model (orange), the bivariate EEG model (green), the bivariate fMRI model (blue), and the trivariate model (red) for the withheld subject (i.e., Subject 53)

The dispersion of the predicted posteriors relates directly to how well the models are constrained: with smaller variance, the models are said to have greater constraint in generating predictions for the unobserved effects illustrated in Fig. 4.8. While the trivariate model does exhibit greater constraint for the behavioral effects, Fig. 4.8 by itself does not necessarily imply that the additional constraint provided by the neural covariates is a beneficial in a predictive analysis. It could be that the best prediction is one that generates more variability, which might be the case if the neural data were not informative for the pattern of behavioral responses that might be observed for this individual.

One way to compare the predictive power of the four models is through cross validation tests, as we discussed above. Recall that for these data, eight subjects performed both EEG and fMRI versions of the task, and so we can use these subjects to ask whether the EEG, fMRI, or both aspects of the data enhance our predictions about the behavioral effects. To do this, we performed a leave-one-out cross-validation test. First, we removed all data for one subject (e.g., Subject 47). Second, we fit all four models to the remaining 53 subjects' data. Third, we generated predictions for the withheld subject's behavioral data, conditional on some aspect of the neural data. For example, the bivariate EEG model only takes EEG data as an input, whereas the trivariate model can take both EEG and fMRI data as inputs. Finally, we examined the models' predictive accuracy by evaluating the likelihood for the withheld data under each model, as demonstrated in Chap. 4. After normalizing the likelihoods for the four models, we arrived at an estimate of the probability that a particular subject's data was best captured by each model, a value that summed to one across models. Figure 4.9 shows a ternary plot of these probabilities for the eight subjects in our data. In the figure, we removed the behavioral-only model from consideration because it did not provide the best account of any subject's data in the experiment. The reference lines in Fig. 4.9 designate the areas best predicted by a particular model. In addition, the color values illustrate the graded nature of the model selection process. The trivariate model is represented as blue, the bivariate EEG model is represented as red, and the bivariate fMRI model is represented as green. The nodes in the graph mark the probability of a given model, conditional on the behavioral data. The figure shows that for each of the subjects in our data, the trivariate model provided better predictions than any of the other three models.

While other joint modeling applications focus on formalizing the connection between a single modality of neural data and the behavioral data, Turner et al. [46] focused on extending this framework to multiple modalities of neural data, namely fMRI and EEG. Through a variety of different comparative analyses at a generative and predictive level, Turner et al. ultimately concluded that having both EEG and fMRI measure improved the joints model's ability to both fit and predict (out of sample) data.

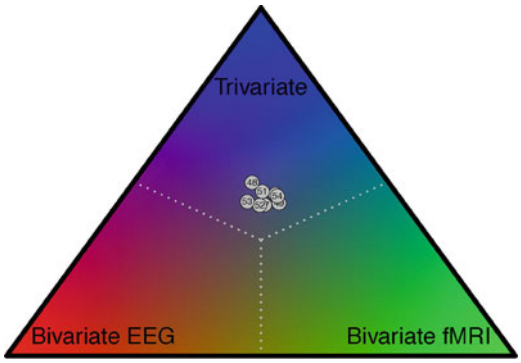


Fig. 4.9 Results of the cross-validation test on the behavioral data from each of the last eight subjects of the experiment (gray nodes). The three vertices of the ternary plot each designate areas favoring one type of joint model: bivariate EEG (red; bottom left), bivariate fMRI (green; bottom right), and the trivariate model (blue; top). Reference lines are plotted to reflect points of indifference between the models

4.4 Incorporating Single Cell Neural Data into Joint Models

A number of neuro-computational models have been developed to link single cell neural recordings and perceptual decision-making [97, 139, 140]. Neurons in the frontal eye field (FEF) and lateral intraparietal (LIP) areas in macaque monkeys increase their firing rate to some threshold value and the time it takes to reach the threshold relates to the decision time in perceptual decision-making tasks [139, 141–144]. These findings suggest that the increase in firing rates in these neurons can be compared to the process of evidence accumulation in stochastic accumulator models.

In one modeling approach, Purcell et al. [97] developed an integrative account that tightly constrained the link between the neural and behavioral data. They mapped the firing rate of frontal eye field (FEF) visual neurons onto perceptual evidence and the firing rate of FEF movement neurons onto evidence accumulation in accumulator models. The innovation in their approach is that they used the combined neuron activity from a number of visual FEF neurons as direct input to an accumulator model. Therefore, the accumulator model did not contain a latent process for evidence accumulation as this process was directly constrained by the neural data. This direct input approach has been used to test between different types of accumulator models. By comparing the dynamics of the accumulator models driven by the visual neurons to the measured trajectories of the movement neurons (not used during model development) it was possible to select among different decision-making mechanisms in accumulator models [97, 139, 140].

Recently, a joint model for single cell recording data and perceptual decision making was introduced by Cassey et al. [83]. They modeled data from a seminal experiment by Roitman and Shadlen [142] containing behavioral recordings of two monkeys in a simple decision-making task. The neural data consisted of single-cell neural recordings from the lateral intraparietal area of the cortex. On each trial, a random dot kinematogram appeared on the screen and the monkey indicated whether the coherently moving dots were drifting left or right. Response times and choices were recorded from each trial as well as the timing of action potentials from a set of neurons in the lateral intraparietal area of the cortex. The joint model builds on the work by Purcell et al. [97, 139] by assuming that an evidence accumulation model can provide a tight link between the observed neural firing rate and behavioral responses. In contrast to Purcell et al. [97, 139] where the neural data is used directly as input to an evidence accumulation model, the model also includes an explicit generative model of the neural data. By including a generative model of the neural data, a complete joint model for the neural and behavioral data is specified. This offers a number of advantages. First, it allows the model to make predictions about behavior conditional on partially observed neural data. For example, the model can make predictions when a response is likely to occur on the basis of a limited observable time window of spike trains (e.g., the first 100 msec of neural data). These prediction are possible because the generative neural model can make inferences about missing neural data outside of the observable time window. Second, the model is also able to make inferences about the neural data when only behavioral data are observed. Therefore, the model is able to make predictions about missing data based on a variety of scenarios. Finally, the generative model for neural data can also include a number of assumptions about neuron and monkey differences. The joint model by Cassey et al. [83] included a hierarchical version where neural model parameters could vary between neurons as well as monkeys.

The overall approach is illustrated in Fig. 4.10. The core element of the model is an accumulator model of decision-making based on the LBA model as described in previous chapters. The main assumption in the model is that the neural data can be described by a time-inhomogenous Poisson process where the changes in firing rates of an individual neuron follow a sequence of events. Before the stimulus is presented, the firing rate is assumed to be constant at a baseline rate. After stimulus presentation, the firing rate dips followed by a recovery. The spike rate then ramps up linearly until a threshold is reached after which the firing rate decreases to a post-decision baseline rate. The piece-wise linear function of these changes are described by a number of parameters that relate to the timing of the different phases as well as the firing rates reached at the different phases. For example, the extension of the LBA model makes a distinction between two different components of non-decision time that are difficult to disentangle on the basis of behavioral data alone: the stimulus encoding time and motor response time. The key assumption in the model is that the linear ramping-up phase represents the decision-making period. The time it takes for this process to reach a threshold determines the response time. This part corresponds to the standard LBA model.

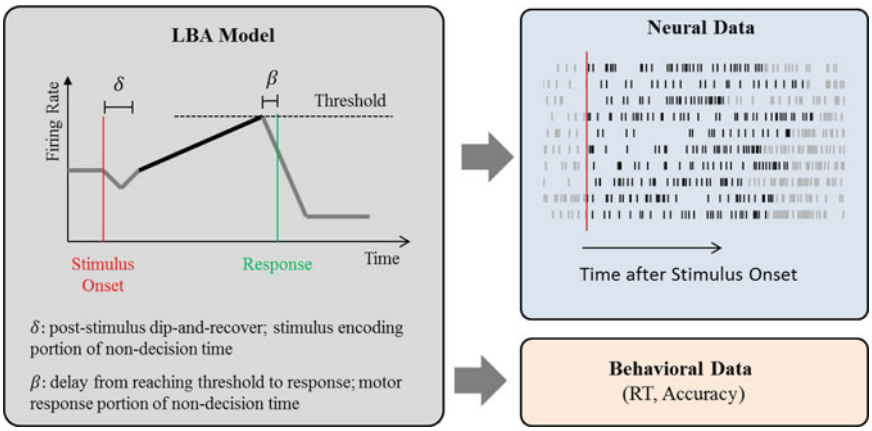


Fig. 4.10 Illustration of the modeling approach by Cassey et al. [83]. The LBA model provides the linking hypothesis between the neural data and the behavioral data. The linking function consists of a piece-wise linear function that describes the changes in the neural firing rate of a time-inhomogenous Poisson process. The sequence of changes progresses through a post-stimulus dip-and-recovery followed by a linear ramping up of firing rate to a threshold and ending with a decrease to a lower baseline after a decision is made. The evidence accumulation in the LBA model occurs during the ramping phase as indicated by the darker line segment. Two parameters, δ and β, are illustrated that relate to the components of non-decision time: stimulus encoding time (δ) and motor-response time (β). The neural data on the right shows a sample of spike trains for an individual neuron as predicted from a distribution over piece-wise linear firing rate functions that vary in the ramping rate. Darker lines show the spikes during the period from stimulus onset to response. The behavioral data consists of the response time (time of saccade) and accuracy as predicted from the LBA model

The joint model was able to fit the full distribution of responses times across different levels of decision difficulty. Simultaneously, the model fit the changes in firing rate of decision-related neurons across time during each decision trial. In addition, two models were compared: a hierarchical model that allowed for neuron differences and a non-hierarchical model. The results showed that the hierarchical model was better able not only to make predictions about missing neural data but also about missing behavioral data (e.g., when a response would be imminent based on partially observed neural data). This demonstrates the importance of incorporating individual differences in the generative neural model. Finally, the joint model was also able to separate between the non-decision time related to stimulus encoding and response preparation which is usually difficult to do on the basis of behavioral data alone. This provides more evidence that joint models can use neural data to provide more constraint on latent variables in behavioral models.

Chapter 5
Future Directions

5.1 Revisiting the Linking Assumption: Structural Equation Models

One important issue in building connections between brain and behavior is reconciling the mismatch in dimensionality of the data structures. For cognitive models, typically only a handful of parameters are used to detail the distribution of behavioral data, whereas in neuroscience one has access to a plethora of neurophysiological measures with both temporal and spatial precision. Due simply to the dimensionality of the neural measures, navigating the number of possibilities for mapping the neural measures to mechanisms in a cognitive model is an extremely difficult task, mostly because there does not seem to be any automatization for this process.

Perhaps the most concrete difficulty we face in assigning a mechanistic interpretation of the neural measures lies in the complexity of inferring the linking function between brain and behavior. In several of our previous applications, we have found that a multivariate normal linking function works well in characterizing the nature of how neural and behavioral measures are related for two reasons. First, the multivariate normal provides a convenient measure for the central tendency of each model parameter, allowing for straightforward analysis of the hierarchical effects. Second, the multivariate normal provides a measure of the relationship between a cognitive model parameter and each neural measure in the form of a pairwise correlation parameter. Despite these conveniences, considering every possible pairwise relationship between the neural features and behavioral model parameters creates a highly complex model. While highly complex models are not necessarily a bad thing, the problem is that they will likely be *too* complex for the data, given the extreme overlap that occurs between neural measures due to spatial or temporal dependencies that may be difficult to articulate well in our neural submodel.

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_5

5.1.1 Complexity

To see how complexity grows with increases in the number of neural features, consider the following analysis. First, let q represent the number of neural measures or "features" in the data (e.g., the number of regions of interest or the number of electrodes), and let k stand for the number of behavioral model parameters we wish to relate to the neural data. If we assume a multivariate normal linking function, the dimensionality of the variance-covariance matrix Σ will be $(p \times p)$, where $p = q + k$. If we impose the most exploratory joint model possible, the correlation between every behavioral model parameter and every neural feature will be estimated, and so the number of free parameters n_{MVN} within Σ is

$$\begin{aligned} n_{MVN} &= \frac{p(1+p)}{2} \\ &= \frac{(q+k)(1+q+k)}{2} \\ &= \frac{q^2+q+2kq+k+k^2}{2}. \end{aligned} \tag{5.1}$$

Here, Equation 5.1 shows that a squared q and k term exists in Equation 5.1, meaning that increasing the number of neural features or the number of behavioral model parameters has a quadratic effect on the complexity Σ. As a consequence, the types of problems in which joint models can be applied easily are severely limited by the complexity of the inference process. In other words, exploratory joint models may only be applied when both the number of model parameters k and the number of neural features q are small in number.

To explore more parsimonious options, Turner et al. [68] looked to techniques prevalent in structural equation modeling. They proposed a linking function based on factor analysis that decomposes the variance-covariance matrix Σ into three matrices, according to the following equation, which is also illustrated in Fig. 5.1:

$$\Sigma = \Lambda \Phi \Lambda^T + \Psi. \tag{5.2}$$

First, Λ contains "factor loadings" that measure the degree to which a neural feature "loads" onto a latent model parameter. As an example, the element in the pth row and kth column represents the degree to which the pth element is associated with the kth factor. Higher loadings represent a closer association to the latent factor, whereas loadings near zero represent low associations to that particular factor. In a typical application, the number of factors k is fewer than the number of input variables p, allowing for a less complex representation of Σ. Here, Turner et al. [68] specified that each meaningful mechanism in the model should correspond to a single latent factor, and so the number of model factors was set equal to the number of behavioral model parameters k. The matrix Λ places the neural features on the rows and the latent model factors on the columns, so it has size $(p \times k)$. However, after some

theoretically-motivated constraints, the number of estimable parameters within Λ was qk. Second, the matrix Φ contains what are called "factor variances" that describe the pattern of variability among the factor loadings. The intuition behind Φ is that the factors themselves have variability in their representations of the data, and there may be relationships between the factors that partially explain the pattern of factor loadings in Λ. While the size of Φ is $(k \times k)$, following conventional constraints on the factor variance matrix, the number of estimable parameters in Φ was equal to the number of model factors k. Third, the residual variance matrix Ψ captures patterns in the variance-covariance matrix that are not attributable to patterns of factor loadings. By default, the matrix Ψ has size $(p \times p)$, but Turner et al. applied two constraints to reduce the number of parameters within Ψ to be $p - k = q$. Taking all of these constraints together, the number of free parameters n_{FA} in the factor analysis linking function are

$$\begin{aligned} n_{FA} &= qk + k + q \\ &= q(k+1) + k. \end{aligned} \tag{5.3}$$

Equation 5.3 shows that only linear terms for q and k appear, meaning that the complexity of Σ is only linearly affected by changes in either the number of neural features or number of behavioral model parameters. To illustrate the magnitude of these effects, consider the following example. Suppose our behavioral model has $k = 3$ parameters that we wish to relate $q = 24$ neural features of interest. In this scenario, the number of parameters to be estimated within Σ for the multivariate normal linking function is 378, whereas for the factor analysis linking function, the number of parameters is only 99. While both of these linking functions are reasonable as far as the computational cost is concerned, suppose the number of neural features grows to $q = 128$, as it might if we were considering data from an EEG experiment. In this scenario, the multivariate normal function would have 8,646 parameters to be estimated within Σ, whereas the factor analysis linking function would have 515 parameters. In our experience, the multivariate normal linking function would be computationally too costly to estimate effectively, whereas the factor analysis linking function would be manageable.

Figure 5.1 provides an illustration of the factor analysis linking function (see Equation 5.2) and the particular constraints applied in Turner et al. [68]. The color of each element in the matrices represents that element's value. When a constraint is applied to a given matrix, the elements are color coded to be either white – representing a constraint of zero – or black – representing a constraint of one. The variance-covariance matrix Σ is shown on the far left-hand side, and its decomposition is shown on the right-hand side. Figure 5.1 shows that while the three matrices Λ, Φ, and Ψ are somewhat sparse, together they can still produce a rich pattern of parameter covariation as expressed in Σ.

At first glance, decreasing the complexity of Σ may seem like an advantageous solution. However, decreasing the complexity must also come with a cost of a loss of explanatory power, and the decrease in complexity may prohibit a joint

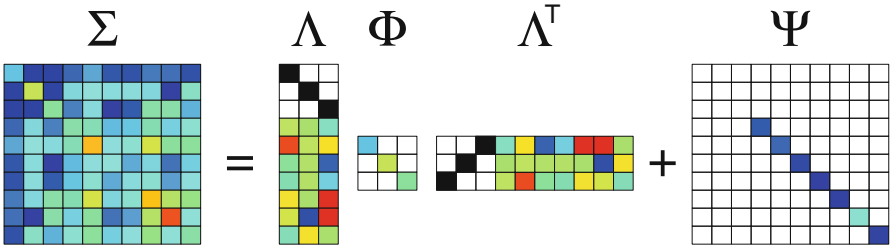


Fig. 5.1 Factor analysis linking function investigated in Turner et al. [68]. The color of each element in the matrices represents that element's value. When a constraint is applied to a given matrix, the elements are color coded to be either white – representing a constraint of zero – or black – representing a constraint of one. The intuition behind this equation is to decompose the variance-covariance matrix Σ into a factor loading matrix Λ, a factor variance matrix Φ, and a residual matrix Ψ (see text for details). Even though the three matrices Λ, Φ, and Ψ are somewhat sparse, together they can still produce a rich pattern of parameter covariation as expressed in Σ

model from generalizing to new data well. Given these concerns, Turner et al. [68] performed two simulation studies, one investigating the generative properties of the factor analysis and multivariate normal linking functions, and one investigating their predictive properties. For the generative analysis, Turner et al. first generated hypothetical data from a joint model assuming a multivariate normal linking function – a model we will refer to as the MVN model henceforth – and then fit both a MVN model and a joint model assuming a factor analytic linking function – a model we will refer to as the FA model henceforth. The goal was to examine whether or not the FA model could accurately recover the variance-covariance matrix Σ from the more complex model (i.e., the MVN model). They showed that both the MVN and FA models were able to accurately recover Σ. In a secondary analysis, they also showed that when the data were generated from a FA model, both the MVN and FA models were able to accurately recover Σ as well. However, this secondary result is perhaps unsurprising, as the FA model was the true data-generating model and the MVN model was more complex, so both models should have been able to recover Σ properly.

Perhaps the more powerful illustration of the utility of the factor analysis linking function was in the predictive analysis presented in Turner et al. [68]. The goal in the predictive analysis was to investigate whether or not the FA model, being drastically less complex relative to the MVN model, could predict out-of-sample data more accurately than the MVN model. To test this idea, Turner et al. first generated data from the FA model, and then separated it into a training and test set. They then fit both the FA model and the MVN model to the training data, and used the resulting parameter estimates to generate predictions for the test data. In this first analysis, they showed that the FA model was able to predict the test data more accurately than the MVN model. This result might be expected given that the FA model was the true data generating model, and there are well known tradeoffs between the complexity of a model and its ability to generalize to new data [74–76]. Specifically,

as a general rule, the more complex a model becomes, the poorer its performance in generalization tests. As such, given that the MVN model is already more complex and it was not the true data generating model, it might be expected that the FA model was able to outperform the MVN model. As a follow up, Turner et al. repeated the analysis but this they generated the training/test data from the more complex MVN model. After fitting both the FA and MVN models to the training data, they again used the resulting parameter estimates to generate predictions for the test data. In their evaluation, Turner et al. showed that the FA model was able to perform at least as well as the MVN model, suggesting that not only could the FA model capture the rich structure within the variance-covariance matrix used by MVN model, but that the FA model could extract the most diagnostic features in the data so as to produce more accurate predictions for new data.

5.1.2 Real-World Application

To explore the utility of the factor analysis linking function, Turner et al. [68] fit the FA model to data reported in Van Maanen et al. [11], which collected response choice, response times, and pre-stimulus BOLD responses for every trial. In addition to these measures, a speed-accuracy manipulation was used where subjects were asked to respond as quickly as possible on some trials, or as accurately as possible on others. These conditions were interleaved in the experiment, where a condition cue was used to provide the context for decision making. The experiment used a moving dots task where subjects were asked to decide whether a cloud of semi-randomly moving dots appeared to move to the left or to the right.

Traditionally, speed-accuracy experimental manipulations have been modeled by assuming a change in only the threshold parameter across the two experimental contexts, such that the threshold parameter should be larger when accuracy is emphasized to facilitate a longer stimulus integration period [145]. However, a recent evaluation of this traditional formulation found that sometimes speed emphasis instructions can also influence the drift rate parameter. In the application presented in Turner et al. [68], they examined whether the brain-behavior connections also varied as a function of task instruction.

A joint model similar to the one presented in Turner et al. [18] and discussed in Chap. 4 was used to fit these data, but the linking function was modified to be a factor analysis linking function instead of a multivariate normal one. To examine the effects of the speed emphasis task instructions, Turner et al. [68] first separated the data into the "speed" and "accuracy" conditions. The model was fit to each of these conditions separately so that we could compare and contrast the brain-behavior relationships that existed across the two task conditions. The factors in the model corresponded to the mechanisms of the DDM: drift rate, starting point, and nondecision time. The model structure assumes that model mechanisms like threshold and mean drift rate are the same across all subjects, but still allows for fluctuations in the model parameters from one trial to the next. Importantly,

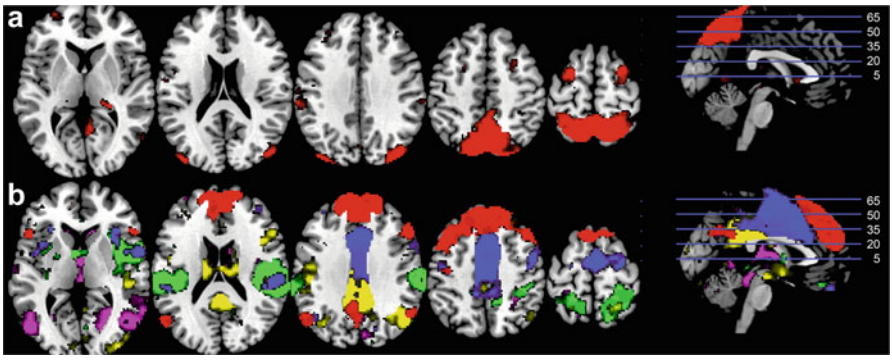


Fig. 5.2 Brain regions most associated with the drift rate factor in the joint model presented in Turner et al. [68]. Panel **a** shows the brain areas with the most prominent factor loading onto the drift rate factor under speed emphasis, whereas Panel **b** shows the brain areas loading onto the drift rate factor under accuracy emphasis. For illustrative purposes, we chose a factor loading threshold of 0.6 to determine which brain areas to present

the model attributes across-trial variability to the factor structure defined by the mechanisms used in the cognitive model.

To gain a sense of how the mechanisms in the model are related to brain activation across the two conditions, Fig. 5.2 shows the brain regions with the most prominent factor loadings for drift rate factor from the joint model. For illustration purposes, we first chose a threshold of 0.6 for all factor loadings. Figure 5.2 separates the factor loading structure according to task such that Fig. 5.2a corresponds to the speed emphasis and Fig. 5.2b corresponds to the accuracy emphasis. In each row, five axial slices are shown on the left that correspond to the slices illustrated in the sagittal slice on the right. Figure 5.3 shows a similar plot as Fig. 5.2, but here the brain areas shown correspond to the ROIs that loaded most heavily onto the starting point factor in the joint model. Again, the factor structure is separated into rows such that the factor structure corresponding to speed instruction is shown in Fig. 5.3a, whereas the factor structure corresponding to the accuracy emphasis is shown in Fig. 5.3b.

Although Turner et al. [68] provided a more detailed comparison of the factor structure present across the two task conditions, here we can make some broad brushstrokes regarding the engagement of frontal and parietal lobes. For example, in the accuracy condition (i.e., the bottom row of both Figs. 5.2 and 5.3), the frontal lobe appears to be more engaged in the processing of the stimulus information (i.e., the drift rate factor in Fig. 5.2) whereas the parietal lobe is more engaged in the placement of the starting point (i.e., the starting point factor in Fig. 5.3). However, when transitioning to a speeded decision making regime, the frontal lobe can be mechanistically described as processing information that is less stimulus dependent as the frontal network most heavily loads onto the starting point factor (i.e., see Fig. 5.3a). By contrast, under speed emphasis instruction, the parietal lobe loads more heavily onto the drift rate factor. These preliminary results suggest

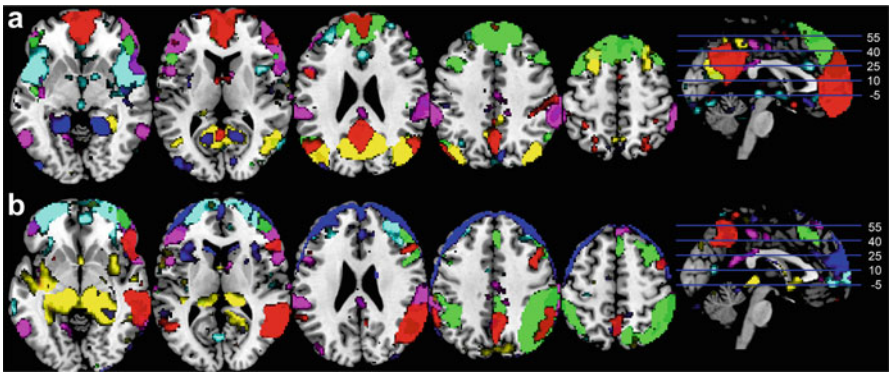


Fig. 5.3 Brain regions most associated with the starting point factor in the joint model presented in Turner et al. [68]. Panel **a** shows the brain areas with the most prominent factor loading onto the starting point factor under speed emphasis, whereas Panel **b** shows the brain areas loading onto the starting point factor under accuracy emphasis. For illustrative purposes, we chose a factor loading threshold of 0.6 to determine which brain areas to present

that elements of a brain network can be used differently to accomplish the same task when different instructions are emphasized. However, more detailed analysis involving more subjects and different types of task manipulations will be required to provide stronger evidence of a differential brain engagement across task contexts.

5.2 Linking Structure to Function

The human brain may be mapped at three different levels: the macroscopic, the mesoscopic, and the microscopic levels [146]. The macroscopic level is measured in centimeters, the scale of large neural networks that spread across the whole brain. The mesoscopic level is measured in millimeters, which is the scale of individual cortical and subcortical nodes, and the microscopic level is measured in micro- and nanometers, which is the scale of individual neurons and neurotransmitters.

In this section, we highlight a recent approach that links brain structure to brain function within a model-based cognitive neuroscience framework on the mesoscopic and macroscopic level. This is done by using state-of-the art ultra-high field (UHF) 7Tesla (T) magnetic resonance imaging (MRI), diffusion weighted imaging (DWI), and network analyses. These cognitive neuroscience techniques allow us to capture psychological processes such as decision-making with multiple-choice alternatives.

5.2.1 Mapping Structure and Function on Different Levels

Decision-making has attracted much attention in the empirical and neurocomputational neurosciences [8, 86, 147]. However, many studies focus on two-alternative forced-choice decision-making tasks, limiting the ecological validity of these studies [148, 149]. In a recent study, we employed a multi-alternative decision making task using a random dot motion (RDM) paradigm with the question of whether an increase in choice options prolonged the accumulation of evidence [148, 150]. To incorporate such results in the theoretical framework of decision-making, quantitative mathematical and neurocomputational models have been developed that make predictions about the latent psychological processes and brain structures involved in multiple-choice decision-making [151]. Importantly, several of these models focus on information processing in the basal ganglia (BG), and describe the computations of its different structures, including the subthalamic nucleus (STN). The STN is thought to play an important role in action selection due to its unique anatomical position and is thought to contribute to inhibiting multiple motor plans [152, 153].

Importantly, several neurocomputational theories make concrete predictions on the mesoscale in that STN activity during decision-making should increase with the number of choice alternatives. With more alternatives competing for selection, more STN neurons will be selective for these alternatives, inhibiting other options; there will be higher conflict and a prolonged need for movement inhibition until the conflict is resolved; and there will be a need for more extensive normalization of action probabilities.

Here, by simulating predictions from the multihypothesis sequential probability ratio test (MSPRT), a neurocomputational model of action selection, we generated predictions of how the activity in the STN and the observed behavior respond to the number of choice alternatives in a multiple-choice RDM task. These MSPRT predictions were then tested by acquiring 7 Tesla (T) functional Magnetic Resonance Imaging (MRI) data while 15 participants complete a multiple-choice decision-making RDM task with three, five or seven alternatives (see Fig. 5.4).

The information processing in the BG is thought to be modulated by areas such as the pre-supplementary motor area (pre-SMA) and the anterior cingulate cortex (ACC). These two structures are known to be involved in perceptual decision-making [154] and are anatomically connected to the STN [155]. Previous work has shown that the pre-SMA modulates the response threshold [86], while the ACC is implicated in switching between response regimes [11]. Taking these anatomical and functional findings into account, these two areas may have an important modulatory role on the STN activation in multiple-choice decision-making.

The present study also addressed a question of the directionality of the connection between cortical and subcortical areas. More specifically, we tested the functional and effective connectivity between the pre-SMA, ACC, and STN during decision-making using a novel multi-modal combination of 7T structural and functional MRI, Diffusion Weighted Imaging (DWI) data, and ancestral graph (AG)

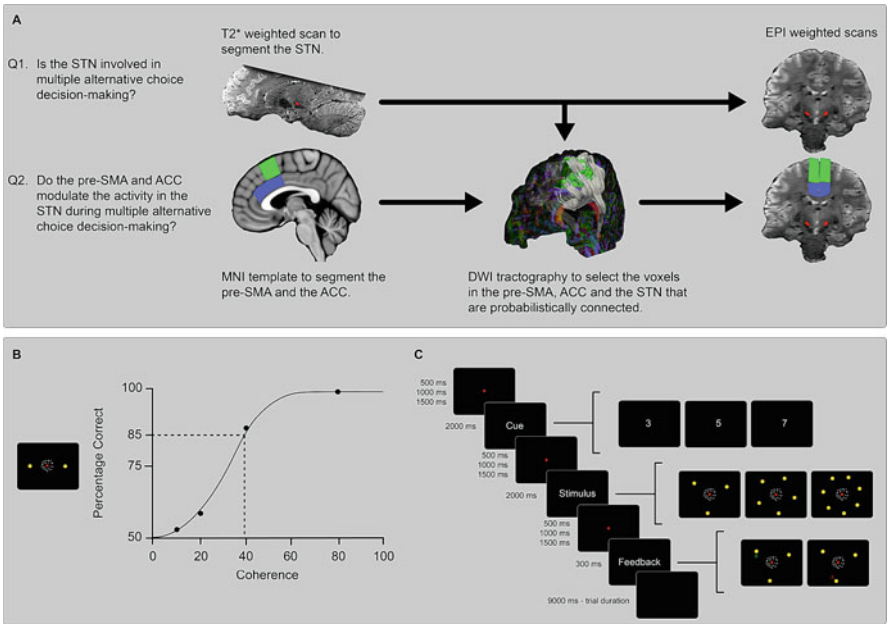


Fig. 5.4 Potential analysis pipeline and methods for linking structure to function. (**a**) Using individually segmented STN masks as a ROI in a standard fMRI analysis the question is addressed whether the STN is involved in multiple-alternative choice decision-making. To answer the question whether the pre-SMA or ACC modulate the activity in the STN the BOLD signal is extracted per trial in the three ROIs. Using DWI tractography only the voxels that are probabilistically connected between the pre-SMA or ACC and STN are selected. (**b**) The level of coherently moving dots was determined per individual. This was done by interpolating from the psychometric curve produced by the proportional-rate diffusion and selecting the coherency level that was associated with 85% accuracy. (**c**) Each trial lasted for nine seconds. The duration of the fixation dot was jittered and varied between 500, 1000 and 1500 ms. The cue was always valid and the locations of the targets did not vary within or between subjects. A response was considered correct if the final location of the joystick was within 20 degree of the correct target. If the response was outside of this window the response was considered incorrect. The participants were aware of this response window and were instructed to respond as fast and as accurate as possible. Seven percent of the trials were neutral trials in which a fixation dot was displayed for 9 s. (Taken with permission from Keuken et al. [150], Elsevier)

modeling ([156]; see Fig. 5.4). AG is a type of graphical modeling that enables the testing of the presence and nature of functional and effective connections between nodes or regions of interest (ROI).

The results showed that by means of the excellent spatial resolution and sensitivity provided by ultra-high field MRI and fMRI, the MSPRT predictions regarding the involvement of the STN in multiple-alternative decision-making were confirmed. When the number of alternatives increases task difficulty, the STN becomes more active. The results support the view that the STN functions as a brake mechanism, thereby facilitating the accumulation of more evidence for the

choice at hand. Additionally, using a novel combination of DWI, fMRI, and AG modeling, it was shown that the ACC correlates with, but no evidence was found that it directly modifies the activity in the STN during multiple-choice decision-making. In sum, this study provides a first step in linking brain information on a meso- and macroscopic level within a model-based cognitive neuroscience framework.

Chapter 6
Other Approaches

As we discussed in Chap. 1, our understanding of cognition has largely been advanced by two nearly non-overlapping and non-interacting groups. The first group, mathematical psychologists, is strongly motived by theoretical accounts of cognitive processes, and instantiates these theories by developing formal models of cognition. The models often assume a system of computations and mathematical equations intended to characterize a process that might actually take place in the brain. To formally test their theory, mathematical psychologists rely on their model's ability to fit behavioral data. A good fit is thought to reflect an accurate theory, whereas a bad fit would refute it [71]. By virtue of their abstraction, these analyses can tell us *why* we observe systematic patterns in behavioral data through the mechanisms of the cognitive model, but they say nothing about the location of the mechanisms' neural basis. The second group, cognitive neuroscientists, rely on statistical models to understand patterns of neural activity, often without any attempt to make a connection to the computations that might underlie some hypothesized mechanism. For example, some statistical approaches (e.g., multivariate pattern analysis) explicitly condition on the neural data to determine which aspects of the data produce better predictions for behavioral outcomes. Such an analysis can tell us *which* brain regions are predictive of a particular behavior and even *by how much*, but they say nothing about neither *how* nor *why* particular brain regions produce said behavior.

While both of these groups are concerned with explaining behavior, their approaches are quite different. One approach relies on a system of mechanisms constructed to mimic an idealized version of what could be happening in the mind. This approach has the benefit of forcing a researcher into explicitly articulating their theory by making assumptions that give rise to a cognitive model, but has the disadvantage of being inherently abstract. The other approach relies almost purely on statistical analyses of behavioral outcomes. Because there are so many explanations for how behavioral data might manifest, the interpretation of patterns

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_6

of brain activation is complicated. Most importantly, these interpretations lack mechanism and theoretical motivation. One consequence of interpretation without mechanism is a general inability to aggregate across experimental paradigms. There are of course many other limitations, but we refer the reader to [16] and [57] for more elaborated discussions.

Due to the limitations of considering one level of analysis, many have turned toward combining at least two levels of analysis to further our understanding of cognition. This emerging field is known as model-based cognitive neuroscience, as it infuses the powerful frameworks of mathematical models with the resolution of neurophysiological measures. To combine both approaches, we must perform some type of link between the two levels of analysis. However, there are many ways to perform this link, and this chapter describes several unique strategies for linking brain and behavioral data.

At this point, the number of linking approaches is large, and their degree of information flow between brain and behavior fill the entire spectrum of possibilities. There are many ways we could organize the approaches. One way is to discuss the "tightness" of the link between the two measures [157]. However, we follow the "goal-based" organization in Turner et al. [57], where they found it convenient to separate the various approaches on the basis of the goals of the researcher. In their terminology, there are at least three different goals a researcher might have in mind when forming a link. Within each goal, a set of "approaches" are used to illustrate specific types of strategies that have been used in the literature. First, a researcher might want to use the neural data as a way to constrain or formalize a behavioral model. There are several ways in which the neural data can constrain modeling choices, and we will discuss three such approaches in the Sect. 6.1. Second, a researcher might use a behavioral model as a way to interpret or predict neural data. Behavioral models assume a set of mechanisms that theoretically mimic a cognitive process of interest, making them an interesting way to impose theory in data analyses. In Sect. 6.2, we discuss two such approaches for accomplishing this goal. Each of these goals produce an end result that leans toward one level of analysis. The first goal is directed toward the development and evaluation of a behavioral model, whereas the second goal is directed toward finding where a mechanism is executed at a lower level than it is instantiated. As such, the first goal tends toward the algorithmic or computational level of analysis, whereas the second goal tends toward the implementation level. The third and final goal treats both levels of analysis as equally important by simultaneously considering both brain and behavioral measures into a *single* cognitive model. With the proper model in place, one can simultaneously achieve constraint on the behavioral model while retaining the ability to interpret the neural data. This third goal has been the focus of the previous chapters, and so we do not discuss this goal further here.[1]

[1]Note that we do not necessarily think this is a comprehensive list; in fact, we suspect that there is room for further development, and possibly the creation of entirely new analytic approaches.

6.1 Neural Data Constrain Behavioral Model

We begin our discussion by considering the goal of using neural data to constrain the development or inference of a behavioral model. In this endeavor, the neural data are considered important, but only in the sense that they inform the mechanisms within the behavioral model. We have identified three specific approaches consistent with this goal, illustrated in Fig. 6.1: the Theoretical Approach, the Two-stage Behavioral Approach, and the Direct Input Approach. As in Chap. 1, Fig. 6.1 represents the specific approaches as graphical diagrams where observable measures (i.e., data) are depicted as shaded square nodes, latent model parameters are depicted as empty circles, and arrows depict dependencies. Here, we denote the neural data N, and the behavioral data B. The neural data N could be neurophysiological recordings, functional magnetic resonance imaging (fMRI), electroencephalography (EEG), or other physiological measures. The behavioral data B could be response probabilities, response times, confidence ratings, or other typical behavioral data collected in a cognitive experiment. We now discuss each of these in turn.

6.1.1 Theoretical Approach

In the Theoretical approach, psychological theories are developed on the basis of considerations from both neuroscience and behavioral data. The top-left panel of Fig. 6.1 illustrates the Theoretical approach as statistically independent models of the neural and behavioral data because the link between these measures is established only through the researcher themselves (i.e., represented by the dashed arrow). In this approach, the dominant procedure uses neural measures to inspire the development of psychological models. First, the researcher observes particular

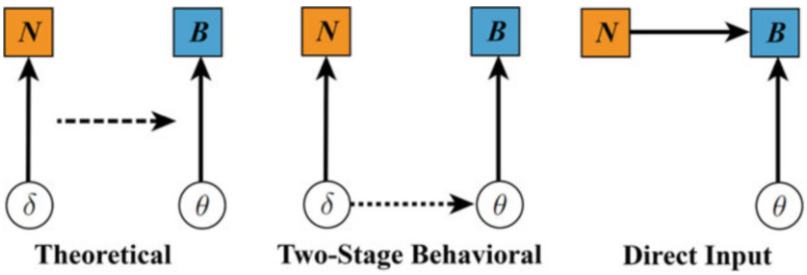


Fig. 6.1 An illustration of approaches for linking neural and behavioral data when the goal is to use the neural data to constrain the behavioral model. N represents the neural data, B represents the behavioral data, and θ and δ represent model parameters. Dashed lines indicate conceptual constraints (e.g., the Theoretical Approach), whereas solid lines indicate statistical constraints. Squares represent observable measures of cognitive processes, whereas the circles represent (latent) model parameters

aspects of brain function, such as information about the structure (e.g., individual neurons or densely connected brain regions) or function (e.g., dorsal and ventral pathways of visual stimulus processing) of the brain. Next, the researcher develops a model of behavior that, at its core, abides by these neural observations. With an initial model structure imposed by N, the researcher is now able to evaluate the relative merits of nested theoretical assumptions, and make incremental adjustments in the model to provide better fits to behavioral data B. Unlike other approaches discussed in this article, the Theoretical approach may draw inspiration from physiological or anatomical observations, but there is no mathematical or statistical link between the neural data N and either the model architecture or the model parameters that predict the behavioral data B.

Although the absence of an explicit link between neural and behavioral data may seem craven, the Theoretical approach has proven to be a powerful framework for motivating psychological theory. Perhaps the most prominent example of a Theoretical approach is the enormous class of neural network models. Neural network models have a long history, with one classic example being Rosenblatt's Perceptron machine [158]. In the development of the Perceptron, Rosenblatt made choices in his model that reflected operations observed in individual neurons, such as that the firing of individual neurons should be discrete [159]. Although these original neural network models were heavily criticized [160], pioneering work allowing for continuous activations in neuron-like units [161–165] evolved neural network models into more complex and successful theoretical approaches such as the parallel distributed processing (PDP) [166] models. Superficially, these models allow for the presence of individual nodes embedded within layers of a network, and these nodes are massively interconnected across layers, resembling neural structures in the brain. Through a process known as back-propagation, PDP models can be trained on behavioral data to learn important aspects of the decision rule, facilitating further systematic explorations of representation, learning, and selective influence (i.e., by a process referred to as "lesioning").

As another example, consider the Leaky Competing Accumulator [38] model (see Box 1 in Chap. 1). The LCA model was proposed as a neurally plausible model for choice response time in a k-alternative task. The model possesses mechanisms that extend other diffusion-type models by including leakage and competition by means of lateral inhibition. These additional mechanisms have proven effective in explaining how, for example, time sensitive stimulus information can give way to differences in individual subject performance. For example, Usher and McClelland [38] and Tsetsos et al. [167] have shown the effects of primacy and recency for some subjects in a time-varying stimulus information paradigm. In these multi-alternative choice experiments, one response option may receive the strongest "input" (e.g., the brightness level) for the first 500 ms, but then the stimuli transition such that a different response option receives the strongest input relative to the first. In both of these studies, different parameterizations of the LCA model were used to demonstrate how primacy effects could be appreciated by having a large value for lateral inhibition relative to the strength of the input (i.e., the drift rate), and recency effects could be captured through a large leakage term relative to the input [38, 167].

As a specific example of how the neurosciences have guided the assumptions in the LCA model, it is well known that the firing rate of individual neurons can never be negative. However, these firing rates can be attenuated by way of inhibition – a process carried out by other neurons in the system. To instantiate these neuronal dynamics, the full LCA model enforces a constraint such that if the degree of evidence for any choice alternative becomes negative, the degree of evidence for that accumulator should be reset to zero [38]. The floor-on-activation constraint was later found to be critical in capturing patterns of individual differences in multi-alternative choice that could not be captured by other diffusion-type models [167]. It is worth noting that other neurological constraints allow the LCA model to provide a unique characterization of behavioral data that would not otherwise be realized; specifically, the role of lateral inhibition relative to leakage in the model plays an interesting role in characterizing subject-specific patterns in behavioral data [73, 167–176].

Imposing neural constraints on a behavioral model involves a series of choices about what the appropriate constraints are, and to which mechanisms they should be applied. Given the highly subjective nature of how neural constraints should be imposed on a behavioral model, it should not be surprising that a great deal of controversy surrounds some applications of the Theoretical approach. While neural network modelers have undoubtedly derived inspiration from the brain in building their models, the mechanistic implementation of these inspirations is often interpreted as a strong commitment, which opens the gates for scrutiny about plausibility and falsifiability [71, 160, 177]. Furthermore, in some cases these additional neural mechanisms do not provide any advantage in terms of quantitative fit statistics to behavioral data over their simpler counterparts (e.g., see [178], but also see [73] and [176] for a different perspective). If a "neurally plausible" model provides no advantage in fitting behavioral data over an "off the shelf" behavioral model, it begs the question of which model should be preferred. Of course, there are many other ways of linking neural measures to model mechanisms that are more agnostic, requiring weaker commitments to a particular linking hypothesis [3, 57]. For now, we set aside the interesting theoretical debate about neural plausibility, and turn to other analytic approaches.

6.1.2 Two-Stage Behavioral Approach

The first formal linking approach uses neurophysiology to replace *parameters* of a behavioral model. For example, consider a model that explains some neural data N with parameters δ, and behavioral data B with parameters θ. The neural parameters δ could be divided into a set of parameters characterizing a key neural signal δ_1, and a set of nuisance parameters δ_2 so that $\delta = \{\delta_1, \delta_2\}$. Now suppose the behavioral model parameters could be divided into a set of parameters that are reflective of

the behavioral signal θ_1, and a set of parameters θ_2 that are not. The structure of the Two-stage Behavioral approach is to simply replace the set of parameters θ_1 with the parameters of the neural signal δ_1. One important difference is that the constraint applied in the Two-stage Behavioral approach is procedural, meaning that the process of replacing model parameters is sequential, as illustrated by the dashed arrows in the top-middle panel of Fig. 1.3. Parameter estimates for δ are first estimated from the data, and are then used as constraints to specify the range of possible values that θ can take when fitting the behavioral model to behavioral data B. The Two-stage Behavioral approach makes a strong commitment to how the neural signal is represented in the abstract mechanisms assumed by the behavioral model, and as a result, it is a stronger instantiation of neurophysiology than the Theoretical approach discussed above.

The Two-stage Behavioral approach is nicely illustrated by the work of Wang and colleagues [179], who developed a spiking neural network model of perceptual decision making. This model aims to account for the same kinds of behaviors as the DDM and the LCA model, but is far less abstract, with thousands of simulated spiking neurons, dense patterns of excitatory and inhibitory connections, pools of neurons associated with a single response, and the dynamics of individual neurons defined by several differential equations. While the model has dozens of potentially free parameters, most of them are defined directly by neural data. For example, the time constants of integration of different inhibitory and excitatory receptor types are based directly on physiological measures. While low-level spiking neural network models of this sort capture well many of the details of neurons and neural circuits and provide reasonable first-order predictions of behavioral data, they are difficult to simulate and quantitative fits to behavioral data are simply impossible using even state-of-the-art computer hardware [180]. Indeed, as a result of this additional complexity, very few efforts have been devoted to systematically studying the model's predictions for choice response time data. However, a few approximations have been developed for fitting purposes, and these approximations behave similarly to popular models in cognitive science such as the LCA model [168, 179, 181].

6.1.3 Direct Input Approach

The Two-stage Behavioral Approach represents one way in which the neural data can guide the behavioral model through neural model parameters, but other approaches are more direct. For example, rather than translating the neural data N to the neural model parameters δ, and then using δ to constrain the behavioral model parameters θ, one could instead use the neural data to directly replace dynamics of the behavioral model. We call this analytic approach the Direct Input approach, and it has been nicely illustrated by the Vanderbilt group [17, 97, 172]. In particular, Purcell et al. [97] examined perceptual decision making within the sequential sampling model architecture assumed by many modern models [37, 38, 182, 183].

Their works focused on understanding how different types of neurons in the frontal eye field (FEF) carry out different computations specified in accumulator models, namely that visually-responsive neurons in FEF encode the drift rate driving the decision process and that movement-related neurons in FEF instantiate the accumulation process itself. To test their linking proposition most directly [2, 3], they replaced the parameterized mechanisms thought to be embodied by the visually-responsive neurons, namely the time for perceptual processing and the drift rate, with the neurophysiological data recorded from visually-responsive neurons. Rather than having abstract mathematical and computational components drive the decision process, the neural data (N) were used to drive the decision process directly. To do this, the neural data were used to replace components of the model that would otherwise have been latent, and would need to be estimated from behavioral data. The only remaining free parameters were those that defined the decision making process (i.e., race, feedforward, lateral, or gated accumulation), and that defined speed-accuracy tradeoffs (i.e., threshold of accumulation). When constrained by neural inputs, they observed that only some decision making architectures could fit the full set of behavioral data. They were then able to distinguish further between models based on how well the predicted accumulator model dynamics matched the observed neural dynamics in movement-related neurons, the neurons they hypothesized to carry out an accumulation of evidence.

Although the Direct Input Approach is commonly used to feed neural data into a cognitive model, one could potentially invert the direction of influence in Fig. 6.1 to analyze the neural data as a function of some behavioral variable, such as accuracy [107] or response time [4, 108]. Once the neural data have been sorted as a function of the levels of the behavioral outcome, one might analyze the distribution of neural data between these levels [184]. Such a procedure has been the dominant analytic approach in neuroscience since its inception, but is not model-based, and so we will not consider it here. However, the model-based analogue of this analysis would be to use the model's machinery to drive the analysis of neural data. We refer to this approach as the Latent Input Approach, and will discuss it further in the next section.

6.2 Behavioral Model Predicts Neural Data

Another goal that a researcher could have would be to investigate where in the brain a particular model mechanism is carried out. In this approach, one would first fit a behavioral model to behavioral data, to gain a sense of what parameters accurately describe the behavioral data. Once the parameters have been inferred, their variability across subjects or trials can be related to patterns in the neural data to determine whether or not the two variables are systematically related. In this section, we discuss two different approaches for accomplishing this goal: the Latent Input and the Two-stage Neural Approaches.

6.2.1 *Latent Input Approach*

The Latent Input approach can be thought of as a model-based version of a standard general linear model (GLM) analysis. In a GLM-type analysis, we might investigate how neural activity varies as a function of experimental condition, or even response type, such as correct or incorrect. In these examples, the variable we related to the neural activity is meant to sieve through the neural data to determine which brain areas are systematically impacted as a function of the variable of interest. However, it is easy to image using a different variable to sieve through the neural data, perhaps even using a variable that takes the form of an estimated model parameter. In a nut shell, the Latent Input approach does exactly this, where the variable of interest is some type of estimate or prediction from a model of interest.

There are three stages to a typical Latent Input approach, illustrated in the middle-left panel of Fig. 6.2 as increasingly light gray arrows. First, the parameters of a cognitive model θ are estimated by fitting the model to behavioral data B alone. Second, the resulting parameter estimates are used to generate predictions about neural data N^*, which typically represents some "internal state" of the cognitive model in terms of the neural measure. Third, one searches through the neural data N to identify correlates of the model's internal state N^*.

To get a sense of what we mean by the "internal state" of the model, let's consider an application. The Latent Input Approach is commonly used to relate mechanisms of reinforcement learning models to brain data [185–189]. One classic example of a reinforcement learning model is the Rescorla-Wagner (RW) model that characterizes the process of learning a conditioned response through repeated presentations of a conditioned stimulus [190]. In the model, the value of the unconditioned stimulus is represented as u, and the value of the conditioned stimulus

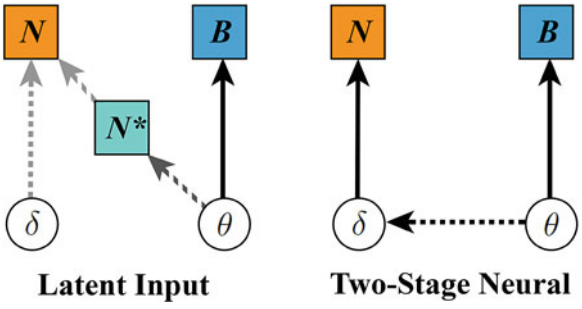


Fig. 6.2 An illustration of approaches for linking neural and behavioral data when the goal is to use the behavioral model to predict neural data. N represents the neural data, B represents the behavioral data, N^* represents simulated internal model states, and θ and δ represent model parameters. When an approach is procedural, progression through processing stages is represented by arrows of decreasing darkness (e.g., the Latent Input Approach). Dashed lines indicate conceptual constraints, whereas solid lines indicate statistical constraints. Squares represent observable measures of cognitive processes, whereas the circles represent (latent) model parameters

on Trial t is represented as v_t. To learn the stimulus environment, the model assumes that v_t is updated sequentially according to a learning rate parameter α, and an evaluation of the prediction error ϵ. Specifically, after a decision is made and the unconditioned stimulus is presented, the model's internal state of the value of the conditioned stimulus is updated according to the rule

$$v_t = v_{t-1} + \alpha\epsilon. \tag{6.1}$$

A common assumption about the prediction error ϵ is that it should be the difference between what the true state of the reward u is and what the RW model believe it to be on trial t, such that

$$\epsilon = |u - v_{t-1}|. \tag{6.2}$$

Eventually, the internal representation of the value v converges to u, ϵ approaches zero, and the model "learns" the stimulus-to-response pairing.

The process of fitting the RW model to data is how we obtain an estimate for the parameter α. However, the parameter α is just the learning rate for one subject: it does not vary from one trial to the next. To obtain an internal state of the model, we must increase our resolution in relating the model to the behavioral data. To do this, we can use the estimated value of α to determine what the prediction error is for each trial by evaluating Equation 6.2. The prediction error values predicted from the model now represent the model's internal state, and so it can be entered into an fMRI analysis as a time series by convolving them with a hemodynamic response function (HRF), and regressing the result against the fMRI data through the GLM. Interestingly, the prediction errors ϵ are not parameters; instead they represent what the model believes to be the observer's misunderstanding of the value of the unconditioned stimulus. The distinction between internal state and model parameter is important because it separates the Latent Input approach from other possible Two-stage approaches, such as in Van Maanen et al. [11], which we discuss below.

As the previous example makes clear, Latent Input approaches can identify candidate neural substrates for theoretical concepts, such as prediction error, that are not directly observable but can be defined within a cognitive model. Entering latent model measures into the imaging analyses is relatively straightforward. Indeed, multiple model measures can be considered simultaneously. For example, Davis et al. [191] simultaneously analyzed cognitive operations related to recognition and representational uncertainty by including two related measures in the imaging analysis from a cognitive model fit to trial-by-trial category learning data.

A final advantage of the Latent Input approach is in how the approach can be used to differentiate among competing cognitive theories. It is often the case that many models of the same behavior exist, and these models rarely agree about the mechanisms underlying the cognitive process. Unfortunately, the disagreement between mechanisms can be difficult to resolve, and some have even argued that behavioral data alone are insufficient for disentangling some models [192]. So, what happens when different models are used to interpret neurophysiological

measures, and can these interpretations be objectively evaluated in a way that produces one clear mechanistic champion? Recently, Mack et al. [13] developed a strategy for evaluating models on the basis of consistency with neural data. Their approach is based on the Latent Input approach, but differs in two important ways. First, their method does not assume that the model used to interpret the neural data is correct. Instead, they posit a set of competing explanations for the underlying cognitive process, and the *best* explanation is to be determined from each model's correspondence to the neural data. Second, they exploited machine learning techniques called multivariate pattern analysis (MVPA), and representation similarity analysis (RSA) to assess the correspondence between model and data.

Mack et al. [13] used their method to test two theories of category representation: the exemplar and prototype representations. The exemplar representation is a concrete one, and it assumes that members of a category are explicitly stored in memory, and a categorical decision for a new stimulus is determined as a function of similarity to these memory stores. On the other hand, the prototype representation assumes that categories are represented as the average of the features of these memory stores. In this sense, the prototype representation is abstract – a category could be represented in a location of feature space that is not representative of any particular known category member. These particular theories of representation have been fiercely debated for decades [193–195]. Indeed, in their first analysis, Mack et al. showed that both representations provided nearly indistinguishable fits to the behavioral data. However, the secondary analysis used their method to test consistency with neural data. To do this, they derived a measure of the model's internal state by monitoring the models' evaluation of the similarity of new stimuli to known categories. This internal state is the mechanism in the model that drives the decision process, and so at the outset, one might expect that the internal state is not particularly diagnostic. However, in both an MVPA and RSA, the patterns of brain activity across trials showed better correspondence to the internal state of the exemplar representation. These findings serve as a powerful example of how the neurosciences – combined with a Latent Input approach – allow researchers to draw conclusions that they might not otherwise have reached.

6.2.2 Two-Stage Neural Approach

The second approach is related to the Two-stage Behavioral approach discussed above, except that the direction of flow is opposite from the Two-stage Neural approach. Here, the parameters of the behavioral model θ are used to guide the analysis of the neural data N. As illustrated in Fig. 6.2, the constraint applied to the model is procedural, meaning that after the behavioral model has been fit to the behavioral data, the behavioral model parameters θ are regressed or correlated with the neural model parameters δ. Strong correlations are used to support mechanistic interpretations of brain function.

The Two-stage Neural approach has been especially successful in the field of perceptual decision making [196]. For example, Forstmann and colleagues [86, 88, 197] show in various experimental setups that accumulator model parameters that reflect response caution correlate with averaged BOLD responses in pre-supplementary motor area and striatum, two regions in the brain that are thought to be involved in mediating cognitive control. These studies illustrate that individual differences in behavior, captured by hypothesized processes, are driven by individual differences in how the brain works. This approach thus strengthens our understanding of the role of certain brain areas in cognition, but it also adds credence to the type of cognitive model that is adopted to describe behavior.

In other applications, researchers have related trial-to-trial neural measures to parameter estimates that also fluctuate from trial-to-trial [11, 198–201]. For example, Behrens et al. [198] used an optimal model that updated the expected reward for one of two responses on a trial-by-trial basis. The parameters of this model were also updated on a trial-by-trial basis, based on the actual trial outcome (i.e., the choice of the participant) and the expected outcome (i.e., the model prediction). Behrens and colleagues found that the level at which participants were responsive to changes in the rewards was predictive of anterior cingulate cortex activation on a trial-by-trial basis, supporting the idea that anterior cingulate cortex activation reflects changes in the environment [202].

A slightly different approach was taken by Van Maanen and colleagues [11, 15, 203]. Using the LBA model, these authors estimated the most likely combination of drift rate and starting point of evidence accumulation, given the distribution of these parameters across trials. The most likely combination of parameters is determined by the set of parameters that specify the response time. While powerful, this method is difficult because the most likely parameter estimates are highly uncertain, due to the large variability in the joint distribution of the model parameters, and due to the simplification of the model to include only two sources of variability. Nevertheless, Van Maanen et al. [11] showed that trial-to-trial fluctuations in BOLD in pre-supplementary motor area correlated with the trial-to-trial measure of threshold, but only for speed-stressed trials. This finding was corroborated by Boehm et al. [15], who found a similar correlation between the trial-to-trial model parameter and a trial-to-trial estimate of the Contingent Negative Variation (CNV). The CNV is a slow rising potential, thought to represent neural activation in a cortico-basal ganglia loop including the supplementary/pre-supplementary motor areas [204, 205].

Although the Two-stage Neural Approach has been instrumental in elucidating various mechanistic explanations of neural data, because this approach uses a procedural constraint, it does not apply the full set of constraints that are possible from the data. In this way, Two-stage frameworks are somewhat limited because the neural data cannot influence the parameter estimates of the behavioral model [12]. To accomplish such a goal, the desired approach would need to automatically learn the covariation of the neural and behavioral parameters in harmony with the analysis of the neural and behavioral data. While there are potentially many other ways of completing this procedure, the methods and applications described within this book have hopefully given you some insight into the joint modeling approach.

6.3 Conclusions

In this chapter, we have discussed several other ways of linking neural and behavioral measures. Our approach was to divide these linking strategies on the basis of other intended goals of the researcher: using neural data to constrain or guide the development of a behavioral model, and using the behavioral model to predict neural data. As we generally refer to the goal of simultaneously considering both neural and behavioral data together, we have referred to this goal as "joint" modeling. Ultimately, while our division is convenient for purposes of taxonomy, there are many different ways of exploring brain and behavior relationships, each way bringing with it advantages and disadvantages. We recommend that a variety of linking methods be used in each new research endeavor, moving from purely exploratory methods to more confirmatory methods as more evidence is obtained enabling the formation of stronger hypotheses.

Chapter 7
Conclusions

Fundamentally, this book considered the problem of integrating multiple measures of cognition. Specifically, the focus has been to relate neurophysiological measures of brain function to the mechanisms instantiated within cognitive models. Cognitive models are intended to make a theory about a cognitive process explicit, and the ability of the model to capture important trends in behavioral data is the impetus for the model's success. The efforts described in this book were dedicated to creating a framework for extending the application of cognitive theories to both neural and behavioral measures. In short, both neural and behavioral data can be thought of as manifestations of the one thing psychologists care most about: the mind.

David Marr [206] proposed a set of three levels of analysis to describe how theories about the mind should be developed. The framework is intended to be hierarchical, where one level helps to constrain other levels in a three-tier system. First, at the highest level, the computational level considers the goal of the organism as it operates conditional on the structure of the environment. At this level, no consideration is given to the particular mechanism used and so many approaches at this level lack a good deal of constraint in accurately characterizing the organism's goal [207–210]. Second, the algorithmic level details the type of representation and mechanisms that underlie cognition, effectively describing how a goal is carried out and consequently how data manifest, as they would in a typical behavioral experiment. At this level, mathematical and statistical abstractions are the primary tools used for advancing theory, where biological constraints are rarely applied [211]. Third, the implementation level describes how interactions at the most fundamental level in the system are carried out in a biological substrate, such as those observed between neurons and their connections in the brain [212]. In this way, the implementation level can be used to constrain the set of possible mechanisms that could be invoked at the algorithmic level, by using actual biological principles.

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1_7

Given the hierarchical structure of Marr's proposal, one may immediately imagine scenarios where conducting analyses at different levels would lead to the best result. Indeed, Marr considered how the connections between these levels could be used to productively constrain knowledge at any level independently. Unfortunately, the structure of the developing scientific community is highly specialized by virtue of the intricacies involved within each level of analysis. For example, modern theories about perceptual decision making detail how information is sequentially sampled through time until a decision is reached [36–38, 213, 214]. These theories have a number of interesting differences and similarities, provoking the need for intensely testing and evaluating the assumptions made by each theory as they relate to behavioral data. However, testing these theories is no easy task. First, one must make a specific assumption about what mechanism in the model should be allowed to vary as a function of the experimental manipulation. Second, evaluating how well the theory aligns with the data requires that one fit a model to said data. Considering only these models, a number of ingenious statistical methods have been proposed to facilitate model fitting [175, 176, 215–220], each with their own advantages and disadvantages. Learning the underlying theories of each of the aforementioned models as well as how to fit the models to data requires intense study for several years, a timetable that spans the length of the maturation cycle of an aspiring graduate student. Effectively, by the time a developing student has mastered all of these skills, it becomes time to apply for positions at the next level in the academic hierarchy, leaving little time to pursue other tantalizing questions such as "how is information integrated in the brain in the first place?"

Of course, similar stories exist at other levels of analysis, such as neuroscientific studies of single-unit neurons. However, the simple example above makes clear how the structure of the academic environment naturally evolves the scientific community into "compartments", where one level of analysis is exploited in the pursuit of a major theoretical development, and other levels are no longer carefully considered. In Chap. 1, Fig. 1.1 illustrated the major approaches for studying how a cognitive process (top right side) might have given rise to the observed behavioral data (top left side) or neurophysiological data (bottom). In terms of Marr's levels of analysis, mathematical psychologists tend to focus on the computational and algorithmic levels, experimental psychologists focus more on computational levels, and cognitive neuroscientists focus more on the implementation level. Each approach can be used to develop, advance, substantiate, or refute a particular theory about a given cognitive process. However, in maintaining a tight focus, certain opportunities are missed. We believe that at some point theoretical progress is hindered by a lack of constraint brought on by focusing on a single level of analysis [16]. In essence, as more complex data become available, more advanced theories are developed to provide the degree of flexibility needed to capture the data. As the theories become more complex, we arrive at a theoretical stalemate where each theory can account for the data just as well as other theories. At this point, continuing to pursue a singular focus is an unproductive way to advance one's theory, and so we join the list of a growing number of researchers who advocate for multiple levels of analysis [1–3, 16, 221].

The models considered in this book are not designed with the intention of instantiating Marr's levels of analysis. Instead, we see joint modeling as a way of unraveling an extremely puzzling tapestry of data. It is a first step toward instantiating explicit linking functions relating measures of the brain to theories about the mind [3]. Analogous to the field of mathematical psychology, our goal was to motivate researchers to attempt different types of linking functions as a way to formally test the assumptions of their theory. Because joint models are statistically constrained by both neural and behavioral data, they serve as a framework for imposing reciprocity in either a confirmatory or exploratory manner [1, 68]. It is our hope that joint models will eventually be replaced with models that are completely integrative: theories instantiated by mechanisms with a careful eye toward all three of Marr's levels. Until then, joint models can be used in a progressive way, where stronger and stronger constraints can be applied (e.g., within the prior or the linking structure itself) as we gain a better and better understanding of the neural bases of theoretically-derived computational mechanisms.

References

1. B.U. Forstmann, E.J. Wagenmakers, T. Eichele, S. Brown, J.T. Serences, Trends Cogn. Sci. **15**, 272 (2011)
2. D.Y. Teller, Vis. Res. **24**, 1233 (1984)
3. J.D. Schall, Annu. Rev. Psychol. **55**, 23 (2004)
4. D.P. Hanes, J.D. Schall, Science **274**, 427 (1996)
5. E.N. Dzhafarov, Psychometrika **58**, 281 (1993)
6. G. Corrado, K. Doya, J. Neurosci.: Off. J. Soc. Neurosci. **27**, 8178 (2007)
7. K. Friston, Science **326**, 399 (2009)
8. J.I. Gold, M.N. Shadlen, Annu. Rev. Neurosci. **30**, 535 (2007)
9. R.B. Mars, M.C. Klein, F.X. Neubert, E. Olivier, E.R. Buch, E.D. Boorman, M.F.S. Rushworth, J. Neurosci.: Off. J. Soc. Neurosci. **29**, 6926 (2009)
10. R.B. Mars, N.J. Shea, N. Kolling, M.F.S. Rushworth, Q. J. Exp. Psychol. **65**, 252 (2012)
11. L. van Maanen, S.D. Brown, T. Eichele, E.J. Wagenmakers, T. Ho, J. Serences, J. Neurosci. **31**, 17488 (2011)
12. B.M. Turner, B.U. Forstmann, E.J. Wagenmakers, S.D. Brown, P.B. Sederberg, M. Steyvers, NeuroImage **72**, 193 (2013)
13. M.L. Mack, A.R. Preston, B.C. Love, Curr. Biol. **23**, 2023 (2013)
14. T.J. Palmeri, Trends Cogn. Sci. **18**, 67 (2014)
15. U. Boehm, L. Van Maanen, B. Forstmann, H. Van Rijn, NeuroImage **96**, 95 (2014)
16. B.C. Love, Top. Cogn. Sci. **7** (2015)
17. T. Palmeri, J. Schall, G. Logan, in *Oxford Handbook of Computational and Mathematical Psychology*, ed. by J.R. Busemeyer, J. Townsend, Z.J. Wang, A. Eidels (Oxford University Press, Oxford/New York, 2015)
18. B.M. Turner, L. Van Maanen, B.U. Forstmann, Psychol. Rev. **122**, 312 (2015)
19. T.J. Palmeri, B.C. Love, B.M. Turner, Model-based cognitive neuroscience. J. Math. Psychol. **76**, 56–64 (2017)
20. B.U. Forstmann, R. Ratcliff, E.J. Wagenmakers, Annu. Rev. Psychol. **67**, 641 (2016)
21. J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann, San Francisco, 1988)
22. J.P. Borst, N.A. Taatgen, H. Van Rijn, J. Exp. Psychol.: Learn. Mem. Cogn. **36**, 363 (2010)
23. J.P. Borst, J.R. Anderson, Proc. Natl. Acad. Sci. U. S. **110**, 1628 (2013)
24. J.R. Anderson, *How Can the Human Mind Occur in the Physical Universe?* (Oxford University Press, New York, 2007)
25. J.R. Anderson, D. Byrne, J.M. Fincham, P. Gunn, Cereb. Cortex **18**, 904 (2008)

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1

26. J.R. Anderson, J.M. Fincham, Y. Qin, A. Stocco, Trends Cogn. Sci. **12**, 136 (2008)
27. J.P. Borst, N.A. Taatgen, A. Stocco, H. Van Rijn, PLoS ONE **5**, e12966 (2010)
28. J.P. Borst, M. Nijboer, N.A. Taatgen, H. Van Rijn, J.R. Anderson, PLoS ONE **10**, e0119673 (2015)
29. J.P. Borst, J.R. Anderson, J. Math. Psychol. **76**, 94 (2017)
30. J.R. Anderson, S. Betts, J.L. Ferris, J.M. Fincham, Proc. Natl. Acad. Sci. U. S. **107**, 7018 (2010)
31. J.R. Anderson, Neuropsychologia **50**, 487 (2012)
32. A. Mohammad-Djafari, O. Féron, Int. J. Imaging Syst Technol **16**, 215 (2006)
33. M.D. Nunez, R. Srinivasan, J. Vandekerckhove, Front. Psychol. 8–18 (2015)
34. M. Frank, C. Gagne, E. Nyhus, S. Masters, T.V. Wiecki, J.F. Cavanagh, D. Badre, J. Neurosci. **35**(2), 485 (2015)
35. M.D. Nunez, J. Vandekerckhove, R. Srinivasan, How attention influences perceptual decision making: single-trial EEG correlates of drift-diffusion model parameters (2016, in press)
36. S. Brown, A. Heathcote, Cogn. Psychol. **57**, 153 (2008)
37. R. Ratcliff, Psychol. Rev. **85**, 59 (1978)
38. M. Usher, J.L. McClelland, Psycholog. Rev. **108**, 550 (2001)
39. G.D. Logan, T. Van Zandt, F. Verbruggen, E.J. Wagenmakers, Psychol. Rev. **121**, 66 (2014)
40. D. van Ravenzwaaij, A. Provost, S.D. Brown, J. Math. Psychol. **76**, 131 (2017)
41. L. van Maanen, H. Van Rijn, Top. Cogn. Sci. **2**, 168 (2010)
42. L. van Maanen, H. Van Rijn, N.A. Taatgen, Cogn. Sci. **36**, 62 (2012)
43. L. van Maanen, H. Van Rijn, J.P. Borst, Psychon. Bull. Rev. **16**, 987 (2009)
44. C.A. Rodriguez, B.M. Turner, T. Van Zandt, S.M. McClure, Eur. J. Neurosci. 1–11 (2015)
45. B.M. Turner, in *An Introduction to Model-Based Cognitive Neuroscience*, ed. by B.U. Forstmann, E.J. Wagenmakers (Springer, New York, 2015), pp. 199–220
46. B.M. Turner, C.A. Rodriguez, T. Norcia, M. Steyvers, S.M. McClure, NeuroImage **128**, 96 (2016)
47. J.N. Rouder, J. Lu, Psychon. Bull. Rev. **12**, 573 (2005)
48. J.N. Rouder, J. Lu, P. Speckman, D. Sun, Y. Jiang, Psychon. Bull. Rev. **12**, 195 (2005)
49. W.Y. Ahn, A. Krawitz, W. Kim, J.R. Busemeyer, J.W. Brown, J. Neurosci. Psychol. Econ. **4**, 95 (2011)
50. M. Plummer, in *Proceedings of the 3rd international Workshop on Distributed Statistical Computing*, Vienna (2003)
51. C. Ranganath, A.P. Yonelinas, M.X. Cohen, C.J. Dy, S.M. Tom, M. D'Esposito, Neuropsychologia **42**, 2 (2004)
52. L.L. Eldridge, B.J. Knowlton, C.S. Furmanski, S.Y. Bookheimer, S.A. Engel, Nat. Neurosci. **3**, 1149 (2000)
53. R.M. Shiffrin, M. Steyvers, Psychon. Bull. Rev. **4**, 145 (1997)
54. S. Dennis, M.S. Humphreys, Psychol. Rev. **108**, 452 (2001)
55. A.F. Osth, S. Dennis, Psychol. Rev. **122**, 260 (2015)
56. W.H. Batchelder, D.M. Riefer, Psychon. Bull. Rev. **6**, 57 (1999)
57. B.M. Turner, B.U. Forstmann, B.C. Love, T.J. Palmeri, L. Van Maanen, J. Math. Psychol. **76**, 65 (2017)
58. R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis* (Pearson Prentice Hall, Upper Saddle River, 2007)
59. J.A. Rice, *Mathematical Statistics and Data Analysis* (Duxbury Press, Belmont, 2007)
60. A. Heathcote, S.D. Brown, E.J. Wagemakers, in *An Introduction to Model-Based Cognitive Neuroscience*, ed. by B.U. Forstmann, E.J. Wagenmakers (Springer, New York, 2015), pp. 25–48
61. M.D. Lee, W. Vanpaemel, Cogn. Sci. **32**, 1403 (2008)
62. W. Vanpaemel, J. Math. Psychol. **54**, 491 (2010)
63. W. Vanpaemel, J. Math. Psychol. **55**, 106 (2011)
64. W. Vanpaemel, M.D. Lee, Psychon. Bull. Rev. **19**, 1047 (2012)
65. M.D. Lee, E.J. Wagenmakers, *Bayesian Modeling for Cognitive Science: A Practical Course* (Cambridge University Press, Cambridge, 2013)

66. C.J.F. ter Braak, Stat. Comput. **16**, 239 (2006)
67. B.M. Turner, P.B. Sederberg, S.D. Brown, M. Steyvers, Psychol. Methods **18**, 368 (2013)
68. B.M. Turner, T. Wang, E. Merkel, NeuroImage **153**, 28 (2017)
69. D. Wabersich, J. Vandekerckhove, Behav. Res. Methods **46**, 15 (2014)
70. J.J. Palestro, G. Bahg, P.B. Sederberg, Z.L. Lu, M. Steyvers, B.M. Turner, J. Math. Psychol. **84**, 20 (2018)
71. S. Roberts, H. Pashler, Psychol. Rev. **107**, 358 (2000)
72. B.M. Turner, S. Dennis, T. Van Zandt, Psychol. Rev. **120**, 667 (2013)
73. A.R. Teodorescu, M. Usher, Psychol. Rev. **120**, 1 (2013)
74. I.J. Myung, M.A. Pitt, Psychon. Bull. Rev. **4**, 79 (1997)
75. I.J. Myung, J. Math. Psychol. **44**, 190 (2000)
76. I.J. Myung, M. Forster, M.W. Browne, J. Math. Psychol. **44**, 1 (2000)
77. J.I. Myung, M. Montenegro, M.A. Pitt, J. Math. Psychol. **51**, 198 (2007)
78. M. Montenegro, J.I. Myung, M.A. Pitt, REM integral expressions (2011). Unpublished manuscript
79. A. Gelman, J.B. Carlin, H.S. Stern, D.B. Rubin, *Bayesian Data Analysis* (Chapman and Hall, New York, 2004)
80. C.C. Liu, M. Aitkin, J. Math. Psychol. **52**, 362 (2008)
81. Q.F. Gronau, A. Sarafoglou, D. Matzke, A. Ly, U. Boehm, M. Marsman, D.S. Leslie, J.J. Forster, E.J. Wagenmakers, H. Steingroever, J. Math. Psychol. **81**, 80 (2018)
82. I.J. Myung, M. Pitt, in *Stevens' Handbook of Experimental Psychology*, 3rd edn., ed. by H. Pashler, J. Wixted (Wiley, New York, 2002), pp. 429–460
83. P. Cassey, G. Gaut, M. Steyvers, S.D. Brown, A generative joint model for spike trains and saccades during perceptual decision making. Psychon. Bull. Rev. **23**, 1757–1778 (2016)
84. J.W. Mink, Prog. Neurobiol. **50**, 381 (1996)
85. Y. Smith, M.D. Bevan, E. Shink, J.P. Bolam, Neuroscience **86**, 353 (1998)
86. B.U. Forstmann, G. Dutilh, S. Brown, J. Neumann, D.Y. von Cramon, K.R. Ridderinkhof, E.J. Wagenmakers, Proc. Natl. Acad. Sci. **105**, 17538 (2008)
87. R. Bogacz, E.J. Wagenmakers, B.U. Forstmann, S. Nieuwenhuis, Trends Neurosci. **33**, 10 (2010)
88. B.U. Forstmann, A. Anwander, A. Schäfer, J. Neumann, S. Brown, E.J. Wagenmakers, R. Bogacz, R. Turner, Proc. Natl. Acad. Sci. **107**, 15916 (2010)
89. B.U. Forstmann, M. Tittgemeyer, E.J. Wagenmakers, J. Derrfuss, D. Imperati, S. Brown, J. Neurosci. **31**, 17242 (2011)
90. T. Behrens, H. Johansen-Berg, M.W. Woolrich, S.M. Smith, C.A. Wheeler-Kingshott, P.A. Boulby, G.J. Barker, E.L. Sillery, K. Sheehan, O. Ciccarelli, A.J. Thompson, J.M. Brady, P.M. Matthews, Nat. Neurosci. **6**, 750 (2003)
91. C. Donkin, S. Brown, A. Heathcote, E.J. Wagenmakers, Psychon. Bull. Rev. **18**, 61 (2011)
92. M. Stone, Psychometrika **25**, 251 (1960)
93. C. Donkin, A. Heathcote, S. Brown, in *9th International Conference on Cognitive Modeling – ICCM2009*, Manchester, ed. by A. Howes, D. Peebles, R. Cooper (2009)
94. D.J. Spiegelhalter, N.G. Best, B.P. Carlin, A. van der Linde, J. R. Stat. Soc. B **64**, 583 (2002)
95. A. Tomohiro, Biometrika **94**, 443 (2007)
96. M.A. Pitt, I.J. Myung, S. Zhang, Psychol. Rev. **109**, 472 (2002)
97. B. Purcell, R. Heitz, J. Cohen, J. Schall, G. Logan, T. Palmeri, Psychol. Rev. **117**, 1113 (2010)
98. R. Ratcliff, Psychol. Rev. **120**, 281 (2013)
99. M. Jones, E.N. Dzhafarov, Psychol. Rev. **121**, 1 (2014)
100. A. Heathcote, E.J. Wagenmakers, S.D. Brown, Psychol. Rev. **121**, 676 (2014)
101. M. Peruggia, T. Van Zandt, M. Chen, Case Stud. Bayesian Stat. **VI**, 319 (2002)
102. P. Craigmile, M. Peruggia, T.V. Zandt, Psychometrika **75**, 613 (2010)
103. M.E. Raichle, A.M. MacLeod, A.Z. Snyder, W.J. Powers, D.A. Gusnard, G.L. Shulman, Proc. Natl. Acad. Sci. U. S. **98**, 676 (2001)
104. D.A. Gusnard, M.E. Raichle, Nat. Rev. Neurosci. **2**, 685 (2001)
105. M.E. Raichle, A.Z. Snyder, NeuroImage **37**, 1083 (2007)

106. K. Christoff, A.M. Gordon, J. Smallwood, R. Smith, J.W. Schooler, Proc. Natl. Acad. Sci. U. S. **106**, 8719 (2009)
107. T. Eichele, S. Debener, V.D. Calhoun, K. Specht, A.K. Engel, K. Hugdahl, D.Y. von Cramon, M. Ullsperger, Proc. Natl. Acad. Sci. U. S. **16**, 6173 (2008)
108. D.H. Weissman, K.C. Roberts, K.M. Visscher, M.G. Woldorff, Nat. Neurosci. **9**, 971 (2006)
109. M. Mittner, W. Boekel, A.M. Tucker, B.M. Turner, A. Heathcote, B.U. Forstmann, J. Neurosci. **34**, 16286 (2014)
110. C.P. Robert, G. Casella, *Monte Carlo Statistical Methods* (Springer, New York, 2004)
111. M.J. Mulder, E.J. Wagenmakers, R. Ratcliff, W. Boekel, B.U. Forstmann, J. Neurosci. **32**, 2335 (2012)
112. C. Summerfield, E. Koechlin, Front. Hum. Neurosci. **4**, 208 (2010). https://doi.org/10.3389/fnhum.2010.00208
113. X. Bai, V.L. Towle, E.J. He, B. He, NeuroImage **35**, 598 (2007)
114. C.N. White, E. Congdon, J.A. Mumford, K.H. Karlsgodt, F.W. Sabb, N.B. Freimer, E.D. London, T.D. Cannon, R.M. Bilder, R.A. Poldrack, J. Cogn. Neurosci. **26**, 1601 (2014)
115. K. Amano, N. Goda, S. Nishida, Y. Ejima, T. Takeda, Y. Ohtani, J. Neurosci. **26**, 3981 (2006)
116. F.H. Lin, A.R. McIntosh, J.A. Agnew, G.F. Eden, T.A. Zeffiro, J.W. Belliveau, NeuroImage **20**, 625 (2003)
117. E. Martinez-Montes, P.A. Valdes-Sosa, F. Miwakeichi, R.I. Goldman, M.S. Cohen, NeuroImage **22**, 1023 (2004)
118. C.F. Beckmann, S.M. Smith, NeuroImage **25**, 294 (2005)
119. J. Liu, V.D. Calhoun, *Proceedings of IEEE ISBI Conference*, Arlington (2007), pp. 1028–1031
120. V.D. Calhoun, T. Adali, J. Liu, *Proceedings of the 28th IEEE EMBS Annual International Conference*, New York (2006), pp. 3672–3675
121. V.D. Calhoun, J. Liu, T. Adali, NeuroImage **45**, 163 (2009)
122. T. Eichele, V.D. Calhoun, S. Debener, Int. J. Psychophysiol. **75**, 53 (2009)
123. A.R. Franco, J. Ling, A. Caprihan, V.D. Calhoun, R.E. Jung, G.L.E.A. Heileman, J. Sel. Top. Signal Process. **2**, 986 (2008)
124. S.J. Teipel, A.L. Bokde, T. Meindl, E.J. Amaro, J. Soldner, M.F. Reiser, S.C. Herpertz, H.J. M oller, H. Hampel, NeuroImage **49**, 2021 (2010)
125. L. Xu, G. Pearlson, V.D. Calhoun, NeuroImage **44**, 777 (2009)
126. V. Calhoun, L. Wu, K. Kiehl, T. Eichele, G. Pearlson, Acta Neuropsychiatry **22**, 127 (2011)
127. V.D. Calhoun, T. Adali, IEEE Trans. Inf. Technol. Biomed. **13**, 711 (2009)
128. L. Astolfi, F. Cincotti, D. Mattia, S. Salinari, C. Babiloni, A. Basilisco, P.M. Rossini, L. Ding, Y. Ni, B. He, M.G. Marciani, F. Babiloni, Magn. Reson. Imaging **22**, 1457 (2004)
129. K. Hamandi, H.W. Powell, H. Laufs, M.R. Symms, G.J. Barker, G.J. Parker, L. Lemieux, J.S. Duncan, J. Neurol. Neurosurg. Psychiatry **79**, 594 (2008)
130. F. De Martino, G. Valente, A.W. de Borst, F. Esposito, A. Roebroeck, R. Goebel, E. Formisano, Magn. Reson. Imaging **28**, 1104 (2010)
131. T. Eichele, K. Specht, M. Moosmann, M.L. Jongsma, R.Q. Quiroga, H.E.A. Nordby, Proc. Natl. Acad. Sci. U. S. **102**, 17798 (2005)
132. N.M. Correa, T. Adali, Y. Li, V.D. Calhoun, Signal Process. Mag. **27**, 39 (2010)
133. N.M. Correa, T. Eichele, T. Adali, Y. Li, V.D. Calhoun, NeuroImage **50**, 1438 (2010)
134. N.M. Correa, Y. Li, T. Adali, V.D. Calhoun, J. Sel. Top. Signal Process. **2**, 998 (2008)
135. J. Sui, T. Adali, Q. Yu, J. Chen, V.D. Calhoun, J. Neurosci. Methods **204**, 68 (2012)
136. S. Dähne, F. Bieβmann, W. Samek, S. Haufe, D. Goltz, C. Gundlach, A. Villringer, S. Fazli, K.R. Müller, Proc. IEEE **103**, 1507 (2015)
137. A. Krishnan, L.J. Williams, A. McIntosh, H. Abdi, NeuroImage **56**, 455 (2015)
138. C.A. Rodriguez, B.M. Turner, , S.M. McClure, PLoS ONE **e90138**, 9 (2014)
139. B.A. Purcell, J.D. Schall, G.D. Logan, T.J. Palmeri, J. Neurosci. **32**(10), 3433 (2012)
140. B.A. Purcell, T.J. Palmeri, Relating accumulator model parameters and neural dynamics. J. Math. Psychol. **76**, 156–171 (2017)
141. P.W. Glimcher, Annu. Rev. Neurosci. **26**, 133 (2003)
142. J.D. Roitman, M.N. Shadlen, J. Neurosci. **22**, 9475 (2002)

143. M.N. Shadlen, R. Kiani, Neuron **80**(3), 791 (2013)
144. M. Mulder, L. Van Maanen, B. Forstmann, Neuroscience **277**, 872 (2014)
145. R. Ratcliff, J.N. Rouder, Psychol. Sci. **9**, 347 (1998)
146. B.U. Forstmann, G. de Hollander, L. van Maanen, A. Alkemade, M.C. Keuken, Nat. Rev. Neurosci. **18**, 57 (2016)
147. T. Ho, S. Brown, J. Serences, J. Neurosci. **29**, 8675 (2009)
148. A.K. Churchland, R. Kiani, M.N. Shadlen, Nat. Neurosci. **11**, 693 (2008)
149. C.A. K., D. J., Curr. Opin. Neurobiol. **22**, 1 (2012)
150. M.C. Keuken, L. Van Maanen, R. Bogacz, A. Schäfer, J. Neumann, R. Turner, B.U. Forstmann, Hum. Brain Mapp. **36**, 4041 (2015)
151. R. Bogacz, K. Gurney, Neural Comput. **19**, 442 (2007)
152. A. Alkemade, A. Schnitzler, B. Forstmann, Brain Struct. Funct. **220**, 3075 (2015)
153. K. Gurney, T.J. Prescott, P.A. Redgrave, Biol. Cybern. **84**, 401 (2001)
154. M.C. Keuken, C. Müller-Axt, R. Langner, S.B. Eickhoff, B.U. Forstmann, J. Neumann, Front. Hum. Neurosci. **8**, 445 (2014)
155. M.C. Keuken, H.B.M. Uylings, S. Geyer, A. Schäfer, R. Turner, B.U. Forstmann, Front. Neuroanat. **6** (2012)
156. L. Waldorp, I. Christoffels, V. van de Ven, NeuroImage **54**, 2695 (2011)
157. G. de Hollander, B.U. Forstmann, S.D. Brown, Biol. Psychiatry: Cogn. Neurosci. Neuroimaging **1**, 101 (2016)
158. M. Rosenblatt, *Principles of Neurodynamics* (Spartan Books, Washington, DC, 1961)
159. W.S. McCullogh, W. Pitts, Bull. Math. Biophys. **5**, 115 (1943)
160. M.L. Minsky, S.A. Papert, *Perceptrons* (The MIT Press, Cambridge, 1969)
161. S. Grossberg, in *Progress in Theoretical Biology*, vol. 5, ed. by R. Rosen, F. Snell (Academic, New York, 1978), pp. 233–374
162. J.A. Anderson, in *Basic Processes in Reading: Perception and Comprehension*, ed. by D. LaBerge, S.J. Samuels (Erlbaum, Hillsdale, 1977), pp. 27–90
163. D.E. Rumelhart, in *Attention and Performance*, ed. by S. Dornic (Erlbaum, Hillsdale, 1977), pp. 573–603
164. J.L. McClelland, D.E. Rumelhart, Psychol. Rev. **8**, 375 (1981)
165. D.E. Rumelhart, J.L. McClelland, Psychol. Rev. **89**, 60 (1982)
166. J. McClelland, D. Rumelhart, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 2: Psychological and Biological Models (MIT Press, Cambridge, 1986)
167. K. Tsetsos, M. Usher, J.L. McClelland, Front. Neurosci. **5**, 1 (2011)
168. R. Bogacz, E. Brown, J. Moehlis, P. Holmes, J.D. Cohen, Psychol. Rev. **113**, 700 (2006)
169. D. van Ravenzwaaij, H.L.J. van der Maas, E.J. Wagenmakers, Psychol. Rev. **119**, 201 (2012)
170. J. Gao, R. Tortell, J.L. McClelland, PLoS ONE **6**, 1 (2011)
171. R. Bogacz, M. Usher, J. Zhang, J.L. McClelland, Philos. Trans. R. Soc.: B. Biol. Sci. **362**, 1655 (2007)
172. B. Purcell, J. Schall, G. Logan, T. Palmeri, J. Neurosci. **32**, 3433 (2012)
173. K. Tsetsos, G. Gao, J.L. McClelland, M. Usher, Front. Neurosci. **6**, 1 (2012)
174. O. Ossmy, R. Moran, T. Pfeffer, K. Tsetsos, M. Usher, T.H. Donner, Curr. Biol. **23**, 981 (2013)
175. B.M. Turner, P.B. Sederberg, Psychon. Bull. Rev. **21**, 227 (2014)
176. B.M. Turner, P.B. Sederberg, J.L. McClelland, J. Math. Psychol. **72**, 191 (2016)
177. D.W. Massaro, J. Mem. Lang. **27**, 213 (1988)
178. R. Ratcliff, P.L. Smith, Psychol. Rev. **111**, 333 (2004)
179. K.F. Wong, X.J. Wang, J. Neurosci. **26**, 1314 (2006)
180. A. Umakantha, B. Purcell, T. Palmeri, Mapping between a spiking neural network model and the diffusion model of perceptual decision making (working title) (2015). Manuscript in preparation
181. A. Roxin, A. Ledberg, PLoS Comput. Biol. **4**, e1000046 (2008)
182. J. Busemeyer, J. Townsend, Math. Soc. Sci. **23**, 255 (1992)
183. J. Busemeyer, J. Townsend, Psychol. Rev. **100**, 432 (1993)
184. G. Woodman, M. Kang, K. Thompson, J. Schall, Psychol. Sci. **19**, 128 (2008)

185. J.P. O'Doherty, P. Dayan, K. Friston, H. Critchley, R.J. Dolan, Neuron **28**, 329 (2003)
186. J.P. O'Doherty, A. Hampton, H. Kim, Ann. N. Y. Acad. Sci. **1104**, 35 (2007)
187. J.P. Gläscher, O'Doherty, WIREs Cogn. Sci. **1**, 501 (2010)
188. A.N. Hampton, P. Bossaerts, J.P. O'Doherty, J. Neurosci. **26**, 8360–8367 (2006)
189. T.V. Wiecki, J. Poland, M.J. Frank, Clin. Psychol. Sci. **3** (2015)
190. R.A. Rescorla, A.R. Wagner, in *Classical Conditioning II: Current Research and Theory*, ed. by A.H. Black, W.F. Prokasy (Appleton Crofts, New York, 1972), pp. 64–99
191. T. Davis, B.C. Love, A.R. Preston, J. Exp. Psychol. Learn. Mem. Cogn. **38**(4), 821 (2012)
192. J. Ditterich, Front. Neurosci. **4**, 184 (2010)
193. D.L. Medin, M.M. Schaffer, Psychol. Rev. **85**, 207 (1978)
194. J.P. Minda, J.D. Smith, J. Exp. Psychol.: Learn. Mem. Cogn. **28**, 275 (2002)
195. S.R. Zaki, R.M. Nosofsky, R.D. Stanton, A. Cohen, J. Exp. Psychol.: Learn. Mem. Cogn. **29**, 1160 (2003)
196. M. Mulder, L. van Maanen, B.U. Forstmann, Neuroscience **277**, 872 (2014)
197. E.L. Mansfield, F. Karayanidis, S. Jamadar, A. Heathcote, B.U. Forstmann, J. Neurosci. **31**(41), 14688 (2011). https://doi.org/10.1523/JNEUROSCI.2390-11.2011
198. T. Behrens, M. Woolrich, M. Walton, M. Rushworth, Nat. Neurosci. **10**, 1214 (2007)
199. B.W. Brunton, M.M. Botvinick, C.D. Brody, Science **340**(6128), 95 (2013). https://doi.org/10.1126/science.1233912
200. J.C. Erlich, B.W. Brunton, C.A. Duan, T.D. Hanks, C.D. Brody, Elife **4** (2015). https://doi.org/10.7554/eLife.05457
201. T.D. Hanks, C.D. Kopec, B.W. Brunton, C.A. Duan, J.C. Erlich, C.D. Brody, Nature **520**(7546), 220 (2015). https://doi.org/10.1038/nature14066
202. M.F.S. Rushworth, R.B. Mars, C. Summerfield, Curr. Opin. Neurobiol. **19**(1), 75 (2009). https://doi.org/10.1016/j.conb.2009.02.005
203. T. Ho, S. Brown, L. van Maanen, B.U. Forstmann, E.J. Wagenmakers, J.T. Serences, J. Neurosci. **32**, 7992 (2012)
204. Y. Nagai, H.D. Critchley, E. Featherstone, P.B.C. Fenwick, M.R. Trimble, R.J. Dolan, Neuroimage **21**(4), 1232 (2004). https://doi.org/10.1016/j.neuroimage.2003.10.036
205. M.M. Plichta, I. Wolf, S. Hohmann, S. Baumeister, R. Boecker, A.J. Schwarz, M. Zangl, D. Mier, C. Diener, P. Meyer, N. Holz, M. Ruf, M.F. Gerchen, D. Bernal-Casas, V. Kolev, J. Yordanova, H. Flor, M. Laucht, T. Banaschewski, P. Kirsch, A. Meyer-Lindenberg, D. Brandeis, J. Neurosci. **33**(36), 14526 (2013). https://doi.org/10.1523/JNEUROSCI.0631-13.2013
206. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (Freeman, New York, 1982)
207. J.R. Anderson, *The Adaptive Character of Thought* (Psychology Press, 1990)
208. M. Oaksford, N. Chater, *Bayesian Rationality: The Probabilistic Approach to Human Reasoning* (Oxford University Press, Oxford, 2007)
209. J. Tenenbaum, C. Kemp, T. Griffiths, N. Goodman, Science **331**, 1279 (2011)
210. M. Jones, B. Love, Behav. Brain Sci. **34**, 169 (2011)
211. J. Busemeyer, J. Townsend, Z. Wang, A. Eidels, *Mathematical and Computational Models of Cognition* (Oxford University Press, 2013)
212. P. Dayan, L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (MIT Press, Cambridge, 2005)
213. M.N. Shadlen, W.T. Newsome, J. Neurophysiol. **86**, 1916 (2001)
214. S. Brown, A. Heathcote, Psychol. Rev. **112**, 117 (2005)
215. R. Ratcliff, F. Tuerlinckx, Psychon. Bull. Rev. **9**, 438 (2002)
216. F. Tuerlinckx, Behav. Res. Methods Instrum. Comput. **36**, 702 (2004)
217. J. Vandekerckhove, F. Tuerlinckx, Psychon. Bull. Rev. **14**, 1011 (2007)
218. D.J. Navarro, I.G. Fuss, J. Math. Psychol. **53**, 222 (2009)
219. B.M. Turner, P.B. Sederberg, J. Math. Psychol. **56**, 375 (2012)
220. B.M. Turner, T. Van Zandt, Psychometrika **79**, 185 (2014)
221. B.U. Forstmann, E.J. Wagenmakers, *An Introduction to Model-Based Cognitive Neuroscience* (Springer, New York, 2015)

Index

B. M. Turner et al., *Joint Models of Neural and Behavioral Data*,
Computational Approaches to Cognition and Perception,
https://doi.org/10.1007/978-3-030-03688-1

L