

Parameter Dependencies in an Accumulation-to-Threshold
Model of Simple Perceptual Decisions

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree
Master of Arts in the Graduate School of The Ohio State University

By

Vyacheslav Y. Nikitin, B.S.

Graduate Program in Psychology

The Ohio State University

2015

Master's Examination Committee:

Patricia Van Zandt, Advisor

Michael Edwards

Jay Myung

Paul De Boeck

© Copyright by

Vyacheslav Y. Nikitin

2015

Abstract

It is a common assumption in sequential sampling models of simple perceptual decisions that parameters are statistically independent across trials. This thesis addresses theoretical and empirical implications of assuming statistically dependent parameters. Three questions are answered: how to formulate flexible multivariate distributions of parameters of sequential sampling models, what are the predictive consequences of parameter dependencies for mean sample paths and joint distribution of responses and response times, and what correlation matrix is consistent with a benchmark dataset collected from a brightness discrimination task without explicit correlation manipulations.

The key to studying dependent parameters is a flexible framework of copulas that allow arbitrary combinations of dependence structures with marginal distributions. Adding correlations to a widely-used diffusion model shows that initial points and absorption times of mean sample paths can be strongly affected by correlations. Whereas the impact of correlation on the joint distribution of behavior is potentially strong adjustment of asymmetry in reaction time distributions of the two responses. Finally, in an experiment without explicit manipulation of correlations, the posterior distribution is consistent with small to moderate correlations between parameters.

Thus, under typical experimental conditions, the usual assumption of statistical independence is an adequate simplification of how parameters of simple decision making vary across trials.

This is dedicated to the crucible of science

Acknowledgments

I want to thank my adviser Patricia Van Zandt for advice, proofreading and invaluable computing resources that made this thesis possible. I am also grateful to my committee members - Jay Myung, Michael Edwards and Paul De Boeck - for provoking questions and insightful comments during the development of this thesis.

Vita

- 2010 B.S. Psychology
- 2011-2012 University Fellow,
The Ohio State University
- 2012-present Graduate Teaching Associate,
The Ohio State University

Fields of Study

Major Field: Psychology

Table of Contents

	Page
Abstract	ii
Dedication	iv
Acknowledgments	v
Vita	vi
List of Tables	ix
List of Figures	x
1. Introduction	1
1.1 Formal Modeling Approach	11
1.2 Data-Generating Experimental Tasks	18
1.3 Benchmark Behavioral Data	20
1.4 Cognitive Models	24
1.5 Bayesian Statistical Framework	27
1.5.1 Bayesian Models	28
1.5.2 MCMC Background	30
1.5.3 Metropolis-Hastings Algorithm	34
2. Motivating Questions	38
2.1 Modeling Parameter Dependencies	38
2.2 Predictions of Dependent Parameters	41
2.3 Correlation Structure in a Benchmark Dataset	43

3.	Models	45
3.1	Ratcliff Diffusion Model	46
3.2	Copula-based Multivariate Distributions	59
3.3	Generalized Decision Models	68
4.	Theoretical and Empirical Studies	74
4.1	Design and Parameter Settings	75
4.2	Study A1 - Mean Sample Paths	79
4.2.1	Mean Sample Paths Calculation	82
4.2.2	Results	85
4.2.3	Discussion	109
4.3	Study A2 - Response times and Responses	112
4.3.1	Joint Distribution Calculation	113
4.3.2	Results	115
4.3.3	Discussion	128
4.4	Study B - Benchmark Dataset Analysis	132
4.4.1	Outlier filtering	133
4.4.2	Bayesian Models	135
4.4.3	MCMC Sampler	139
4.4.4	Convergence Diagnostics	143
4.4.5	Characterizing Parameter Dependencies	145
4.4.6	Results	146
4.4.7	Discussion	156
4.5	Summary	161
	Bibliography	164
	Appendix A. Study A1 - Mean Sample Paths	174
	Appendix B. Study A2 - Response Times and Responses	191
	Appendix C. Study B - Benchmark Dataset Analysis	200

List of Tables

Table		Page
4.1	Weibull function parameters used to obtain predictions in study A. The values were reported by Vandekerckhove, Tuerlinckx, and Lee (2011).	76
4.2	Non-correlation parameters used to obtain predictions in study A. The values were reported by Vandekerckhove et al. (2011). Starting point values were transformed to decision bias values.	77
4.3	Correlation parameters used to obtain predictions in study A. The values were picked to range from low to high and represent three possible correlation patterns. For sample paths only column one matters, especially rows 4 - 6. Predicting response and response times depends on all the values.	78
4.4	Summary of the posterior distribution of the non-correlation parameters of the normal copula model for subject “kr”. Each parameter is summarized by a mean and lower/upper boundaries of its HDI. Note: ACC is accuracy condition and SPD is speed condition.	154
4.5	Summary of the posterior distribution of the correlation parameters of the normal copula model for all three subjects. Each parameter is summarized by a mean and lower/upper boundaries of its HDI. Note: ACC is accuracy condition and SPD is speed condition.	155

List of Figures

Figure	Page
1.1 Features of choice and reaction time under different experimental conditions. Top left: a subset of a trial-series showing across-trial dependencies, effect of instructions and extreme reaction times. Top right: shows a positive skew of reaction time and effect of brightness proportion on it, spanning from easy dark stimuli to ambiguous stimuli to easy bright stimuli. Bottom left: shows mean reaction time asymmetry with dark response during trials with dark stimuli faster than during light stimuli. Bottom right: shows mean reaction time asymmetry with light response during trials with light stimuli faster than during dark stimuli. Data was generated by participant “nh”.	23
3.1 The top two panels show samples from a Frank copula and a t copula. Bivariate distributions resulting from applying a t and a gamma transformation to copula samples are in the bottom panel. Parameter settings: $\theta_F = -15, \theta_t = -0.95, \omega = 2, \nu = 3, \lambda = 1.5, \alpha = 3, \beta = 6$. Parameters result in approximately the same Kendall’s measure of correlation.	65
4.1 Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = 0.15$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions. . .	87

- 4.2 Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = 0.5$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions. . . 88
- 4.3 Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = 0.85$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions. . . 89
- 4.4 Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = -0.15$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions. . . 90
- 4.5 Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = -0.5$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions. . . 91
- 4.6 Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = -0.85$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions. . . 92

4.7 Behavioral predictions for the brightness discrimination task under $[+ - -]$ pattern of correlations among $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Correlations increase magnitude from top to bottom. Left column and right column correspond to “accuracy” and “speed”, respectively. Within each subplot, the lines connect reaction time quantiles conditional on response, and show how behavior changes from high to intermediate luminance.	119
4.8 Behavioral predictions for the brightness discrimination task under $[+ + +]$ pattern of correlations among $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Correlations increase magnitude from top to bottom. Left column and right column correspond to “accuracy” and “speed”, respectively. Within each subplot, the lines connect reaction time quantiles conditional on response, and show how behavior changes from high to intermediate luminance.	127
4.9 Behavioral predictions for the brightness discrimination task under $[- - +]$ pattern of correlations among $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Correlations increase magnitude from top to bottom. Left column and right column correspond to “accuracy” and “speed”, respectively. Within each subplot, the lines connect reaction time quantiles conditional on response, and show how behavior changes from high to intermediate luminance.	129
4.10 Log-likelihood values corresponding to points in the MCMC chains for the normal copula model. Initial run of the chain leads to several increases in the value of log-likelihood followed by a sequence of values in a stable range. The pattern is consistent with successful convergence.	148
4.11 MCMC chains for $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$ without the burn-in iterations for the independent copula model. The stability of traces indicates convergence and rapid movement shows that AMH algorithm is appropriate for the model.	149
4.12 Values of Geweke statistic for all 23 parameters of the normal copula model. Geweke statistic is a difference in means of later minus the early part of a chain. All values are within two standard deviations indicating convergence to the equilibrium density.	150
4.13 Cross-correlations of all 23 parameters based on MCMC chains without the burn-in period. Several pairs of parameters show small to high correlations consistent with other sequential sampling models, and support using adaptive algorithms for efficient sampling.	152

Chapter 1: Introduction

Under the same environmental conditions humans, and other phylogenetically younger animals, show a limited number of alternative behaviors. Variability of behavior suggests that evolution of complex perceptual and cognitive functions presents animals with choices for any given situation. Given that behavior is organized successfully in light of choices, their nervous systems must have a mechanism for transforming competing choices into a decision. Thus, understanding how the brain organizes behavior of complex animals can be advanced by understanding decision mechanisms it employs and their relation to perceptual and motor processing.

Laboratory investigations of decision-making involving humans span a wide range of decision situations. Humans can be asked to make complex, multi-step decisions involved in selecting among health care policies (Peters, Hart, Tusler, & Fraenkel, 2014), gambles (Tversky & Kahneman, 1981), competitive strategies (Camerer, 2003), and to make simple one-shot decisions based on a single dimension of a stimulus such as presence of a visual pattern (Smith, 1995), relative brightness of stimuli (Ratcliff & Rouder, 1998), or motion direction (Ratcliff & McKoon, 2008). In this thesis, I concentrate on simple one-step decisions among two alternatives because resulting behavioral data can be obtained under rigorous control (Van Zandt & Townsend, 2013), the decision process is more amenable to mathematical modeling (Coombs,

1983; Lewandowsky & Farrell, 2011), and the uncovered principles may generalize to all forms of decisions (Boring, 1937; Heitz & Schall, 2013; Shadlen & Kiani, 2013).

An example of laboratory tasks I concentrate on is a lexical decision task (Wagenmakers, 2009). On each trial, a participant is shown a string of letters, and asked to quickly and accurately decide whether it is a word or not by pressing one of two buttons on a response box. Thus, the design of an experiment couples participant's decision with a simple movement (Van Zandt & Townsend, 2013), and enables easy collection of responses, response times and physiological measures that can provide substantial clues about the simple decision mechanism (Gold & Shadlen, 2007; Luce, 1986; Mulder, van Maanen, & Forstmann, 2014; Shadlen & Kiani, 2013).

The data bearing on psychological theory of simple decisions comes from both neural and behavioral studies. On one hand, the key findings from neural studies involve growth and decay shapes of firing rate functions of neurons interpreted as implementing competing decisions (Gold & Shadlen, 2007). On the other hand, behavioral data contributes choice probabilities and choice response time distributions, which have had the most effect on theory so far and are the primary focus of the thesis (Bogacz, 2007; Edwards, 1965; Luce, 1986; Mulder et al., 2014; Ratcliff & Rouder, 1998; Shadlen & Newsome, 1996; Smith & Ratcliff, 2004; Stone, 1960; Townsend & Ashby, 1983; Vickers, 1979). The primary features of behavioral data include positively skewed response time distributions, asymmetries between reaction times of different choices, speed-accuracy trade off and sequential relations between consecutive responses and response times (Craigmile, Peruggia, & Van Zandt, 2010; Ratcliff, 2014).

The currently dominant theoretical understanding of simple one-stage decisions is based on an analogy with sequential sampling of data for the purpose of hypothesis testing (Bogacz, Brown, Moehlis, Holmes, & Cohen, 2006; Stone, 1960; Vickers, 1979; Wald & Wolfowitz, 1948). One version of psychological theory consistent with sequential sampling postulates that on every trial a participant’s brain forms a variable representation of the decision-relevant stimulus feature. To make a decision, participants have a mechanism to sample noisy signals from the representation and accumulate them into separate representations standing for competing choices. A decision is formed when one of the choice representations, called evidence integrators, reaches a preset and stable threshold of evidence. Associated movement is organized downstream depending on which evidence integrator reached the threshold first.

There are several competing models that implement sequential sampling idea in a mathematical formalism (S. D. Brown & Heathcote, 2008; Pike, 1973; Ratcliff, 1978; Stone, 1960; Usher & McClelland, 2001). Assumptions of the models provide a precise description of how various parameters of processing combine to form simple decisions, parameters that are supposedly capturing higher-level properties of a task-evoked neural network, a distributed and interconnected set of neural populations underlying task performance (Cassey, Heathcote, & Brown, 2014; Cole, Bassett, Power, Braver, & Petersen, 2014; Gold & Shadlen, 2007; Heitz & Schall, 2013; Mulder et al., 2014). The models highlight time to encode evidence, rate of evidence uptake, evidence threshold, initial evidence and time to execute a motor response as important processing parameters in explaining patterns of behavior, but make different assumptions about how these parameters combine to generate choice behavior.

One way to categorize all the models is based on three types of assumptions that they make. The first kind of assumptions describe the number, interactions and dynamics of evidence integrators during a trial, potentially expressed as a system of stochastic processes (Smith, 2000). The second kind makes assumptions about other stages of processing involved in a trial, such as encoding of stimulus features and motion response, which frequently amounts to assigning an overall time constant that describes their duration (Ratcliff & Smith, 2004). Lastly, simple decision models incorporate distributional assumptions that describe how parameters vary across trials to capture variability in the processing system (Jones & Dzhafarov, 2014b; Ratcliff & Smith, 2004).

Mathematical assumptions of the competing models make different theoretical claims about processing. Testable predictions about responses and response time distributions, and neurophysiology if additional linking assumptions are made, can be deduced from the theoretical assumptions (Coombs, 1983; Luce, 1986; Ratcliff, Cherian, & Segraves, 2003; Ratcliff, Hasegawa, Hasegawa, Smith, & Segraves, 2007; Vickers, 1979). A set of models, widely used for providing process-based interpretation of neural signals and behavior (Donkin, Brown, Heathcote, & Wagenmakers, 2011; Heathcote & Hayes, 2012; Ratcliff & McKoon, 2008), have been developed that account, with high accuracy, for a large number of patterns in behavioral data including response time distributions, accuracy, and relations between response times and accuracy, across a variety of manipulations and paradigms (S. D. Brown & Heathcote, 2008; Ratcliff & Tuerlinckx, 2002; Usher & McClelland, 2001). There is also some qualitative success in describing neural recordings (Ratcliff et al., 2003, 2011, 2007). The problems with these models are that their predictive patterns mimic each

other statistically with common sample sizes while making different theoretical claims (Donkin et al., 2011; Ratcliff & Smith, 2004; Van Zandt, Colonius, & Proctor, 2000) and being unable to account for the whole pattern of neural and behavioral data. At the same time, far from everything is understood about simple decisions (Heitz & Schall, 2013; Zandbelt, Purcell, Palmeri, Logan, & Schall, 2014).

One of the ways to push toward a better understanding of processing underlying choice behavior is to incorporate additional theoretical principles into promising models, without changing them fundamentally, and attempt to account for novel kind of data or unexplained patterns in the same kind of data (Newall, 1973). This approach was taken in this thesis, but other strategies like adaptive experimentation, that attempts to squeeze maximal information from the same tasks/data (Kim, Pitt, Lu, Steyvers, & Myung, 2014), and developing neuropsychological models, that attempts to broaden the kinds of data that models interact with (Ashby & Helie, 2011), are also viable. Looking at the history of the sequential sampling field, one of the largest differences between currently popular models and their predecessors is addition of assumptions about variability of parameters across trials (Laming, 1968; Ratcliff & Smith, 2004; Vickers, 1979). Variability assumptions express the idea that processing is not static across trials within a condition. Starting evidence may include evidence remaining from the previous trial or movement time may be faster if the same stimulus repeats. Expressed in terms of probability distributions, variability in parameters can account for additional data involving asymmetries in relations between speed and accuracy of processing, which the old models missed (Luce, 1986; Ratcliff & Smith, 2004).

While some gains in understanding have already been made from incorporating trial-to-trial variability in parameters into models, I suggest that further gains can be made by better characterizing this sort of variability. Consider that currently popular models differ in assuming what parameters vary and what their distributional forms are (Jones & Dzhafarov, 2014b), but they all commit to two questionable claims: on each trial a parameter is set to a value independent of the previous trials and all parameters are mutually independent (S. D. Brown & Heathcote, 2008; Ratcliff & Tuerlinckx, 2002; Usher & McClelland, 2001).

The first assumption is challenged by the typically present auto-correlation in response times and responses (Jones, Curran, Mozer, & Wilder, 2013; Peruggia, Van Zandt, & Chen, 2002). Behavioral data suggests that, say, the threshold of a participant on one trial depends on the threshold on the previous several trials. One line of recent model development incorporated the principle of auto-correlated parameters through additional processes including learning of stimulus probabilities, error-correction and residual evidence (S. D. Brown, Marley, Donkin, & Heathcote, 2008; Cho et al., 2002; Gao, Wong-Lin, Holmes, Simen, & Cohen, 2009; Goldfarb, Wong-Lin, Schwemmer, Leonard, & Holmes, 2012; Wagenmakers, Farrell, & Ratcliff, 2004). Depending on the model, starting evidence, rate of evidence uptake and evidence thresholds were functions of values on previous trials and parameters of additional processes. Such modifications enabled models to predict a variety of sequential effects and improve understanding of decision process dynamics.

However, the other questionable assumption, the issue of mutual independence of processing parameters, has not been examined. An alternative to independence is that

processing parameters vary in systematic ways from trial to trial. Understanding relations between parameters characterizing particular processes, like decision formation, and relations of parameters across processes should increase understanding of each process as well as the whole processing architecture. The increase in understanding would come because relations between process parameters are different than relations between experimental variables and process parameters, relation between process parameters and behavioral variables, and relations between experimental variables and behavioral variables. In line with these distinctions, several sources of evidence provide support for existence of statistical dependencies among parameters that indicate a coherent internal structure.

One form of evidence comes from model-based analyses of response time distributions under block-wise manipulation of speed-accuracy instructions. Using a couple of prominent sequential sampling models, researchers showed that the best explanation of accuracy emphasis is based on simultaneous rise in the rate of evidence accumulation and evidence threshold (Rae, Heathcote, Donkin, Averell, & Brown, 2014; Vandekerckhove & Tuerlinckx, 2007; Vandekerckhove et al., 2011), and sometimes also increase in non-decision time (Voss, Rothermund, & Voss, 2004; Zhang & Rowe, 2014). In further support of change in non-decision time, a purely empirical study, using measurements of lateralized readiness potential with electroencephalograph, demonstrated that under speed instructions the time interval between onset of motor processing and motor response decreases (Rinkenauer, Osman, Ulrich, Müller-Gethmann, & Mattes, 2004). Both kinds of studies suggest that a control process that sets a speed-accuracy regime, as manipulated by instructions, can modulate activity in two or more parts of a neural network underlying task performance and its effects

are picked up by the models' parameters (Cole et al., 2014; Hare, Schultz, Camerer, O'Doherty, & Rangel, 2011; Heitz & Schall, 2013; Turner, Forstmann, et al., 2013; Wang, 2008).

Another block-wise manipulation indicating systematic trade-offs among parameters is due to practice. Using a sequential sampling model to analyze 10,000 trials of the lexical decision task, researchers found a coordinated adjustment of processing parameters with practice (Dutilh, Vandekerckhove, Tuerlinckx, & Wagenmakers, 2009). The non-decision time decreased, rate of evidence uptake increased, evidence threshold decreased and initial evidence became less biased with practice. In a similar model-based study, but based on a motion discrimination task, participants showed decrease in the evidence threshold and increase in the rate of evidence uptake over 4032 trials (Zhang & Rowe, 2014). Overall, practice effects are also consistent with a principle that decision and non-decision parameters change in a dependent manner.

On a trial-by-trial level, the phenomenon of post-error slowing also suggests dependencies among parameters as a feature of sequential decision-making (Rabbitt, 1979). Post-error slowing refers to the phenomenon that, when a participant makes a mistake, response time on the next trial slows down. Explanations of this phenomenon have usually evoked only an increase in the threshold, but a model-based analysis showed that rate of evidence uptake, threshold and non-decision time may all be affected (Dutilh, Forstmann, Vandekerckhove, & Wagenmakers, 2013). Dutilh et al. (2013) examined old and young adults with lexical decision and motion discrimination tasks. Old adults showed an increase in evidence threshold, decrease in rate of evidence uptake and increase in non-decision time across both tasks. Young adults showed an increase in non-decision time in the motion discrimination task, but in the

lexical decision task they increased evidence thresholds and increased non-decision time.

Another effect that is consistent with dependent parameters is based on effects of prior stimuli on response times. Goldfarb et al. (2012) collected behavioral data for a task requiring discrimination between upper and lower case letter Os. They analyzed mean response times, classified by correct and error response, on the second trial of two-stimulus sequences consisting of all possible combinations of repetitions and alternations. Conditioned on stimulus sequence, participants showed trade-offs between correct and error response times: faster response time on error trials corresponded to slower response times on correct trials, or vice versa. A model with trial-varying initial evidence captured the qualitative pattern, but systematically missed response times for error trials. Given the mechanics of sequential sampling models, the misses may have been due to not accounting for adjustments in thresholds or rate of information uptake that shift simultaneously with initial evidence.

Simultaneous adjustment in parameters has also served to account for sequential effects Ratcliff, Van Zandt, and McKoon (1999). Participants were given a numerosity task where clusters of asterisks need to be discriminated as large or small. Some participants showed elevated probabilities and speed when repeating the previous response rather than altering the response. Explanation of both the response probability and response time was successful only when initial evidence and rate of evidence of uptake were allowed to adjust simultaneously.

Assuming that there are stable relations among parameters, as suggested by behavioral data, an increase in our knowledge about how simple choice behavior is

organized can come from exploring these relations. Without facts of brain or behavior providing sufficient constraints on such relations, a simple principle of processing dynamics underlying visually-guided choice behavior may be adopted to guide initial theoretical and empirical studies: From trial to trial, and across conditions, psychological parameters identified by sequential sampling models vary systematically in a dependent manner. The suggested “parameter dependence principle” is in qualitative contrast with the usually assumed independence of processing parameters (Ratcliff & Smith, 2004).

Given the sharp contrast between dependent and independent parameters, the goals motivating this thesis were to generalize one of the currently popular sequential sampling models of decision and use it to explore relations among parameters by comparing it to a model with independent parameters. The approach I took involves adding a correlation structure that can describe correlations between parameters without making a theoretical claim as to how correlations arise. Using a set of models with and without dependencies, the thesis addresses two general questions: What parameter dependencies imply for predictions of the models and what does behavioral data imply for dependencies among parameters of the same models? Addressing these questions through theoretical and empirical comparisons of the standard version and a generalized version of a sequential sampling model can give better insight into processing underlying choice behavior.

In the rest of this introduction chapter, I lay out reasons for taking formal modeling approach to test the parameter dependence principle and elaborate the structure of experiments generating simple choice data. As an example of a typical experiment, I present a benchmark dataset, and important response time and accuracy features

that form the focus of comparisons between models. Then, I present standard sequential sampling framework and then introduce a method of constructing multivariate distributions that is useful for developing a generalized model of decision making. Finally, I describe a statistical framework grounded in Bayesian theory that will be used to estimate the correlation structure from behavioral data.

1.1 Formal Modeling Approach

Psychology has as one of its chief aims understanding of the cognitive architecture (mind) that generates the rich repertoire of human behavior and is sensitive to many environmental variables (Anderson, 2009; Anderson et al., 2004). More specifically, a psychologist wants to learn about the number, spatial and temporal arrangement, properties and interactions of elementary processes that organize behavior. A fundamental obstacle to satisfying psychologist's aspirations is that cognition is not directly observable, even though it may consist of observable, measurable matter, i.e. neural populations Hebb (1980). Cognition is analogous to genes that are made up of nucleic acids, but whose location on DNA molecule has to be inferred from a variety of data. In case of cognition, we can observe a person engage in different behaviors under the same conditions, but the process of decision is latent. Even if we had the whole wiring diagram of the person's brain and spiking patterns under a variety of situations, it still would not be obvious where, how and when action selection takes place or its dependence on perception and attention (J. W. Brown, 2014). So, how to go about understanding cognition?

Concentrating on behavioral methodology, the study of human cognition can be considered as an instance of a grey box problem (Ljung, 1999, 2010). Working on a

grey box problem, a researcher has direct information about the system's input and output, and some information about internal processes transforming inputs into outputs. The original problem can then be formulated as one of understanding internal processes from known combinations of inputs, outputs, and process-related measures. For psychology, the problem is uncovering the cognitive architecture from information about overt behavior taking place in some environment while potentially taking measures of the nervous system.

A powerful approach to providing an approximate solution to the grey box problem is construction of competing mathematical models that can be tested against data using statistical methods (Casella & Berger, 2002; Gelman et al., 2014; Ljung, 1999, 2010). Each model would represent a mechanistic account of how inputs are transformed into outputs, with neurophysiological recordings potentially acting as mediating variables (Purcell et al., 2010). A model that best balances parsimony, fit to known empirical regularities and interpretability can be taken as a formal instantiation of the best theoretical principles characterizing internal processes and provide further guidance in experimental work. This approach fits well with most psychological research because it is often easy to come up with several, categorically different conjectures about underlying cognitive processes (Townsend & Ashby, 1983; Vickers, 1979) and there is plenty of data to test them (Gold & Shadlen, 2007; Luce, 1986).

The primary benefit of developing mathematical models rather than verbal descriptions is that they force coherency on theoretical assumptions, enable deriving exact conclusions from assumptions and improve falsifiability of proposed principles through quantitative statistical testing (Busemeyer & Diederich, 2010; Coombs, 1983;

Kim et al., 2014; Lee & Wagenmakers, 2013; Lewandowsky & Farrell, 2011). For instance, a model of memory processes that can predict a power relation between time and memory retention is easier to falsify than a verbal account that only makes ordinal predictions (Cavagnaro, Myung, & Pitt, 2013). Ultimately, the promise of mathematical models is to bring psychology closer to sensible theoretical principles that characterize elementary processes and how they evolve over time, under different experimental conditions for different individuals.

Before presenting an example of the formal modeling framework, I describe notation used for the rest of the thesis. I use lower-case non-bolded symbols for scalars, lower-case bolded symbols for column vectors, and upper-case bolded symbols for matrices. I discriminate between random variables and their realized values by qualifying my statements when necessary, but otherwise their case will reflect their mathematical nature (i.e. scalar or vector), as for other symbols. Observables, response variables or covariates, are represented with Roman letters while parameters are represented with Greek letters.

A formal relationship between response variables $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^k$, covariates $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^l$ and parameters $\theta \in \Theta \subseteq \mathbb{R}^m$, will be stated as a parametric statistical model $M = \{F(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta})\}$, where F is a probability distribution function of \mathbf{y} conditioned on \mathbf{x} . In words, a parametric model is a collection of probability distributions indexed by a finite-dimensional parameter that provides a full description of variability in \mathbf{y} . If a probability density (mass) function f exists, then a model expression can be stated in terms of f . Also, covariates may not always be present, so \mathbf{x} may be dropped. Lastly, a short-hand notation $\mathbf{y} \sim F(\mathbf{y} | \boldsymbol{\theta})$ may be used, where \sim stands for “distributed as”.

As an example, consider a random vector $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$, where T is the transpose operator. If we assume that $\mathbf{y} \in \mathbb{R}^n$ varies according to a multivariate normal probability density f , parameterized with a vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and a positive definite matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$, then the parametric model of \mathbf{y} is a collection of probability densities

$$M = \left\{ f(\mathbf{y} | \boldsymbol{\mu}, \sigma^2) = \det(\sqrt{2\pi}\boldsymbol{\Sigma})^{-1} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma} (\mathbf{y} - \boldsymbol{\mu})\right) \right\}, \quad (1.1)$$

where $\det(\cdot)$ is a determinant operator.

With notation fixed, next I briefly present the formalism of the standard signal detection theory as an example of approach taken to exploring dependencies among parameters during simple choice behavior (Macmillan & Creelman, 2004). The original phenomenon motivating development of signal detection models was the human ability to make accurate decisions with distorted sensory stimuli. This phenomenon can be studied experimentally using a computerized task. For example, during the auditory signal detection task an observer is presented with a sequence of stimuli drawn either from a *Signal* distribution representing a tone combined with white noise, or a *Noise* distribution representing white noise. On each trial, an observer has to respond “Yes” if he or she decides the stimulus belongs to the *Signal* distribution and “No” otherwise. Under these conditions, a researcher knows to which class each presented stimulus belongs and observes a sequences of responses, typically summarized as hit rate (proportion of true positive responses) and false alarm rate (proportion of false positive responses). The psychological problem is to characterize cognitive processes involved in transforming stimulus information into responses from the generated data.

Signal detection models are statistical models that decompose hit and false alarm rates into cognitively interpretable parameters characterizing stimulus representation

and decision process without specifying mechanistic details (Lee & Wagenmakers, 2013; Macmillan & Creelman, 2004). The basic idea is that behavior in a signal detection task can be explained by a theory that proposes the following principles:

1. Presentation of a stimulus causes the observer to form a representation of the stimulus “strength.”
2. An observer is able to establish a decision threshold to which this perceived strength can be compared.
3. If the perceived strength is greater than this threshold then the observer responds “Yes,” otherwise he or she responds “No.”
4. The perceived strength varies from trial to trial such that its statistical structure (mean and variance) is determined by the class of the presented stimulus, where the strengths of Noise stimuli tend to be perceived as being below the threshold and the strengths of Signal stimuli tend to be perceived as being above.

Formalization of these principles can be done using the language of probability theory (Casella & Berger, 2002). Let a random variable $c \in \{0, 1\}$ represent a stimulus randomly drawn either from the *Noise* class ($c = 0$) or the *Signal* class ($c = 1$). The probability mass function of c is determined by the experimenter. In response to the sampled stimulus, an internal representation of strength value, represented by a random variable $x \in \mathbb{R}$, is generated. We commonly assume that x conditioned on the stimulus class is normally distributed with class-specific mean $\mu_i \in \mathbb{R}$ and variance

$\sigma_i^2 > 0$ parameters, so that

$$\begin{aligned} x \mid c = 1 &\sim \mathcal{N}(\mu_s, \sigma_s^2) \text{ and} \\ x \mid c = 0 &\sim \mathcal{N}(\mu_n, \sigma_n^2), \end{aligned} \tag{1.2}$$

where $|$ stands for “conditioned on”. Finally, let $\tau \in \mathbb{R}$ represent the position of a threshold on a strength dimension. Then a random variable

$$r = \begin{cases} 1 & \text{if } x \geq \tau \\ 0 & \text{if } x < \tau \end{cases} \tag{1.3}$$

represents a “Yes” ($r = 1$) or a “No” ($r = 0$) response.

As written, this version of the model has five free parameters $(\mu_n, \mu_s, \sigma_n, \sigma_s, \tau)$, but because normal distributions form a location-scale family, where the mean and standard deviation are the location and scale parameters, respectively, the means and standard deviations are identifiable only relative to each other. Therefore, we set $\mu_n = 0$ and $\sigma_n = 1$ without loss of generality. The other three parameters remain free, and quantify representational and decision features of signal detection.

In actual analysis of behavioral data, it is common to use the derived parameter $d' = (\mu_s - \mu_n)/\sqrt{(\sigma_s^2 + \sigma_n^2)/2} = \mu_s/\sqrt{\sigma_s^2/2}$, which is psychologically interpreted as quantifying how well the encoding process separates *Noise* from *Signal* stimuli (Macmillan & Creelman, 2004). The threshold τ represents the decision criterion and the ratio of variances $\sigma_n^2/\sigma_s^2 = 1/\sigma_s^2$ between *Noise* and *Signal* distributions characterizes relative noise in representations. These three parameters govern the predicted pattern of responses.

To begin testing a specified cognitive model one needs to derive, or be able to simulate, predictions for behavioral data. In the signal detection case, the random

mechanism specified above, and an additional assumption that all responses during an experiment are mutually independent, allow for an analytical derivation of a probability mass function parameterized by the three free parameters. The derived model for the number of hits $y_h \in \{0, 1, \dots, N_s\}$ and the number of false alarms $y_{fa} \in \{0, 1, \dots, N_n\}$ for a given participant presented with N_s *Signal* stimuli and N_n *Noise* stimuli is a product of two binomial probability mass functions. The joint probability of y_h and y_n , conditioned on the parameters, is

$$f(y_h, y_{fa} | p_h, p_{fa}, N_s, N_n) = \\ \binom{N_s}{y_h} p_h^{y_h} (1 - p_h)^{N_s - y_h} \binom{N_n}{y_{fa}} p_{fa}^{y_{fa}} (1 - p_{fa})^{N_n - y_{fa}}, \quad (1.4)$$

with the probabilities p_h and p_{fa} being the probabilities that $r = 1$ conditioned on the stimulus class. The hit and false alarm probabilities are in turn functions of the representational and decisional parameters μ_s , τ and σ_s , such that

$$p_h = P\{r = 1 | c = 1\} = \Phi\left(\frac{\mu_s - \tau}{\sigma_s}\right) \text{ and} \\ p_{fa} = P\{r = 1 | c = 0\} = \Phi(-\tau), \quad (1.5)$$

where $\Phi(\cdot)$ is the standard normal distribution function and $P\{\cdot | \cdot\}$ is a conditional probability.

When a statistical model relating cognitive parameters to data is specified for a given theory, one can use a rich body of statistical methods to explore its predictions via simulations, to estimate its free parameters from data, and to compare and select from competing models of cognitive processing involved in a given task (Berger, 1997; Casella & Berger, 2002; Gelman et al., 2014). Connecting the signal detection model to performance data enables a researcher to test underlying principles and make a range of inferences about cognitive processes from behavior. For instance, it is

possible to test goodness of fit of a model with $\sigma_s^2 = 1$ against a model where σ_s^2 is a free parameter to determine the noise properties of the stimulus representation. However, the range of possible inferences is limited, so the specified model cannot say anything about (for example) formation and adaptation of the representations to dynamic stimulus stream or motor processing. It would require a new model or expanded model to address additional questions about processes involved in a signal detection task.

The example of a standard signal detection model shows how experimental variables, psychological parameters and behavior can be tied together using a mathematical model. With a mathematical model, theory of signal detection can be better tested and behavioral data can be interpreted in a psychologically interesting way. I carry over these analytical advantages to my studies of simple decision making described below. In the next section, I turn back to the problem of simple decision making and set the stage for the later model development by elaborating on the structure of experimental tasks used to obtain theory-pertinent data.

1.2 Data-Generating Experimental Tasks

To obtain input-output and physiological data informative of processes underlying simple choice behavior, we need a laboratory task that recruits processes of interest. I define simple choice behaviors to be a class of behaviors that rely on a single deliberation stage to form a discrete decision about an objectively determinable state of a noisy stimulus pattern during at most a four second window, starting with stimulus presentation and ending with movement registration. A large collection of tasks can evoke simple choice behaviors and can be classified in terms of five dimensions: the

sensory modality, the number of choices, the number of stimulus attributes relevant to a decision, whether a participant or an experimenter controls initiation of responses and response modality. In the thesis, I concentrate on participant-controlled two-choice tasks, where finger movements are driven by a single attribute of a visual stimulus. For short, I call them two-choice tasks.

Two-choice tasks, have been the predominant kind for obtaining empirical regularities, especially behavioral data, and developing theoretical models of simple decision making (Laming, 1968; Luce, 1986; Ratcliff & McKoon, 2008; Ratcliff & Smith, 2004; Vickers, 1979; Wagenmakers, 2009). An example of a two-choice task is a numerosity task (Ratcliff, Love, Thompson, & Opfer, 2012). On a given trial, two clouds of points, consisting anywhere from a few dots to a hundred, are presented simultaneously on opposite sides of a computer screen. The relative number of points determines uncertainty about which cloud is more numerous. An observer has to make a decision, as accurately and as fast as she can, according to a rule that maps stimulus identity onto movements, such as “press the button on a keyboard corresponding to the cloud with the larger number of points”. Accuracy and response time are measured using a computer. Simultaneous recording of fMRI, EEG or neuron spiking can also be collected.

Across trials of a two-choice task, several experimental variables are usually manipulated that produce recurrent empirical regularities in responses and response times (Luce, 1986), and presumably physiological variables. Continuing with the numerosity task, a researcher could vary relative number of points, instructions emphasizing either speed or accuracy, probability of the “large” cloud appearing on the left side

and probability of transitions between “large” and “small”. All of these manipulations tend to affect both accuracy and speed of responses, but in different ways, providing non-trivial constraints to test proposed models (Townsend & Ashby, 1983; Vickers, 1979).

The combination of experimental variables and response variables generated by two-choice tasks make up the input-output data that can be used to learn about cognitive processing. In the next section, I present a benchmark dataset generated by a brightness discrimination task that demonstrates the kind of behavioral patterns models need to be consistent with.

1.3 Benchmark Behavioral Data

Two-choice tasks are simple, but under different experimental manipulations they generate many informative patterns in neural and behavioral data. In my studies of parameter dependencies, I used the Ratcliff and Rouder (1998) brightness discrimination data as a source of behavioral data. No physiological data was collected during this experiment, however a similar task has been done with monkeys with implanted electrodes (Ratcliff et al., 2011). The Ratcliff and Rouder (1998) dataset has become somewhat of a benchmark for testing new models because it has a large number of observations, and demonstrates many of the recurrent accuracy and response time patterns (S. D. Brown & Heathcote, 2008; Vandekerckhove & Tuerlinckx, 2008; Vandekerckhove et al., 2011; Verdonck & Tuerlinckx, 2014).

Ratcliff and Rouder (1998) ran a brightness discrimination task with three participants. Each participant had one 35 min practice session and ten 35 min experimental sessions. Each session consisted of eight blocks of 102 trials, for a total of 8160 trials.

Each trial began with participants seeing a gray screen for 500 ms. Then, an array of white and black pixels appeared at the center. Participants had to decide whether the array came from a “high” or “low” brightness distribution. They responded as soon as they could by pressing one of two buttons with the corresponding finger. Upon a response, the stimulus would change back to the gray screen for a duration of 300 ms. During the final 300 ms, a feedback message was displayed indicating whether the decision was correct or not. Both responses and response times were recorded upon completion of a trial.

During each session, two experimental variables were manipulated. First, the ratio of white to black pixels, representing stimulus difficulty, was manipulated across trials. On each trial, a ratio was drawn from one of two symmetric, unimodal probability mass functions spanning over 33 levels of brightness and overlapping to ensure imperfect accuracy. Second, instructions changed every 204 trials to direct participants to either maximize speed or maximize accuracy.

Overall, with a simplifying assumption that practice effects are negligible after the training session, the experiment generated a data matrix \mathbf{D} with 24,480 rows observations, and five columns of variables. Among the five variables, response times and choices will be treated as response variables, and participant, brightness level and instructions will be treated as covariates.

In the rest of this section, I examine patterns in response variables under different values of covariates. Figure 1.1 provides a summary of the main patterns found in behavioral data in two-choice tasks based on participant “nh”, with other participants having similar patterns (Ratcliff & Rouder, 1998). Upper right subplot shows reaction times under different brightness proportions and reveals many features.

One important feature of behavioral data is large variation in response times (Luce, 1986). For example, under maximal brightness levels reaction times vary roughly from 250 - 1000 ms and under neutral brightness levels the reaction times can exceed 4000 ms. The shape of variation is positively skewed and depends greatly on the brightness proportion. The effect of brightness, however, appears to be symmetric, with really bright and dark stimuli resulting in similar performance. Another central feature of behavioral data is its asymmetry between responses consistent with stimulus (“correct”) and responses inconsistent with stimulus (“error”) (Swensson, 1972). Such asymmetries are shown in the lower two subplots that are conditioned on response “dark” and different instructions. The left subplot corresponds to the accuracy emphasis and shows that mean reaction times across brightness proportions when stimulus agrees with the response tend to be faster than the inconsistent responses. The right subplot, conditioned on the speed emphasis, shows the opposite trend; the mean reaction times across brightness proportions when stimulus agrees with the response are slower than the inconsistent responses.

In addition to the dependency of reaction time and choice, the bottom subplots also show that speed is positively related to choice frequencies when stimulus quality changes. As brightness proportion decreases, meaning that stimulus gets darker, “dark” responses become faster and more likely to occur. In contrast, the upper left subplot shows a trade-off between speed and response proportion under different speed-accuracy instructions (Heitz, 2014; Luce, 1986; Vickers, 1979). The batches of slower reaction times correspond to the “accuracy” instruction and batches of faster reaction times are due to “speed” instruction.

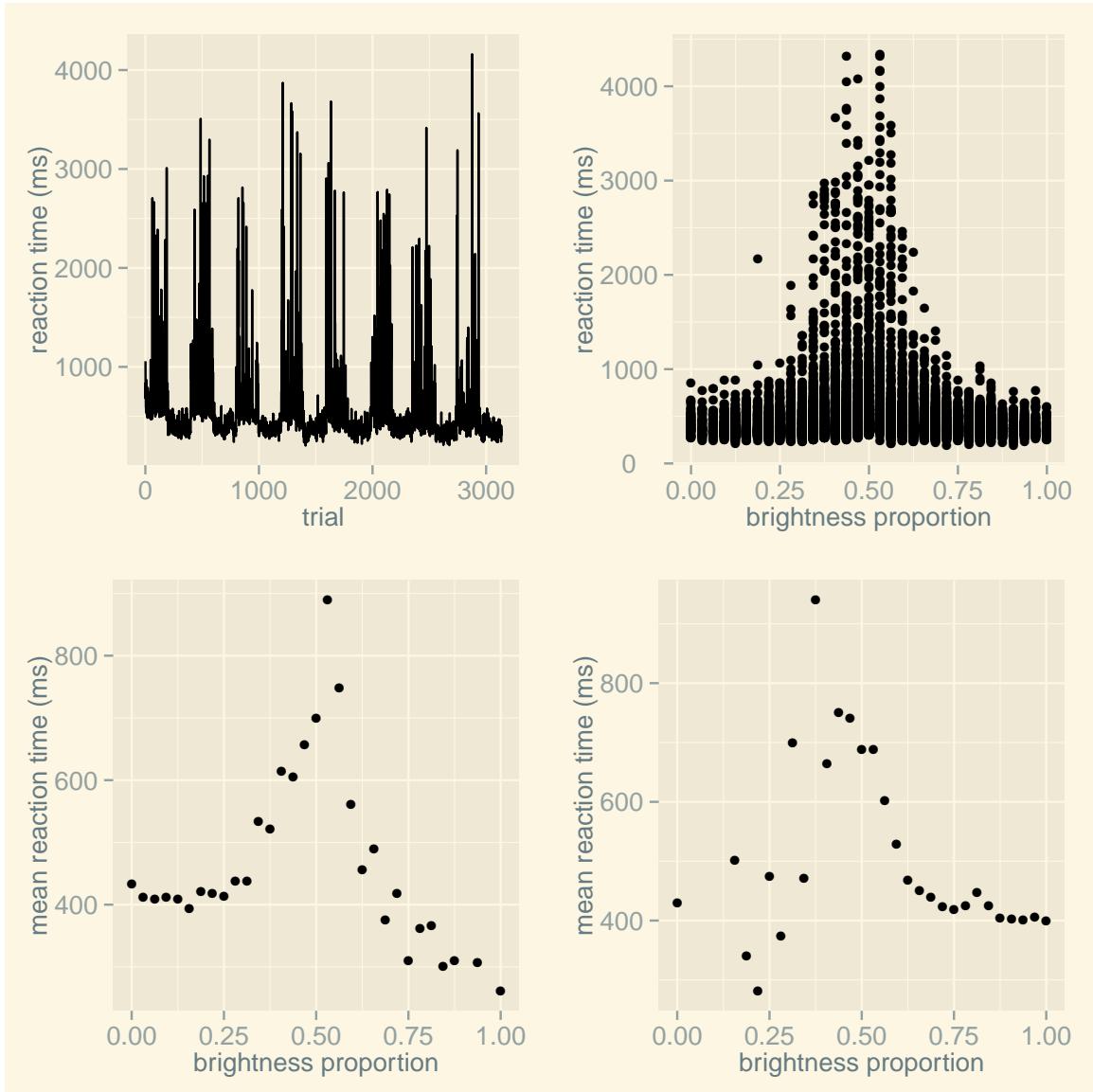


Figure 1.1: Features of choice and reaction time under different experimental conditions. Top left: a subset of a trial-series showing across-trial dependencies, effect of instructions and extreme reaction times. Top right: shows a positive skew of reaction time and effect of brightness proportion on it, spanning from easy dark stimuli to ambiguous stimuli to easy bright stimuli. Bottom left: shows mean reaction time asymmetry with dark response during trials with dark stimuli faster than during light stimuli. Bottom right: shows mean reaction time asymmetry with light response during trials with light stimuli faster than during dark stimuli. Data was generated by participant “nh”.

The upper left subplot also shows extreme reaction times. There are occasional spikes in reaction time reaching 3 - 4 seconds. Relative to most responses, these have extreme duration. Presence of extreme observations may indicate genuine response times from tails of the distribution carrying information about processing of interest, but it is not completely clear (Craigmile et al., 2010).

The complete pattern of data features - amount of variation in response times, skew of their probability densities, relation between correct and error response times, relation between response time and accuracy, speed-accuracy trade-off, extreme response times - provides strong constraints for developing and testing models of choice behavior. In the next section, I discuss a prominent theoretical framework for understanding simple choice behavior and how to implement it in testable models.

1.4 Cognitive Models

A common approach to conceptualizing cognitive processing underlying two-choice tasks is to decompose them into the three processes of (1) encoding of stimulus features, (2) decision based on a relevant stimulus feature and (3) motor execution of the decision (Luce, 1986). A viable cognitive process model for these tasks must be able to predict patterns in performance data as discussed in the previous section. A delicate balance has to be struck between what processing details to include and what details to exclude from models because it will determine their generalizability across two-choice tasks. For instance, including details about how lexical information is accessed in memory to direct the decision will make a model inapplicable to brightness discrimination. The dominant approach to this balancing problem is the sequential sampling framework.

A fundamental theoretical assumption of the sequential sampling framework is that the same general decision process operates across all two-choice tasks. Then, a generalizable model should provide a detailed account of the decision process while making simplifying assumptions about the stimulus encoding and motor processing, like assuming a constant non-decision time. Over the last 50 years or so, many models of this type have been developed (S. D. Brown & Heathcote, 2008; Ratcliff, 1978; Ratcliff & Tuerlinckx, 2002; Smith, 1995; Stone, 1960; Usher & McClelland, 2001; Vickers, 1979). Some of the most successful ones can explain many patterns in the response time and accuracy data under a variety of experimental conditions and tasks (Ratcliff & Smith, 2004; Wagenmakers, 2009), and connected to neural data (Purcell et al., 2010; Ratcliff et al., 2011).

The underlying principle of sequential sampling models is that when an observer is tasked with selecting a correct response in limited time from a noisy stimulus, he or she accumulates samples of information from the relevant stimulus attribute representation until one of two decision thresholds is reached (Bogacz et al., 2006; Luce, 1986; Townsend & Ashby, 1983; Vickers, 1979). The structure of thresholded accumulation unites concepts of decision bias, caution and information quality under a single mechanism. Differential levels of initial information produce bias, thresholds control caution and rate of information uptake reflects information quality. Combinations of these variables in thresholded accumulation also naturally predict decisions and response times. The time taken to reach a threshold and the crossed threshold determine decision time and decision. The sequential sampling principle is powerful, but to rigorously connect it to behavioral data we need a mathematical model.

Given the sequential sampling principle, any concrete model of simple decisions makes detailed assumptions about the nature of sampled information, temporal structure of sampling, stopping rule and sources of randomness (Bogacz et al., 2006; Ratcliff & Smith, 2004; Teodorescu & Usher, 2013; Vickers, 1979). For example, sampled information can consist of discrete or continuous packets. The size of packets may be fixed or variable. The accumulation of information packets may happen at fixed or variable discrete time points, or continuously. The stopping rule may be absolute, in which case accumulation stops when evidence in favor of one of the choices reaches a threshold, regardless of the amount of evidence favoring the alternative. In contrast, a relative rule requires that evidence favoring one of the choices has to exceed the other by some amount. Finally, models vary in whether starting points, rate of information uptake or thresholds vary across trials. Combinations of modeling assumptions generate the range of possible sequential sampling models.

Examples of sequential sampling models include a random walk model that stops accumulation of discrete packets of information over discrete time steps according to a relative rule (Laming, 1968), a Poisson counter model that describes accumulation of discrete, fixed information packets arriving at random time points until evidence for either of the choices reaches a threshold (Townsend & Ashby, 1983), and a Wiener diffusion model that describes accumulation of continuous packets in continuous time until a relative threshold is hit (Ratcliff, 1978).

In addition to assumptions about the accumulation process, models also add assumptions about variability in their parameters across trials. For example, an early random walk model assumed variability in starting points (Laming, 1968), a version of a Poisson counter model reported in Ratcliff and Smith (2004) added variability

in the rate of information uptake and thresholds, and a popular version of a Wiener diffusion model incorporates variability in starting points, rate of information uptake and non-decision time.

The sequential sampling framework has been fruitful in providing models that could be used to investigate correlations between parameters. The assumptions of a particular model lead to a precise prediction of how the decision evolves over time, and how response times and responses jointly vary (Ratcliff & Smith, 2004; Smith & Ratcliff, 2004). The equation for accumulation process and joint density can be used to simulate predictions to understand implications of theoretical assumptions, and to extract information from data using statistical methods. In the next section, I focus on the Bayesian statistical framework that allows a principled and intuitive way for extracting information about the supposed latent decision process from behavioral data. The simulation of the accumulation process and the joint density will be discussed in Section 4.2 and 4.2.

1.5 Bayesian Statistical Framework

Once we form a cognitive model with behavioral predictions, how do we connect it to data? In the following subsections, I present a Bayesian approach to inference and a fitting method based on Markov chains that can provide a way to estimate correlations from behavioral data (Berger, 1997; Casella & Berger, 2002; Gelman et al., 2014). In the Bayesian framework, a model of data is combined with a model of unknown parameters to provide a combined description of uncertainty. While theoretically everything tends to work out beautifully, in practice Bayesian models with

complex data distributions are usually analytically intractable and require an iterative algorithm, often called a sampler, to learn about parameters and the model(s) from data.

Recent developments in computing power and software made the Bayesian approach ripe for scientific use (Kruschke, 2010), and it is becoming popular in psychology, too (Craigmile et al., 2010; Edwards, Lindman, & Savage, 1963; Lee & Wagenmakers, 2013; I. J. Myung & Pitt, 1997; J. I. Myung, Karabatsos, & Iverson, 2008; Peruggia et al., 2002; Rouder & Lu, 2005; Rouder, Lu, Speckman, Sun, & Jiang, 2005; Vandekerckhove et al., 2011). The reasons for its appeal include quantifying of uncertainty with probability, making statistical inferences as probability statements and gaining the ability to fit realistic models. On these grounds, I adopted Bayesian approach to a data analysis study specified in Section 4.4, but before that I present the fundamental ideas of hierarchical modeling and Markov Chain Monte Carlo.

1.5.1 Bayesian Models

In the Bayesian framework model parameters are treated as random variables along with data. The full probability model, also called the Bayesian model, is at the center of analysis because all inference problems can be defined in terms of it (Gelman et al., 2014). A Bayesian model requires specifying the probability densities of both data and parameters. The probability densities are assumed to be conditionally dependent in a way that the full joint density can be factored into a probability density of data given parameters, called a sampling density, and the probability density of parameters, called a prior density. Given a data vector $\mathbf{x} \in \mathbb{R}^p$ and its parameter

vector $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^m$, a Bayesian model is written as

$$f(\mathbf{x}, \boldsymbol{\theta}) = f(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \quad (1.6)$$

where $f(\mathbf{x} \mid \boldsymbol{\theta})$ is a sampling density and $\pi(\boldsymbol{\theta})$ is a prior density for $\boldsymbol{\theta}$. The ability to factor a joint density of \mathbf{x} and $\boldsymbol{\theta}$ into a sampling and a prior density is useful for constructing new Bayesian models because it allows breaking the problem into manageable pieces, often consisting of common univariate distributions.

Statistical inference proceeds by conditioning unknown quantities of interest on the observed data \mathbf{x} . For example, estimation of parameters $\boldsymbol{\theta}$ amounts to calculating a posterior density $\pi(\boldsymbol{\theta} \mid \mathbf{x})$, which contains all the information about the parameters. Using Bayes' theorem,

$$\pi(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{f(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_{\Theta} \cdots \int f(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}} \quad (1.7)$$

(Berger, 1997; Casella & Berger, 2002; Gelman et al., 2014). We can then extract more specific probability statements about $\boldsymbol{\theta}$ from the posterior density such as $P(\boldsymbol{\theta} \in A \mid \mathbf{x})$, where $A \subseteq \Theta \subset \mathbb{R}^m$.

An important variant of Bayesian model is a hierarchical Bayesian model (Berger, 1997; Gelman et al., 2014). It fits well with data relevant for cognition: where measurements vary among participants, but due to similarity in cognitive abilities and experimental treatment, participants share regularities. Hierarchical models have gained attention psychology for their capacity to pool information together so that estimates are more precise, even if sample size is small (Craigmile et al., 2010; Rouder & Lu, 2005; Rouder et al., 2005; Rouder, Morey, & Pratte, 2014). In this thesis, however, the hierarchical modeling was not crucial because of large sample sizes and computationally inefficient, so it was applied.

To conclude, the joint density over observations and all parameters is the starting point of Bayesian analysis. The models I used in analyses of the benchmark dataset were too complex for analytical treatment, which brings us to the topic of how complicating the situation with Markov chains actually enables fitting the intractable models to data.

1.5.2 MCMC Background

After a Bayesian model has been specified, inference about parameters, out-of-sample observations or the model can proceed by applying the calculus of probabilities to obtain corresponding conditional densities. With continuous parameters, which is the typical case, calculating conditional densities requires integration, which for complex models is intractable. Application of Bayes' theorem to obtain a posterior density over unknown quantities is complicated by the integral in the denominator of Equation 1.7. In particular, the inability to integrate over parameters relative to their density plagues the best sequential sampling models, so the posterior density cannot be calculated exactly (Rouder et al., 2005; Vandekerckhove et al., 2011).

Due to the high-dimensionality of Bayesian models, numerical integration techniques can fall apart, leaving Monte Carlo methods as the only viable option (Gamerman & Lopes, 2006; Gelman et al., 2014; Givens & Hoeting, 2013; Robert & Casella, 2005). Monte Carlo methods rely on simulating pseudo-random draws from a distribution of interest to solve inference problems. For data models with known mathematical form, approximate inferences can be obtained through a special class of Monte Carlo algorithms, sometimes called samplers, that use Markov chains (MCMC) to sample from posterior densities. Given a posterior sample, accurate conclusions can

be drawn, even for complicated models (Craigmile et al., 2010; Peruggia et al., 2002; Vandekerckhove et al., 2011). In the rest of the section, I present a few definitions from Markov chain theory as background for a general class of MCMC algorithms built on that theory.

MCMC employs discrete-time, continuous state space Markov chains to sample from arbitrary distributions for which a probability density can be specified (Karlin & Taylor, 1975, 1981; Ross, 2014). Let $\mathbf{X} = \{\mathbf{x}_n : n \geq 0\}$ be a p-dimensional, discrete-time stochastic process with state space $\mathcal{S} \subseteq \mathbb{R}^p$. A first-order Markov chain is a stochastic process with a conditional distribution satisfying

$$f(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \dots, \mathbf{x}_0) = f(\mathbf{x}_n | \mathbf{x}_{n-1}), \quad (1.8)$$

for any $n \geq 1$. In words, the distribution of a current state depends only on the previous state, giving Markov chains a memoryless property.

A Markov chain \mathbf{X} can be thought of as a model of stochastic dynamics of a multi-dimensional system over discrete time. Description of dynamics requires an initial position in the state space and a transition rule that guides the system from state to state. Given that \mathcal{S} is continuous and \mathbf{X} is stochastic, a description of the process requires a marginal density of \mathbf{x}_0 , called an initial density, and a conditional density of \mathbf{x}_n given \mathbf{x}_{n-1} , called a transition density. Once densities describing dynamics are defined, evolution of the process can be analyzed analytically or through simulation.

It can be shown that some Markov chains, if given enough time, will go from an arbitrary initial density to an equilibrium density. Equilibrium behavior is a property of a transition density. For a transition density $f(\mathbf{x}_n | \mathbf{x}_{n-1})$ and $A \subseteq S$, a transition

kernel is defined as

$$P\{\boldsymbol{x}_n \in A \mid \boldsymbol{x}_{n-1}\} = \int_A f(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1}) d\boldsymbol{x}_n. \quad (1.9)$$

Starting from any point $\boldsymbol{x}_0 \in S$, a Markov chain has an equilibrium density if an nth-step transition kernel satisfies

$$\lim_{n \rightarrow \infty} P(\boldsymbol{x}_n \in A \mid \boldsymbol{x}_0) = \int_A \pi(x) d\boldsymbol{x}, \quad (1.10)$$

where $\pi(\cdot)$ is called an equilibrium density function.

Once a chain attains equilibrium behavior, whatever the initial position was and no matter how many transitions occur after, the chain remains in equilibrium. The sequence of states $\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \dots$ will be identically distributed according to the equilibrium density function. The equilibrium behavior provides a natural point of contact with the problem of Bayesian inference for complex models. If the equilibrium density is equivalent to the posterior density, then a sample of states of a Markov chain in equilibrium is a sample from the posterior.

It follows then that Bayesian analysis can proceed if a Markov chain can be simulated for which there exists a unique equilibrium density equivalent to the posterior density that can be reached from an arbitrary starting point. The limit-based definition of the equilibrium density implies that it is not guaranteed to exist nor to be unique. Four conditions need to be satisfied to obtain the desired subclass of Markov chains.

First, a Markov chain has to be homogeneous. Homogeneity is defined by a time-invariant transition density, such that

$$f(\boldsymbol{x}_n \mid \boldsymbol{x}_{n-1}) = f(\boldsymbol{x}_1 \mid \boldsymbol{x}_0), \quad (1.11)$$

for any $n \geq 1$. A homogeneous chain has fixed dynamics across time, but marginal densities $f(\mathbf{x}_n)$ are not guaranteed to be equal across time nor is there a guarantee that marginal densities converge to an equilibrium density. Homogeneity is a flexibility condition allowing for different initial densities and the possibility of convergence to an equilibrium density.

Homogeneous Markov chains have to be constrained further to ensure convergence to a unique equilibrium density. The second requirement is that a Markov chain is ϕ -irreducible. Let the time of first re-visit to A from \mathbf{x}_0 be a random variable $t_A = \inf_{n \geq 1} \{\mathbf{x}_n \in A\}$, where set $A \subseteq S$ and starting point $\mathbf{x}_0 \in A$. A Markov chain is ϕ -irreducible if there exists a marginal density $f(t_A)$, such that for all $\mathbf{x}_0 \in A$ and all $A \subseteq S$,

$$P\{t_A < \infty \mid \mathbf{x}_0\} > 0. \quad (1.12)$$

ϕ -irreducibility ensures that, no matter where the chain starts, it is possible to return there in a finite amount of time. In other words, there are no regions where a chain gets stuck, which would create a possibility for multiple equilibrium densities.

In addition to moving around the state space without getting stuck in some region, a chain should also have no cyclical behavior. The third required property of chains with unique equilibrium density is aperiodicity. An aperiodic Markov chain can move from any $A \subseteq S$ back to A in one transition, that is for $n \geq 1$ and $\mathbf{x}_{n-1} \in A$

$$P\{\mathbf{x}_n \in A \mid \mathbf{x}_{n-1}\} > 0. \quad (1.13)$$

As a result, an aperiodic chain can remain in region A for multiple steps.

The first three properties ensure that a Markov chain converges to a unique equilibrium density from most initial positions. To ensure that a process converges from

all initial positions, it must be Harris recurrent. For $A \subseteq S$ with $\pi(A) > 0$ and all $\mathbf{x} \in A$, a ϕ -irreducible Markov process with an equilibrium density $\pi(\cdot)$ is Harris recurrent if

$$P\{t_A < \infty \mid \mathbf{x}_0\} = 1. \quad (1.14)$$

Harris recurrence guarantees that all realized chains come back to the region where they start in finite time.

Recall that we started with a problem of statistical inference with complex Bayesian models that can be solved with Monte Carlo methods. Conditions of homogeneity, ϕ -irreducibility, aperiodicity and Harris-recurrence are sufficient to guarantee that a proposed MCMC algorithm can solve intractable integrals in Bayesian inference. The next issue we consider is how to find such algorithms.

1.5.3 Metropolis-Hastings Algorithm

The Markov chain theory on its own provides only conditions that a valid algorithm should satisfy, but gives no clues toward a working algorithm. As opposed to analysis in probability theory, where having a transition density and figuring out its equilibrium density is the problem, in Bayesian analysis the equilibrium density is known and a transition density needs to be figured out. A very general way to construct transition densities is a class of algorithms called Metropolis-Hastings (Gelman et al., 2014; Givens & Hoeting, 2013; Robert & Casella, 2005). The algorithm provides a way to define a transition density satisfying the four properties required for a Markov chain to converge to a known equilibrium density from an appropriate initial density.

In the situation where MCMC methods apply, we are in possession of a target density, $\pi(\cdot)$. We need a transition density, $p(\mathbf{y} \mid \mathbf{x})$ (or the associated transition kernel), whose equilibrium density is $\pi(\cdot)$. A necessary condition for $p(\mathbf{y} \mid \mathbf{x})$ to be a proper transition density is called detailed balance (also called time reversibility),

$$\pi(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) = \pi(\mathbf{y})p(\mathbf{x} \mid \mathbf{y}). \quad (1.15)$$

Detailed balance says that starting at \mathbf{x} and moving to \mathbf{y} happens as often as starting at \mathbf{y} and moving to \mathbf{x} . In addition, if the transition density $p(\mathbf{y} \mid \mathbf{x})$ has support equal to that of $\pi(\cdot)$, or is contained in it, then it will satisfy properties of homogeneity, ϕ -irreversibility, aperiodicity and Harris recurrence, ensuring convergence from any starting point in the state space to a unique equilibrium density.

Naive or ad-hoc methods for constructing a transition density are likely not to satisfy detailed balance, except for maybe special cases, but their failure is instructive. When a method fails to satisfy detailed balance it implies that, for some combinations of \mathbf{x} and \mathbf{y} , $\pi(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) > \pi(\mathbf{y})p(\mathbf{x} \mid \mathbf{y})$ or the reverse. Loosely, the chain moves too frequently from \mathbf{x} to \mathbf{y} and rarely from \mathbf{y} to \mathbf{x} . The Metropolis-Hastings algorithm is a general approach to MCMC algorithms that provides a way to correct the imbalance by introducing a possibility that a transition may be rejected.

Suppose the chain at time $n - 1$ is in state $\mathbf{x}_{n-1} = \mathbf{x}$. For a move to take place, a Metropolis-Hastings algorithm requires a proposal density $q(\mathbf{y} \mid \mathbf{x})$ from which a candidate value can be drawn. A candidate value is then accepted with probability $\alpha(\mathbf{y}, \mathbf{x})$, called the acceptance probability. The transition density in a situation when the chain moves ($\mathbf{x}_n = \mathbf{y}$) is

$$q(\mathbf{y} \mid \mathbf{x})\alpha(\mathbf{y}, \mathbf{x}). \quad (1.16)$$

In a situation where the chain does not move ($\mathbf{x}_n = \mathbf{x}_{n-1}$), there is a probability mass at \mathbf{x} ,

$$r(\mathbf{x}) = 1 - \int_S q(\mathbf{y} \mid \mathbf{x}) \alpha(\mathbf{y}, \mathbf{x}) d\mathbf{y}, \quad (1.17)$$

with the integral accounting for all possible moves. Combining the two pieces, for $d\mathbf{y} \subseteq S$ and $\mathbf{x} \in S$, the transition kernel of a Metropolis-Hastings chain is

$$P\{\mathbf{x}_n \in d\mathbf{y} \mid \mathbf{x}_{n-1}\} = q(\mathbf{y} \mid \mathbf{x}) \alpha(\mathbf{y}, \mathbf{x}) d\mathbf{y} + r(\mathbf{x}) \delta_{\mathbf{x}}(d\mathbf{y}), \quad (1.18)$$

where $q(\mathbf{y} \mid \mathbf{x}) = 0$ when $\mathbf{y} = \mathbf{x}$, and Dirac's $\delta_{\mathbf{x}}(d\mathbf{y}) = 1$ when $\mathbf{x} = \mathbf{y}$ and 0 otherwise.

The final part of the algorithm is the definition of the acceptance probability. From the consideration of the two kinds of inequalities that violate the detailed balance condition, the acceptance probability for moving from \mathbf{x} to proposed \mathbf{y} is

$$\alpha(\mathbf{y}, \mathbf{x}) = \begin{cases} \min\left(\frac{\pi(\mathbf{y})q(\mathbf{x} \mid \mathbf{y})}{\pi(\mathbf{x})q(\mathbf{y} \mid \mathbf{x})}, 1\right), & \pi(\mathbf{x})q(\mathbf{y} \mid \mathbf{x}) > 0 \\ 1, & \pi(\mathbf{x})q(\mathbf{y} \mid \mathbf{x}) = 0 \end{cases}$$

The form of $\alpha(\mathbf{y}, \mathbf{x})$ implies that the target density needs to be known only up to a normalizing constant, which cancels out in the fraction. The cancellation of normalizing constants is directly relevant for Bayesian inference because these constants are the intractable integrals. The form also implies that the algorithm always moves to places of higher probability density, but moves to places of lower probability with some probability. Algorithm 1 shows Metropolis-Hastings pseudo-code for sampling N draws from a target density $\pi(\cdot)$.

Algorithm 1 Metropolis-Hastings Pseudo-code

Initialize at \mathbf{x}_0 such that $\pi(\mathbf{x}_0) > 0$

for $j = 1, 2, \dots, N$ **do**

 Sample $\mathbf{y} \sim q(\mathbf{y} \mid \mathbf{x}_{j-1})$

 Sample $u \sim \mathcal{U}(0, 1)$

if $\alpha(\mathbf{y}, \mathbf{x}_{j-1}) \geq u$ **then**

 Set $\mathbf{x}_j = \mathbf{y}$

else

 Set $\mathbf{x}_j = \mathbf{x}_{j-1}$

end if

end for

Chapter 2: Motivating Questions

In the first chapter I raised some questions about dependent variation in parameters of characterizing cognitive processes organizing simple choice behavior. I presented some behavioral and modeling evidence for it, and reviewed a theoretical and statistical framework within which answers to these questions may be sought. In this chapter, I elaborate on questions addressed by this thesis. There are three overarching questions that I pursued: How to incorporate trial-to-trial dependencies between parameters into models of cognitive processes underlying behavior on two-choice tasks? How do predictions of cognitive models change under assumption of dependence relative to independence of parameters? What information about dependencies between parameters, given some sequential sampling model of the decision process, does the benchmark dataset have?

2.1 Modeling Parameter Dependencies

One of the main questions motivating this thesis is how to formally describe dependencies of cognitive processes across trials. However, before stating the question in an answerable form, recall that behavioral and modeling data suggest that across trials parameters that capture underlying processes show both auto-correlation and

cross-correlation. As a final target of research we may consider finding a system of equations that can describe such dynamics in a way that reflects a mechanistic theory.

Suppose we are dealing with a sequential sampling model of binary choice that has some initial evidence ξ , rate of evidence accumulation δ , threshold separation α and non-decision time τ^{nd} . On each trial a participant generates a response time t^{rt} and a response r . Then, change in the task-specific processing from trial n to $n + 1$ may be captured by the following dynamic system:

$$\begin{aligned}\alpha(n+1) &= f_\alpha(\alpha(n), t^{rt}(n), r(n), \theta_\alpha), \\ \xi(n+1) &= f_\xi(\xi(n), t^{rt}(n), r(n), \theta_\xi), \\ \delta(n+1) &= f_\delta(\delta(n), t^{rt}(n), r(n), \theta_\delta), \text{ and} \\ \tau^{nd}(n+1) &= f_{\tau^{nd}}(\tau^{nd}(n), t^{rt}(n), r(n), \theta_{\tau^{nd}}).\end{aligned}\tag{2.1}$$

The dependence of parameters at trial $n + 1$ on themselves at trial n and other stable processes captured by θ_i can explain first-order auto-correlation, and dependence on previous choice and its speed can explain cross-correlations.

There have been some quantitative proposals that capture auto-correlation of some parameters across trials, which is a special case of the above system (S. D. Brown et al., 2008; Cho et al., 2002; Gao et al., 2009; Goldfarb et al., 2012; Laming, 1968; Luce, 1986; Vickers, 1979). The thrust of this work relies on the rich pattern of sequential effects that provide suggestive constraints on dynamic rules for various parameters. For example, Goldfarb et al. (2012) examined post-error slowing and suggested a dynamic rule for threshold separation that leads to increases in response time after error and decreases after correct responses. Another suggestive sequential effect is the

speed up of correct responses on the second repeating stimulus with short response-to-stimulus intervals, which Gao et al. (2009) described with a bias in the starting evidence towards the correct boundary.

However, when it comes to cross-correlations of parameters, the behavioral data is less clear and I found no explicit modeling studies of cross-correlations. With no empirical base to guide a mechanistic account we are left with a descriptive, statistical approach. Instead of specifying dynamic rules for all or some of the parameters to impose dependencies, we can replace assumptions of independent variation of the standard models with dependent variation. Dependent variation can be specified as a multivariate distribution that provides means and scales as the standard models, but also adds correlations between parameters. The magnitude and direction of correlations between different pairs of parameters could then suggest a target for a comprehensive mechanistic theory expressed in a system of dynamic equations.

By formalizing parameter relations in a statistical manner, the first question can now be formulated more concretely: what multivariate density should model joint variation among parameters across trials? A couple of key properties of standard sequential sampling models provide important constraints on the search space. One feature of all standard models is that parameters fall on different scales. Rates of evidence uptake take values from \mathbb{R} , evidence thresholds are positive, and initial evidence may fall between 0 and the evidence threshold (Ratcliff & Smith, 2004). These parameters are psychologically meaningful and provide theoretical interpretation of behavior or neural activity. Because sequential sampling models are used to represent psychological theory rather than simply fit or predict data, it is important to preserve the scales. In addition to different scales, shapes of the probability density functions

that we assign to each parameter may all be different to reflect different kinds of marginal variation (Jones & Dzhafarov, 2014b; Ratcliff, 2013).

The combination of scales and marginal densities restrictions is hard to satisfy with common multivariate densities. There is a stock of multivariate densities that could be used to model dependence in certain situations, including the normal, t, gamma, and F (Kotz, Balakrishnan, & Johnson, 2000). However, they are too restricted in their properties to be used for generalizing sequential sampling models. For example, in each of the commonly used densities the support of each dimension is the same, and so are the marginal densities. An n-dimensional multivariate normal has support over \mathbb{R}^n and each marginal is a normal density with some mean and variance. These considerations suggest that we need a method for defining new multivariate densities that fit the above restrictions and express alternative correlation structure.

2.2 Predictions of Dependent Parameters

Assuming that we have specified a joint distribution of parameters, we can now examine the consequences of dependent parameters. Assessing predictions made by models is a natural way of understanding what model assumptions imply. Comparison of a model with dependent parameters to a model with independent parameters can reveal how trade-offs between parameters shape processing and provide insight into which data patterns indicate dependent parameters. The result of this should be a better understanding of whether correlation parameters are important for modeling because they can fit some feature of data that standard models cannot and whether experiments manipulating correlations may lead to new effects that standard models do not predict.

Sequential sampling models of a simple decision process provide two avenues for understanding implications of correlated parameters. On the one hand, because sequential sampling models are dynamic, we can consider how dependencies alter the accumulation process on average. The average accumulation process is mathematically described by the mean sample path. The mean sample path $E[x(t) | \zeta, \alpha, \delta, \sigma]$ is the expected sequence of states through which a stochastic process evolves over time t given the initial state ζ and other parameters. The importance of examining dynamics of accumulation is because it can be related to single-cell recording (Ratcliff et al., 2007), functional magnetic resonance imaging (Hare et al., 2011) or electroencephalography (Polanía, Krajbich, Grueschow, & Ruff, 2014), and hence expose the models to other kinds of data.

On the other hand, instead of looking at the evolution of the process, we can examine its final outcome. The boundary hit by the accumulation process and the hitting time are important outcomes of bounded accumulation process that relate to speed and accuracy data. The joint probability density of response time and response formalizes predictions for behavior based on the assumptions about the decision process and non-decision processes. To obtain the joint density we need to combine a model of parameter variation with a model of behavior dependent on the parameters. Let $f(t^{rt}, r | \boldsymbol{\theta})$ be a model of response time t^{rt} and response r , and $g(\boldsymbol{\theta} | \boldsymbol{\phi}, \boldsymbol{\Sigma})$ be a model of across-trial variation in parameters. Then, a natural way to combine the two densities in a way that leads to predictions, and can be fit to behavioral data, is to form a mixture model (Casella & Berger, 2002).

A mixture model represents a situation when observables belong to latent sub-populations and we want to describe variation of observables aggregated into a single

population. In our case, $\boldsymbol{\theta}$ is unknown and is not of interest, so to describe variation in (t^{rt}, r) we form a mixture model

$$f(t^{rt}, r | \boldsymbol{\phi}, \boldsymbol{\Sigma}) = \int_{\Theta} f(t^{rt}, r | \boldsymbol{\theta}) g(\boldsymbol{\theta} | \boldsymbol{\phi}, \boldsymbol{\Sigma}) d\boldsymbol{\theta}. \quad (2.2)$$

With a mixture model we can obtain predictions for some of the features of simple choice behavior demonstrated in the Ratcliff and Rouder (1998) dataset and see how it is distinguished from the standard models with independent parameters. It would be of most interest to understand what the mixture model predicts for response time probability densities of different responses, response probabilities and relations between response times and response probabilities.

2.3 Correlation Structure in a Benchmark Dataset

The mixture model defined in the previous section is a description of how response times and responses vary. With appropriate statistical methods, we can use the mixture model to analyze the Ratcliff and Rouder (1998) benchmark dataset. The main question is what is the correlation structure of processing parameters underlying simple choice behavior in a brightness discrimination task. Given the paucity of data bearing on this question, the simplest assumption that a statistical model would have to satisfy is that the correlation structure is stable across the whole experiment. The stability implies that experimental manipulations used during collection of the benchmark dataset, that is speed-accuracy instructions and brightness level, only affect means of processing parameters, but not their correlations.

Given that the correlation structure is stable, what is the pattern of dependencies among parameters? This question can be formalized statistically as estimation of the correlation matrix. Or, in a Bayesian framework, the question can be formulated

as a probability of, say, a positive correlation between non-decision time and rate of evidence accumulation. Either way, what's important is inferring the pattern of correlation when there is no experimental attempt to induce them.

Chapter 3: Models

With questions motivating this thesis established, next I describe one of the standard sequential sampling models and how it might be generalized. As I discussed in Section 1.4, there are many sequential sampling models that account for performance of two-choice tasks in a similar manner. To complete this thesis, I used a model developed by Ratcliff and colleagues that belongs to a class of continuous time, continuous evidence accumulators with a relative stopping rule.

The Ratcliff diffusion model is a variant of the Wiener diffusion model (Laming, 1968; Link & Heath, 1975; Ratcliff, 1978; Ratcliff & Rouder, 1998; Ratcliff & Tuerlinckx, 2002). I concentrate on this model because of its long development (Ratcliff, 1978, 2002, 2013; Ratcliff & Rouder, 1998; Ratcliff & Tuerlinckx, 2002), and successful testing with a wide range of experimental data (Ratcliff & McKoon, 2008; Wagenmakers, 2009). In the following section I present a mathematical description of the Ratcliff model, its psychological interpretation, and show some of the features of performance data the model can and cannot predict. The model presentation follows with an introduction of the notion of copulas and how they may be used to construct flexible multivariate distributions (Braeken, Kuppens, De Boeck, & Tuerlinckx, 2013; Braeken, Tuerlinckx, & De Boeck, 2007; Genest & MacKay, 1986; Joe, 1997; Nelsen,

2006; Sklar, 1959). Lastly, I propose two multivariate distributions of parameter variation that effectively generalize the Ratcliff model.

3.1 Ratcliff Diffusion Model

The Ratcliff diffusion model is arguably the most successful sequential sampling model. Underlying the Ratcliff model is a drift-diffusion Wiener process, a special case of a continuous time, continuous state space diffusion process that begins at some starting point and evolves until it reaches one of two absorbing boundaries (Karlin & Taylor, 1975, 1981; Ross, 2014). Formally, a Wiener process is an uncountable, ordered collection of univariate random variables

$$\{x(t); t \in T\} \quad (3.1)$$

with the index set $T \in [0, \infty)$ (Karlin & Taylor, 1981; Smith, 2000). Given two absorbing boundaries at 0 and $\alpha > 0$, and a starting point $\zeta \in (0, \alpha)$, the state space \mathcal{S} of $x(t)$ is the interval $[0, \alpha]$. A realization of the Wiener process, called a sample path, is a real-valued univariate function that maps the index set into the state space. In other words, a sample path is a particular sequence of states through which the Wiener process evolves until it hits a boundary and terminates.

Evolution of the drift-diffusion Wiener process is characterized by drift and diffusion coefficients. A drift coefficient,

$$\delta = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \text{E}[x(t + \Delta t) - x(t)] \in \mathbb{R}, \quad (3.2)$$

quantifies mean instantaneous change in the state of the process and a diffusion coefficient,

$$\sigma^2 = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \text{Var}[x(t + \Delta t) - x(t)] > 0, \quad (3.3)$$

quantifies mean instantaneous squared change in the state of the process.

The drift coefficient determines the direction and magnitude with which an instance of the Wiener process tends to evolve towards a boundary. Positive values bias evolution towards α and negative values towards 0, while the absolute value of δ controls how rapidly the process approaches an absorbing boundary. In contrast, the diffusion coefficient determines variability in evolution. For small values, the state of the process may fluctuate a bit, but for large values, the process may show large fluctuations in direction and moment-to-moment transitions.

There are several equivalent ways to complete a definition of the Wiener process assuming the above facts. One definition arises from considering the evolution of the Wiener process, formalized by the linear, first-order stochastic differential equation

$$dx(t) = \delta dt + \sigma dw(t), \quad (3.4)$$

where δ and σ are defined as above, and $x(t)$ is a standard Wiener process with points distributed as normal variates with mean 0 and variance dt . Change in the process is independent of time and current state of the process, and is a sum of an average increment and scaled Gaussian noise. Applying a stochastic integral leads to the Wiener process $x(t)$ as a unique solution to Equation 3.4.

Another approach is to describe the initial and boundary conditions, and the properties of increments, $x(t_2) - x(t_1)$, of the process. The drift-diffusion Wiener process $x(t)$ with absorbing boundaries satisfies the following:

1. $x(t)$ is an almost surely continuous function,
2. $x(0) = \zeta$, where $0 < \zeta < \alpha$,

3. the value of the probability density at absorbing boundaries is $f(x(t) = 0 | x(0) = \zeta) = f(x(t) = \alpha | x(0) = \zeta) = 0$,
4. $x(t) \sim \mathcal{N}(\delta t + \zeta, \sigma^2 t)$,
5. the increments are stationary; that is for $t_1, t_2 \in T$, where $0 < t_1 < t_2$, $X(t_2) - X(t_1)$ is equal in distribution to $X(t_2 - t_1)$, and
6. for $t_1, t_2, \dots, t_n \in T$, with $t_1 < t_2 < \dots < t_n$, the increments $X(t_1), X(t_2) - X(t_1), \dots, X(t_n) - X(t_{n-1})$ are independent.

From this definition, we see that the Wiener process starts at ζ and continuously evolves until it is absorbed in one of two boundaries. If we took a sample of states over a discrete time grid, then all the increments are independent normal variates that depend only on the differences between time points, but not absolute time.

In addition to its mathematical assumptions, the Ratcliff diffusion model provides linking assumptions that map the Wiener process and its parameters into a psychological process and variables relevant to understanding two-choice behavior. In line with sequential sampling approach to decision making, the drift-diffusion Wiener process models a perfect but noisy integrator of relative information. This integrator starts at some initial amount of decision-relevant information and continuously accumulates continuous information samples from a sensory representation without loss of information (Bogacz et al., 2006; Smith, 2000).

The index set $T = [0, \infty)$ describes the time passed from the beginning of the accumulation, and the state space $S = [0, \alpha]$ describes the possible amounts of information that an integrator can hold under certain conditions. The drift coefficient

δ measures the average rate of information uptake by a perfect integrator, and the diffusion coefficient σ^2 describes the level of noise in the information uptake.

The boundaries of the state space, 0 and α , represent decision thresholds that quantify the amount of relative information needed to form a decision. The lower and upper thresholds could represent each of two choices, say “No” and “Yes” in the auditory signal detection experiment. The separation between boundaries, α , quantifies decision caution, with larger separation implying that more information is required before making a decision. The two thresholds force a relative stopping rule where information is relative because movement towards one threshold is movement away from the other.

The starting point ζ of the process represents the initial amount of information from which the accumulation process begins. The level of initial information represents the bias an observer may have towards one of two choices. The farther away from $\alpha/2$ the starting point ζ is, the more initial information there is for a particular choice. The initial information asymmetry suggests an alternative parameterization of the Wiener process corresponding to a biased integrator that starts accumulation with bias $\beta = \zeta/\alpha \in [0, 1]$, a proportion of the distance from 0 to α .

The drift-diffusion Wiener process model of a perfect integrator provides a natural way to predict the joint probability density of decisions and decision times in a two-choice task. A sample path of the drift-diffusion Wiener process will start at ζ (or β) and take some random time to absorb at one of the boundaries. Let $t^d > 0$ be the time to absorption and $r \in \{0, 1\}$ be the boundary passed at t^d , with 1 corresponding to boundary at α . Translating into psychological terms, the sample path is the evolution of the information integrator to a decision r during the duration of decision time t^d .

The most studied prediction of the Ratcliff diffusion model is the joint variation of decisions and decision times (Ratcliff & McKoon, 2008). The component of the joint probability density function when $r = 1$

$$f(t^d, 1 \mid \alpha, \delta, \zeta, \sigma^2) = \frac{\pi\sigma^2}{\alpha^2} \exp\left(\frac{\delta\zeta}{\sigma^2} - \frac{\delta^2 t^d}{2\sigma^2}\right) \sum_{k=1}^{\infty} k \exp\left(\frac{-k^2\pi^2\sigma^2 t^d}{2\alpha^2}\right) \sin\left(\frac{k\pi(\alpha - \zeta)}{\alpha}\right), \quad (3.5)$$

and the component for the opposite response is obtained by replacing δ with $-\delta$ and $\alpha - \zeta$ with ζ . The form of Equation 3.5 suggests that distributional prediction of the Wiener process model suffers from lack of identifiability (Donkin, Brown, & Heathcote, 2009). A model is identifiable when for any nonequivalent parameters $\theta_1, \theta_2 \in \Theta$, the resulting probability density functions are unique, that is $f(x \mid \theta_1) \neq f(x \mid \theta_2)$ (Casella & Berger, 2002). As a result, identifiability of a model creates the possibility of precise estimation of true parameters with an infinite sample size. For Equation 3.5, assuming that $\{\alpha, \delta, \zeta, \sigma^2\}$ are free to vary, if one multiplies all parameters of the model by a factor c , then the predictions do not change. Thus, parameter recovery is unique up to a scaling operation.

For the Wiener model, resolving this lack of identifiability requires fixing one of the parameters to an arbitrary value. Typically, application of the Wiener model relied on fixing the diffusion coefficient σ^2 to 0.1 or 1 across experimental conditions (Ratcliff, 1978; Ratcliff & McKoon, 2008; Ratcliff & Tuerlinckx, 2002). The stability of noise is a theoretical assumption, and ultimately should be tested against a particular experimental situation. Given that this thesis concentrates on the Ratcliff and Rouder (1998) data, which was generated without manipulations aimed at changing noisiness of the integrator, I will assume that $\sigma^2 = 0.1$, across all experimental conditions, to make predicted distributions identifiable.

The model specified so far allows predictions only about t^d , which represents the unobservable decision time. To fully connect the Wiener process to behavioral data, the model requires assumptions about the perceptual and motor processes surrounding the decision stage. A sufficient adjustment is to relate the decision time to the response time because the response is assumed to reflect the decision without error, or that motor processing exactly reflects which evidence bound was reached.

An additive decomposition of two-choice tasks implies that before information sampling can begin, the sensory stimulus has to be transduced and its features encoded. When the accumulation reaches one of the thresholds, the observer has to organize and execute a motor response for his or her decision to become observable. The mechanistic details of such perceptual and motor processing are typically not working with a sequential sampling model (but see S. D. Brown et al. (2008); Smith and Ratcliff (2009)), but their time can be captured by a single parameter τ^{nd} . Then, response time

$$t^{rt} = t^d + \tau^{nd} \quad (3.6)$$

where t^d is the decision time and τ^{nd} is the combined perceptual and motor processing (non-decision) time. If τ^{nd} is a constant, the joint probability density of response times and responses is a shifted density of decision times and decisions

$$f(t^{rt} - \tau^{nd}, r | \alpha, \delta, \zeta, \sigma^2) = f(t^{rt}, r | \alpha, \delta, \zeta, \sigma^2, \tau^{nd}), \quad (3.7)$$

with support over $t^{rt} \in [\tau^{nd}, \infty)$ and $r \in \{0, 1\}$.

Comparison of the Wiener model's predictions with experimental data shows that, for some experimental paradigms, the model can accurately predict several qualitative and quantitative features of behavioral data. First, large variation in response times,

as shown in Figure 1.1, arises from within-trial variability in accumulation. Variability follows because sometimes the integrator uptakes information rapidly resulting in a fast response time. On other trials, the integrator may oscillate for a while around the initial state before terminating, leading to a slow response time.

Second, noisy within-trial accumulation accounts for errors. To model accuracy one has to assume that the starting point $\zeta = \alpha/2$, the drift rate $\delta \geq 0$ and boundaries represent correct (upper) and error (lower) responses. The starting point restriction reflects an assumption that there is no bias towards correct or error responses, and the non-negative drift rate represents the assumption that stimulus information restricts performance to be between chance and full accuracy. With these assumptions, noisy accumulation sometimes results in the integrator terminating at the lower bound, which causes an error.

Third, the decision model based on the Wiener process can predict the positive skew of response time densities observed in Figure 1.1. The Wiener process tends to move directly towards one of the bounds, and only sometimes oscillates between the bounds or reverses its course, which predicts that response time probability densities for reaching $r = 1$ and $r = 0$ will have a positively skewed shape.

Fourth, the model can explain the speed-accuracy trade-off. Experimentally, the speed-accuracy trade-off is obtained by manipulating performance instructions. The Wiener model can predict the trade off, like the one in Figure 1.1, by adjusting the threshold parameter, α , or, in psychological terms, using instructions to manipulate a participant's level of caution. Assuming an unbiased starting point, $\zeta = \alpha/2$, and $r = 1$ representing the correct response, then increasing α will result in the integrator taking a longer time to produce the correct response. The farther distance from ζ

to 0 will also increase accuracy by allowing some of the accumulation paths initially moving towards the error bound to reverse their trajectory towards the correct bound. However, recall that there are multiple studies either of physiological or cognitive modeling type that conclude that other parameters need to simultaneously adjust with α to full explain effects of instructions (Rae et al., 2014; Rinkenauer et al., 2004; Vandekerckhove et al., 2011).

Fifth, the Wiener model predicts that with higher accuracy participant's correct response times will tend to be faster. This pattern is observed for both speed and accuracy instructions, as shown in Figure When instructions are fixed, only the stimulus quality is manipulated. Intuitively, with better stimulus information the task becomes easier, so the accuracy goes up and processing time is faster. The observed relation is predicted by assuming that the quality of stimulus, such as the ratio of white to black pixels, modulates the drift rate. When $r = 1$ is the correct response, the Wiener model captures the joint effect because increasing the drift rate forces accumulation to move more rapidly and more frequently towards the upper bound.

The five features captured by the Wiener model are only a fraction of the full collection of regularities found in the benchmark and other datasets. The model is theoretically inadequate to account for asymmetric relations between correct and error response times, additional variation in the left tail of the response time density and auto-correlation of response times and responses.

Motivated by some failures of the Wiener model, the Ratcliff model adds additional structure to the Wiener process that helps to account for asymmetry between correct and error response times as well as additional left tail variation (Ratcliff, 1978; Ratcliff & Rouder, 1998). The solution to these empirical failures is to propose additional

variation in cognitive processing, which manifests itself in random fluctuations of parameters across trials. Specifically, the Ratcliff model assumes that there is trial-to-trial variation in stimulus quality, initial state of the integrator and non-decision time.

Mathematically, the idea of processing variation can be modeled with the drift-diffusion Wiener process with parameters varying across trials according to some multivariate distribution. Thus, a sample of response times arises from a mixture of Wiener processes evolving under different combinations of parameters. The full Ratcliff diffusion model assumes that the starting point ζ , non-decision time τ^{nd} and drift rate δ are random variables with probability densities

$$\begin{aligned}\zeta &\sim \mathcal{U}(\lambda - \frac{\gamma}{2}, \lambda + \frac{\gamma}{2}), \\ \tau^{nd} &\sim \mathcal{U}(\chi - \frac{\phi}{2}, \chi + \frac{\phi}{2}), \text{ and} \\ \delta &\sim \mathcal{N}(\nu, \eta^2),\end{aligned}\tag{3.8}$$

where $\lambda > 0$ and $\chi > 0$ are mean parameters, $\gamma > 0$ and $\phi > 0$ are range parameters, $\nu \in \mathbb{R}$ is a mean parameter and $\eta > 0$ is a variance parameter. To make sure that the starting point is within the state space and non-decision time is positive, the parameters satify $0 < \lambda - \frac{\gamma}{2} < \lambda + \frac{\gamma}{2} < \alpha$ and $0 < \chi - \frac{\phi}{2}$.

The distributional assumptions for ζ , τ^{nd} and δ were made for partially for theoretical reasons (Ratcliff, 1978; Ratcliff & Rouder, 1998; Ratcliff & Tuerlinckx, 2002). For example, normally distributed drift represents the assumption of noisy stimulus effects such that, for nominally identical stimuli, the internal representation varies due to a multitude of independent random influences, which gives rise to overall normal variation by the central limit theorem (Casella & Berger, 2002; Vickers, 1979).

The starting point variability may reflect momentary biases and residual evidence across trials, and variability in non-decision time reflects noisiness of the peripheral nervous system. In addition to these theoretical reasons, the distributional assumptions simplify parameter estimation, which traditionally is formulated as optimizing a measure of goodness-of-fit (Tuerlinckx, 2004; Vandekerckhove & Tuerlinckx, 2007; Voss & Voss, 2007).

Given that many other assumptions can be made, a natural concern is the sensitivity of model predictions and substantive conclusions to assumptions of across-trial variability. Ratcliff (2013) demonstrated that the model fit is robust with respect to a wider class of distributions. For a wide range of parameters, swapping in a beta for the normal distribution of drift, a beta for the uniform starting point, a normal or exponential for the uniform non-decision time, one at a time, resulted in accurate recovery of parameters and the same substantive conclusions. Except when parameter values were extreme, the assumption of exponentially-distributed non-decision time, or normally-distributed non-decision time with small boundary α , the study showed that empirical success of the model is moderately insensitive to across-trial variability assumptions.

The Ratcliff assumptions imply that the first passage density is a statistical mixture model with joint density function

$$f(t^{rt}, r \mid \alpha, \lambda, \gamma, \nu, \eta, \chi, \phi) = \int_V \int \int f(t^{rt}, r \mid \alpha, \zeta, \delta, \tau^{nd}) f(\zeta \mid \lambda, \gamma) f(\delta \mid \nu, \eta) f(\tau^{nd} \mid \chi, \phi) d\zeta d\delta d\tau^{nd}, \quad (3.9)$$

where $r = 0, 1$ and the individual densities have been specified above.

This probability density determines all the falsifiable predictions of the Ratcliff diffusion model for two-choice behavioral data and can accurately describe the five features of performance data mentioned above, as well as account for two additional features(Ratcliff, 2002; Ratcliff & Tuerlinckx, 2002).

The first additional data feature the Ratcliff model can handle is occasional asymmetry between correct and error mean response times (Ratcliff & McKoon, 2008; Wagenmakers, 2009). An asymmetry reflects correlation between response times and responses. There are three basic patterns that can arise: slower errors, faster errors, and a cross-over pattern where for easy stimuli errors are faster and difficult stimuli errors are slower. The benchmark data shows both slow and fast errors in Figure depending on the speed-accuracy instruction. These patterns are very diagnostic, and have long been a target for model development (Laming, 1968; Ratcliff, 1978; Ratcliff & Rouder, 1998; Vickers, 1979).

The Ratcliff model can handle all three asymmetry patterns with the across-trial variation assumptions. The slow errors arise only from assuming that the drift rate δ follows a normal distribution. How drift variation produces slow errors can be understood by considering that the mean correct and error response times for the mixture model are weighted averages of correct and error response times with respect to variation in drift rate values.

Assume that t_c^{rt} is the correct response time and t_e^{rt} is the error response time. Then, assuming only the drift rate varies, the correct mean of the mixture model

$$E[T_c^{rt}] = E_\delta [E[T_c^{rt} | \delta]] = \int_{\mathbb{R}} E[T_c^{rt} | \delta] f(\delta | \nu, \eta^2) d\delta, \quad (3.10)$$

and similarly the error mean of the mixture model

$$\mathbb{E} [T_e^{rt}] = \mathbb{E}_\delta [\mathbb{E} [T_e^{rt} | \delta]] = \int_{\mathbb{R}} \mathbb{E} [T_e^{rt} | \delta] f(\delta | \nu, \eta^2) d\delta, \quad (3.11)$$

where $f(\delta)$ is a normal density with mean ν and variance η^2 . Suppose stimulus favors $r = 1$ and ν is positive, so the mass of the $f(\delta)$ is heavily distributed over positive numbers. For a positive drift rate, the error mean is slower than the correct mean. Thus, averaging mean correct and error response times with respect to $f(\delta)$ will result in higher weight of fast correct means and slow error means. Hence, the overall mixture pattern would be slower errors.

Fast errors can be incorporated into a Ratcliff diffusion model by assuming variability in the starting point ζ . With drift rate fixed to a positive value, most Wiener processes will tend to terminate at the correct bound, so for errors to occur the process has to rapidly evolve towards the lower bound. Thus, mean error response times for starting values closer to the lower bound will tend to be faster than mean correct response times for symmetric starting values. When equally weighted by the uniform density, the mixed means will result in faster errors.

Combining variability in the drift rate and starting point in the Ratcliff model can produce all three patterns. The switch between different patterns occurs from relative differences in the standard deviations of the drift rate and the starting point, or from the relative means. When variability of the drift rate is sufficiently higher than variability of the starting point, the pattern of slow errors arises. Reversing the relation will result in fast errors. For relatively close levels of variability, by shifting the mean of the drift rate from high to low, performance moves from fast errors to slow errors.

The second additional pattern the Ratcliff model predicts is extra variability in the lower tail not predicted by the Wiener model. The pattern can be quantified by examining the 0.1 quantile. Assuming trial-to-trial variability in the non-decision time τ^{nd} allows the Ratcliff model to predict this empirical pattern. The reason is that the non-decision parameter sets the lower bound on the fastest response time, so the effect of across-trial variation is strongest in the left tail. Psychologically, fast response times correspond to the decision process being around its peak speed, so most variability arises from perceptual and motor processing.

Even with all the variability assumptions, the Ratcliff model still cannot capture two important features of behavioral data. First, it is typical to observe extremely fast or slow response times within a series of trials, and second, serial dependencies among responses and response times are ubiquitous. Several studies have proposed ways to model these features (Craigmile et al., 2010; Gao et al., 2009; Jones et al., 2013; Ratcliff & Tuerlinckx, 2002; Vandekerckhove & Tuerlinckx, 2007). The auto-correlation in the response times can be captured by assuming, that next trial's parameters somehow depend on the current or even past trial's parameters. Learning and information carry-over are two plausible psychological process that could underlie such auto-correlation. In contrast, the extreme response times can be accounted by fast guesses or delayed decisions, and modeled with a mixture of Wiener processes, delayed Wiener processes and a guessing process.

Overall, the Ratcliff diffusion model accounts accurately for many features of behavioral data coming from two-choice tasks. For this reason I used it as a basis for investigating parameter correlations. One of the questions motivating this thesis is how to model correlations between parameters. As discussed above, I took a

statistical approach that requires a multivariate density function to describe parameter variation. The next section introduces a method for defining new multivariate densities based on the concept of copulas.

3.2 Copula-based Multivariate Distributions

Generalizing existing simple decision models to have correlated parameters requires a multivariate distribution that captures variances and covariances of the parameters. The problem of finding a candidate distribution is both underconstrained, given the lack of information about the covariance structure, and restricted, because the parameters do not have the same scale. Hence, the method of distribution construction should allow combining an arbitrary covariance structure and arbitrary univariate marginal distributions.

For simplicity, consider the example of standard signal detection theory. Recall that the model of hits and false alarms is controlled by three free parameters: $\mu_s \in \mathbb{R}$, $\sigma_s > 0$, and $\tau \in \mathbb{R}$. The usual assumptions say that these parameters do not vary across trials and have no mutual restrictions. Suppose in a paradigm where stimulus level changes across trials, such as in the Ratcliff and Rouder (1998) brightness discrimination experiment, representational parameters μ_s and σ_s could vary. This variation could be such that, when μ_s increases in response to a stronger stimulus, σ_s tends to decrease. This negative correlation corresponds to a representation process with a noise suppression property. If we are interested in testing such a representational hypothesis, then we need to formalize parameter variation in a bivariate distribution for μ_s and σ_s . The obstacles to such a formalization, similar to simple

decision models, are that the exact nature of the relation between μ_s and σ_s is not known and they have different scales.

The framework of copulas, developed in probability theory over the last 60 years and recently imported for data analysis in a variety of fields, is one way to handle obstacles to modeling across-trial variability (Berkes, Wood, & Pillow, 2009; Braeken et al., 2013, 2007; Genest & MacKay, 1986; Joe, 1997; Nelsen, 2006; Sklar, 1959). A copula C is a multivariate probability distribution function with a unit hypercube support, so for some parameter $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^k$ and random variables $u_i \in [0, 1]$, $C(u_1, u_2, \dots, u_p | \boldsymbol{\theta}) : [0, 1]^p \mapsto [0, 1]$. The dimensions of a copula may be independent or dependent, but each dimension is a continuous uniform random variable $u_i \sim \mathcal{U}(0, 1)$, $i = 1, 2, \dots, p$, which has no free parameters. Thus, parameter $\boldsymbol{\theta}$ only provides information about joint variation, but nothing about marginal variation.

To motivate the relevance of copulas to describing across-trial variability, lets consider a problem of sampling hits and false alarms from a signal detection model with variable μ_s and σ_s for a sequence of variable signal stimuli. Formally, for a sequence of n signal stimuli, responses r are generated hierarchically according to

$$\begin{aligned} (\mu_s^{(i)}, \sigma_s^{(i)})^T &\sim F(\boldsymbol{\theta}) \\ r^{(i)} &\sim f(r | \mu_s^{(i)}, \sigma_s^{(i)}, \tau), \end{aligned}$$

where F is an unknown distribution and f is a Bernoulli probability mass function. One defining feature of F is how μ_s and σ_s co-vary. The properties of copulas stated above suggest that copulas are a complete description of statistical dependencies between standardized variables. A hypothesis of negative association between μ_s and σ_s , whether linear or nonlinear form, could then be expressed as a copula. However,

a sample from a copula is not enough to get a sample of (μ_s, σ_s) because their scales differ from unit interval.

Given a sample from a copula, we need to somehow map the standardized $[0, 1]$ dimensions of a copula onto \mathbb{R} and $(0, \infty)$ scales, and change continuous uniform distributions into desired marginal distributions, say normal and gamma . A useful theorem from probability theory, probability integral transform theorem, says that a univariate random variable $x \sim F$ relates to a uniform random variable via an equation $x = F^{-1}(u)$ (Casella & Berger, 2002). As a consequence of this fact, if we had a random sample of standard uniform variables $u_1, u_2, \dots, u_n \sim \mathcal{U}(0, 1)$ and arbitrary distribution F , then we could get a random sample of x by applying $x = F^{-1}(u)$.

Going back to the signal detection example, assume we want μ_s to be normally distributed and σ_s to be gamma distributed. Then, using the probability integral transform theorem, we could transform a random sample from a copula C , encoding a desired form of negative correlation between standardized variables, into μ_s and σ_s^2 by applying normal and gamma marginal distributions to the standardized variables. A random sample of μ_s and σ_s could then be used to obtain predictions of hits and false alarms for n signal stimuli according to the hierarchical scheme.

The prediction of responses under variable signal stimuli suggests that with a copula encoding negative associations, and normal and gamma marginal distributions of parameters, we could obtain a joint distribution of parameters that encodes our hypothesis about across-trial variability. The basis for the connection between copulas and multivariate distributions derives from Sklar's 1959 two-part representation theorem:

Theorem. Let F be a multivariate distribution with continuous marginal distributions F_1, F_2, \dots, F_p , then there exists a unique copula C such that

$$F(x_1, x_2, \dots, x_p) = C(F_1(x_1), F_2(x_2), \dots, F_p(x_p)). \quad (3.12)$$

Conversely, given a copula C and marginal univariate distributions F_1, F_2, \dots, F_p , F is a multivariate distribution.

A corollary of Sklar's theorem, whose importance will be explained below, is

Corollary 1. Let F be a p -dimensional distribution function with continuous quantile functions $F_1^{-1}, F_2^{-1}, \dots, F_p^{-1}$, then for every $(u_1, u_2, \dots, u_p) \in [0, 1]^p$ a copula

$$C(u_1, u_2, \dots, u_p) = F(F_1^{-1}(u_1), F_2^{-1}(u_2), \dots, F_p^{-1}(u_p)). \quad (3.13)$$

The forward implication of Sklar's theorem means that every multivariate distribution can be rewritten in terms of a copula and univariate marginals. One way to think about multivariate distributions is that they simultaneously describe joint and marginal variation of a random vector. The right hand side of Sklar's equation is a combination of a copula and marginal distributions. On its own, a copula carries no information about the range of each dimension of a random vector or the scale, shape and location of each variable. Hence, a copula only describes joint variation. On the other hand, a univariate distribution function defines the range of a variable and other characteristics through its parameters, thus specifying marginal variation.

Let's examine how the forward implication applies to the joint distribution of hits and false alarms of the standard signal detection model by deriving its copula. By definition, the joint distribution function of the number of hits x_h and false alarms x_{fa}

$$F(x_h, x_{fa}) = P\{x_h \leq k_h, x_{fa} \leq k_{fa}\}, \quad (3.14)$$

where $k_h = 0, 1, \dots, N_h$, $k_{fa} = 0, 1, \dots, N_{fa}$. We can further rewrite the joint distribution as

$$\sum_{x_h=0}^{k_h} \sum_{x_{fa}=0}^{k_{fa}} p(x_h, x_{fa}) = \sum_{x_h=0}^{k_h} \sum_{x_{fa}=0}^{k_{fa}} p_h(x_h) p_{fa}(x_{fa}) = \sum_{x_h=0}^{k_h} p_h(x_h) \sum_{x_{fa}=0}^{k_{fa}} p_{fa}(x_{fa}), \quad (3.15)$$

which simplifies to

$$F_h(x_h) F_{fa}(x_{fa}). \quad (3.16)$$

By Sklar's theorem, the copula for the signal detection model is

$$C(u_1, u_2) = u_1 u_2, \quad (3.17)$$

where $u_1, u_2 \in [0, 1]$. The functional form of C says that joint variation between hits and false alarms can be described as a uniform distribution over a unit square, implying that knowing the number of hits tells us nothing about the number of false alarms. This follows from the standard assumption of independent responses in signal detection theory and is reflected in the special case of a n-dimensional independence copula.

The derivation of the independence copula from a bivariate distribution of hits and false alarms shows how Sklar's theorem is also a statement about how copulas can be obtained from known multivariate distributions. Once a copula is derived, we can study joint variation among random variables while ignoring that, marginally, they may be binomial or otherwise. In contrast, the converse implication of Sklar's theorem takes us from a copula and marginal distributions to a related multivariate distribution. The practical consequence is that if we can specify joint variation with a copula and marginal variation with some univariate distributions, then Sklar's theorem gives us a way to define new multivariate distributions.

To demonstrate the usefulness of the converse implication, consider two alternatives of the standard signal detection model that incorporate variability in parameters of a signal representation. To specify two models we need copulas and marginal distributions to characterize variation in μ_s and σ_s . As a first candidate to describe negative correlation between the parameters, consider a Frank copula

$$C_F(u_1, u_2) = -\frac{1}{\theta_F} \ln \left(1 + \frac{(\exp(-\theta_F u_1) - 1)(\exp(-\theta_F u_2) - 1)}{(\exp(-\theta_F) - 1)} \right), \quad (3.18)$$

where $\theta_F \in \mathbb{R} \setminus \{0\}$ is a dependency parameter. A sample from a Frank copula can be seen in Figure 3.1. The characterizing feature of a Frank copula is that u_1 and u_2 co-vary over a rectangular-like contour.

An alternative to a Frank copula is a t copula

$$C_t(u_1, u_2) = \int_{-\infty}^{t_\nu^{-1}(u_1)} \int_{-\infty}^{t_\nu^{-1}(u_2)} \frac{1}{2\pi(1-\theta_t^2)^{1/2}} \left(1 + \frac{x^2 - 2\theta_t xy + y^2}{\nu(1-\theta_t)} \right)^{-(\nu+2)/2} dx dy, \quad (3.19)$$

where $\nu > 0$ is a degrees of freedom, $\theta_t \in [-1, 1]$ is a correlation parameter, $t_\nu^{-1}(\cdot)$ is a t quantile function, $x \in \mathbb{R}$ and $y \in \mathbb{R}$. Note that the t copula is obtained by applying Corollary 1 to the bivariate t distribution function, revealing the importance of the corollary for deriving new copulas. A sample from a t copula is shown in Figure 3.1. The t sample shows that dependence between u_1 and u_2 has an elliptical contour with extended tails.

By picking Frank and t copulas we can now describe joint variation in μ_s and σ_s . To complete defining models of across-trial variation we need two univariate distributions. For μ_s let's use a non-central t distribution F_t with degrees of freedom $\omega > 0$ and a shape parameter $\lambda \in \mathbb{R}$ that introduces a skew. The support of t covers the real number line, so it satisfies the range requirement for μ_s . As for σ_s , let's assign

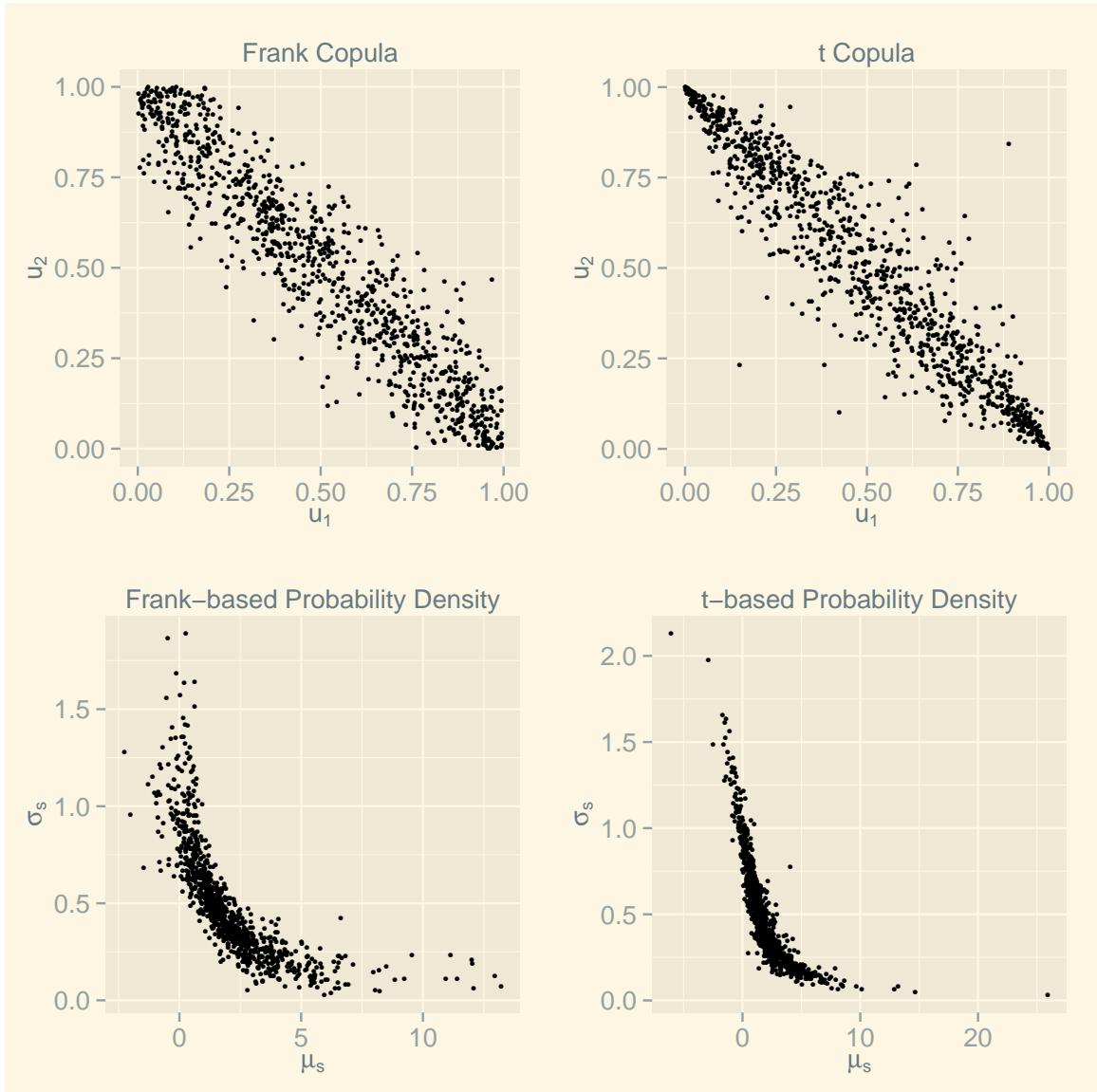


Figure 3.1: The top two panels show samples from a Frank copula and a t copula. Bivariate distributions resulting from applying a t and a gamma transformation to copula samples are in the bottom panel. Parameter settings: $\theta_F = -15, \theta_t = -0.95, \omega = 2, \nu = 3, \lambda = 1.5, \alpha = 3, \beta = 6$. Parameters result in approximately the same Kendall's measure of correlation.

a gamma distribution F_g , which has support over the positive real numbers, with a shape parameter $\alpha > 0$ and a scale parameter $\beta > 0$.

The two copulas can now be combined with the marginal distributions according to Sklar's theorem to give us two distributions of μ_s and σ_s . Model M_1 takes the form of

$$F_1(\mu_s, \sigma_s | \theta_F, \omega, \lambda, \alpha, \beta) = C_F(F_t(u_1 | \omega, \lambda), F_g(u_2 | \alpha, \beta) | \theta_F), \quad (3.20)$$

and model M_2 is

$$F_2(\mu_s, \sigma_s | \theta_t, \nu, \omega, \lambda, \alpha, \beta) = C_t(F_t(u_1 | \omega, \lambda), F_g(u_2 | \alpha, \beta) | \theta_t, \nu). \quad (3.21)$$

Samples from M_1 and M_2 are shown in Figure 3.1. M_1 , on the bottom left, induces a negative, monotonically decreasing relation that shows smooth co-variation between μ_s and σ_s . M_2 , on the bottom right, also induces a negative correlation, but has an abrupt shift in the “slope” after $\mu_s \approx 2$, suggesting two different states that a representation may occupy. Overall, Frank and t copulas result in two bivariate distributions of the signal representation properties that have a noise suppressing feature.

Using the two models we could examine effects of correlated parameters on hits and false alarms. Before showing how to integrate the copula-based distributions with the bivariate binomial mass function, I present another useful corollary of Sklar's theorem.

Corollary 2. *For a differentiable p -dimensional distribution function with continuous quantile functions $F_1^{-1}, F_2^{-1}, \dots, F_p^{-1}$, the probability density function*

$$\begin{aligned} f(x_1, x_2, \dots, x_p) &= \frac{\partial^p F(x_1, x_2, \dots, x_p)}{\partial x_1 \partial x_2 \cdots \partial x_p} \\ &= \frac{\partial^p C(F_1(x_1), F_2(x_2), \dots, F_p(x_p))}{\partial x_1 \partial x_2 \cdots \partial x_p} \\ &= c(F_1(x_1), F_2(x_2), \dots, F_p(x_p)) \prod_{i=1}^p f(x_i), \end{aligned} \quad (3.22)$$

where $c(F_1(x_1), F_2(x_2), \dots, F_p(x_p))$ is a probability density associated with a copula, and $f(x_i)$ are probability densities of each dimension.

One important consequence of Corollary 2 is that we can derive a copula density $c(F_1(x_1), F_2(x_2), \dots, F_p(x_p))$ by dividing the joint density $f(x_1, x_2, \dots, x_p)$ by the product of marginal densities $\prod_{i=1}^p f(x_i)$, revealing again that a copula is the factor that adds dependence between random variables. From the practical side, the implication is that we can define a new multivariate distribution by going directly to the probability density functions. This becomes useful when marginal distribution functions have no closed form (e.g. t distribution).

Going back to the signal detection example, we can now obtain a functional form for hits and false alarms under parameter variability. Let's concentrate on M_2 as a model of μ_s and σ_s . By Corollary 2, the density of M_2 is a product of a t copula density c with a non-central t density f_t and a gamma density f_g , so

$$\begin{aligned} f(\mu_s, \sigma_s \mid \theta_t, \omega, \lambda, \nu, \alpha, \beta) &= c(F_t(\mu_s \mid \lambda, \nu), F_g(\sigma_s \mid \alpha, \beta) \mid \lambda) \times \\ &\quad f_t(\mu_s \mid \lambda, \nu) f_g(\sigma_s \mid \alpha, \beta). \end{aligned} \quad (3.23)$$

Then, in an experiment where we observe N_s signal trials and N_n noise trials, with assumption of variable μ_s and σ_s according to M_2 , the probability mass function for

hits and false alarms is

$$\int_{-\infty}^{\infty} \int_0^{\infty} p(y_h | \mu_s, \sigma_s, \tau, N_s) p(y_{fa} | \tau, N_n) f(\mu_s, \sigma_s | \theta_t, \omega, \lambda, \nu, \alpha, \beta) d\mu_s d\sigma_s = \\ p(y_h | \theta_t, \omega, \lambda, \nu, \alpha, \beta, \tau, N_s) p(y_{fa} | \tau, N_n). \quad (3.24)$$

With two alternative models of hits and false alarms at hand, we could now fit real data and test for the assumption of a dynamic representation of signals. It would be of particular interest to obtain copula parameters. A visual summary of the best model could be captured by a graph of a copula to reveal what sort of joint variation is consistent with data.

In conclusion, copulas provide a useful and powerful way of defining new multivariate distributions with various patterns of dependencies which can be easily summarized with scalar quantities. The signal detection model generalization examined in this section serves as a direct example of how models of simple choice behavior can be generalized to have dependent across-trial variation in their parameters. In the next section, I apply the copula-based approach to propose two multivariate distributions for parameter variation that can be used to generalize the Ratcliff diffusion model.

3.3 Generalized Decision Models

Recall that one of the questions motivating this thesis is developing a model of across-trial variability of parameters of a sequential sampling model. Following the statistical approach, I present two new multivariate distributions that we can use to generalize the Ratcliff diffusion model. Resulting mixture models set the stage for answering the other questions guiding this thesis.

I assume the Wiener process as a description of the decision process underlying simple choice behavior. To avoid the inconvenient domain of the starting point, I work with the bias parameterization of the Wiener process where $\beta = \zeta/\alpha$, instead of ζ , fixes the starting value of accumulation. Three parameters are assumed to vary across trials: drift rate, bias and non-decision time. Our problem is to find some multivariate densities that describe the behavior of the three parameters, each of which falls on a different scale and has a different univariate distribution. Sklar's theorem tells us that to describe joint variation of the three parameters we need a copula and marginal probability density functions for each dimension.

I start with copulas. When searching for copulas I concentrated on families that were flexible and computationally tractable. I interpret flexibility as the ability of a copula to produce different correlations for each pair of parameters ranging from full negative to full positive dependence. Basically, we need copulas that can encode as many different types of correlation structures as possible. Within the set of existing copulas, many copulas, regardless of the dimension, are governed by a single parameter that induces a particular kind of dependence (Joe, 1997; Nelsen, 2006). For example, Archimedean copulas are controlled by a single parameter and encode the same lower tail, upper tail, or all-positive dependence for all pairs of variables. The Frank copula presented in the previous section is an example. Archimedean and other restricted copulas are inadequate because there is no prior information available about the correlation structure of parameters, so it is sensible to start modeling with most flexible copulas.

The elliptical class of copulas satisfies the flexibility and tractability criterion. Elliptical copulas arise from n-dimensional elliptical distributions that can be defined

in terms of their probability density

$$f(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \det(\boldsymbol{\Sigma})^{-1} g((\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})), \quad (3.25)$$

where $\det(\cdot)$ is a determinant, g is a non-negative function on the positive real numbers, $\boldsymbol{\mu} \in \mathbb{R}$ is a location parameter, and $\boldsymbol{\Sigma}$ is a positive definite matrix of scale and correlation parameters (Joe, 1997; Nelsen, 2006). The term “elliptical” refers to the quadratic form (the argument of $g(\cdot)$) that causes iso-density contours to take the shape of a hyper-ellipsoid. From this large class, I use normal and t copulas.

Consider a normal copula first. Taking the three dimensional standard normal distribution function $\Phi_3(\mathbf{x} \mid \mathbf{P}_n)$, $\mathbf{x} \in \mathbb{R}^3$, with a positive definite correlation matrix

$$\mathbf{P}_n = \begin{bmatrix} 1 & \rho_{1,2} & \rho_{1,3} \\ \rho_{1,2} & 1 & \rho_{2,3} \\ \rho_{1,3} & \rho_{2,3} & 1 \end{bmatrix},$$

and the standard normal quantile function $\Phi_i^{-1}(u_i)$, $u_i \in [0, 1]$, Corollary 1 (Equation 3.13) implies that the normal copula

$$C_n(\mathbf{u} \mid \mathbf{P}_n) = \Phi_3(\Phi_1^{-1}(u_1), \Phi_2^{-1}(u_2), \Phi_3^{-1}(u_3) \mid \mathbf{P}_n). \quad (3.26)$$

Thus, the normal copula equals

$$\int_{-\infty}^{\Phi^{-1}(u_1)} \int_{-\infty}^{\Phi^{-1}(u_2)} \int_{-\infty}^{\Phi^{-1}(u_3)} \det(2\pi \mathbf{P}_n)^{-1/2} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{P}_n \mathbf{x}\right) d\mathbf{x}, \quad (3.27)$$

where $\mathbf{x} \in \mathbb{R}^3$. By Sklar’s theorem, coupling the normal copula $C_n(\mathbf{u} \mid \mathbf{P}_n)$ with three normal marginals $F_i(x_i)$ will result in the multivariate normal distribution function $C(F_1(x_1), F_2(x_2), F_3(x_3) \mid \mathbf{P}_n)$, but substituting some other univariate marginals $G_i(x_i)$ into the normal copula will result in a novel probability distribution $C_n(G_1(x_1), G_2(x_2), G_3(x_3) \mid \mathbf{P}_n)$.

Derivation of a t copula is also a consequence of applying Corollary 1. Given a 3-dimensional t cumulative distribution function $T_3(\cdot)$ and a t quantile function $t_i^{-1}(u_i)$, the t copula

$$C_t(\mathbf{u} \mid \mathbf{P}_t, \omega) = T_3(t_1^{-1}(u_1 \mid \omega), t_2^{-1}(u_2 \mid \omega), t_3^{-1}(u_3 \mid \omega) \mid \mathbf{P}_t), \quad (3.28)$$

where \mathbf{P}_t is a positive definite correlation matrix like \mathbf{P}_n , and $\omega > 0$ is a degrees of freedom parameter. Definitions of $T_3(\cdot)$ and $t_i^{-1}(\cdot)$ imply that the t copula equals

$$\int_{-\infty}^{t^{-1}(u_1 \mid \omega)} \int_{-\infty}^{t^{-1}(u_2 \mid \omega)} \int_{-\infty}^{t^{-1}(u_3 \mid \omega)} \mathcal{G}([\omega + 3]/2) / [\mathcal{G}(2\omega)\sqrt{\det(\pi\omega\mathbf{P}_t)}] \times \\ (1 + \mathbf{x}^T \mathbf{P}_t^{-1/2} \mathbf{x}/\omega)^{(\omega+3)/2} d\mathbf{x}, \quad (3.29)$$

where $\mathcal{G}(\cdot)$ is a univariate gamma function and $\mathbf{x} \in \mathbb{R}^3$. Unlike the normal copula, the t copula has a degrees of freedom parameter ω that results in heavier tails. The parameter ω introduces additional flexibility that controls the probability of extreme values in the tails, resulting in positive tail dependence, and making combinations of parameters orthogonal to the correlation structure more probable.

As an example of orthogonality, supposing that correlation between two parameters is positive, it is unlikely that we might observe a pair of values where one point has high magnitude and the second point has low magnitude. However, the occurrence of such rare events is predicted for finite values of ω with considerably higher probability than for the normal copula. When $\omega \rightarrow \infty$, the t copula converges to the normal copula (Joe, 1997). Thus, a model of parameter variation based on the t copula is a further generalization of the normal copula that allows modeling correlations between processing components that sometimes take unusual combinations or extreme values in the tails.

To complete definitions of two models of parameters, Sklar's theorem requires that we specify marginal distributions for each parameter. Because I am modeling parameter dependencies in a statistical manner and variation of parameters is unobservable, I limited my search to univariate distributions that exhibit flexible shapes to enable good approximations of the true shapes of parameter densities. In contrast, if I were following a theoretical approach, I would want to produce marginal variation that is derived theoretically and disallow all other distributional shapes.

For the drift rate δ , consistent with the Ratcliff model, I assume a normal distribution. I replace Ratcliff's continuous uniform assumption for the non-decision parameter τ^{nd} with a gamma distribution, and place a beta distribution on the bias parameter β . The univariate marginals for the three parameters are therefore

$$\begin{aligned}\delta &\sim \mathcal{N}(\nu, \eta^2), \\ \beta &\sim \mathcal{B}(\lambda, \gamma^2), \text{ and} \\ \tau^{nd} &\sim \mathcal{G}(\chi, \phi^2),\end{aligned}\tag{3.30}$$

where $\nu \in \mathbb{R}$, $\lambda > 0$, $\chi > 0$ are mean parameters and $\eta^2 > 0$, $\gamma^2 > 0$ and $\phi^2 > 0$ are variances.

Now we can combine specified copulas and univariate marginals. Applying Sklar's theorem leads to $i = 1, 2$ new multivariate distributions of the form

$$F_i(\delta, \beta, \tau^{nd}) = C_i(F_1(\delta), F_2(\beta), F_3(\tau^{nd})),\tag{3.31}$$

from which, by Corollary 2, we can obtain probability densities by

$$f_i(\delta, \beta, \tau^{nd}) = c_i(F_1(\delta), F_2(\beta), F_3(\tau^{nd})) f_1(\delta), f_2(\beta), f_3(\tau^{nd}).\tag{3.32}$$

The copula density and marginal densities of parameters have known closed-form expressions (Casella & Berger, 2002; N. L. Johnson, Kotz, & Balakrishnan, 1994,

1995; Nelsen, 2006). Direct application of Sklar's theorem ensures that both models of parameter variability are valid distributions or densities.

This completes the generalization of the Ratcliff diffusion model of simple decisions. We can now proceed to answering the two remaining questions motivating this thesis.

Chapter 4: Theoretical and Empirical Studies

With models of behavior defined, I can now outline two studies carried out to answer the remaining questions. Recall that I sought to compare predictions of the models and analyze the nature of the correlation structure consistent with the benchmark data. In Study A, I explored effects of parameter correlations on models' predictions under an experimental design similar to Ratcliff and Rouder (1998). Independent and both elliptical models were used. The objective was to compare predictions of mean sample paths, and commonly found features of response and response times that follow from their joint densities. The results and discussions are in Section 4.2 and 4.3.

Study B applied the independent and normal copula models to the benchmark dataset. Posterior densities of parameters using adaptive MCMC were obtained. The goal was to infer the directions and plausible magnitudes of correlation parameters. The results and discussion are in the Section 4.4.

Before delving into prediction studies, I first describe experimental paradigm and parameter settings under which predictions were obtained.

4.1 Design and Parameter Settings

When selecting parameters, I followed the principle that predictions should be obtained for a particular task under realistic experimental conditions as reflected in the usually obtained model parameters. Because models were also planned to be fit to the benchmark dataset, it was most natural to explore predictions for the underlying brightness discrimination task assuming various levels of correlations.

Recall that the Ratcliff and Rouder (1998) benchmark dataset was generated by a two-choice free response perceptual task under 33 brightness levels and speed-accuracy instructions. They had 3 participants, but they showed similar behavior and parameters Ratcliff and Rouder (1998); Vandekerckhove and Tuerlinckx (2007); Vandekerckhove et al. (2011), so I concentrated on a single individual. To calculate predictions, we need parameter values that reflect experimental manipulations experienced by the individual. One approach could be to estimate parameters representing effects of manipulations from a range of plausible values that were recovered over many experiments (Matzke & Wagenmakers, 2009). However, previous fits of the benchmark dataset, both Bayesian (Vandekerckhove et al., 2011) and non-Bayesian (Ratcliff & Rouder, 1998), provide plausible parameter values. I chose the means of posterior densities from (Vandekerckhove et al., 2011) to calculate predictions.

Bayesian modeling done by Vandekerckhove et al. (2011) made certain assumptions about how experimental manipulations influence parameters. In their best-fitting model, the authors assumed that brightness level affects only mean drift rate and speed-accuracy instructions affect drift rate, boundaries and starting position, with the other parameters remaining constant across the experimental sessions. The assumptions also agree with many previous reports of best fitting models (Ratcliff &

McKoon, 2008; Wagenmakers, 2009). I calculated predictions under these assumptions, too.

For the drift rates, Vandekerckhove et al. (2011) provided a minimal front-end model of how the rate of evidence accumulation arises from brightness levels under particular instructions. The formal relation is described by the Weibull function. For instruction $i \in \{1, 2\}$ and brightness level $j \in \{1, 2, \dots, 11\}$, I assumed mean drift rate regresses on brightness level through a sigmoidal-shaped Weibull function

$$\nu_{i,j} = \nu_i^{lo} + (\nu_i^{hi} - \nu_i^{lo}) \left(1 - \exp \left(-[p_j / \nu_i^{sc}]^{\nu_i^{sh}} \right) \right), \quad (4.1)$$

where $\nu_{i,j}$ is the drift rate, p_j is the brightness proportion, ν_i^{lo} is the lower asymptote, ν_i^{up} is the upper asymptote, ν_i^{sc} is the scale parameter and ν_i^{sh} is the shape parameter. The estimates of Weibull parameters taken from Vandekerckhove et al. (2011) are in the Table 4.1.

Instruction	ν_{lo}	ν_{hi}	ν_{sh}	ν_{sc}
Accuracy	-0.352	0.329	4.413	0.526
Speed	-0.565	0.511	5.227	0.521

Table 4.1: Weibull function parameters used to obtain predictions in study A. The values were reported by Vandekerckhove et al. (2011).

The Weibull function provides a parsimonious way of connecting 66 experimental conditions to drift rates via 8 parameters. However, for obtaining predictions I did not use all 66 conditions. Based on previous fits to the benchmark dataset (Ratcliff & Rouder, 1998; Vandekerckhove et al., 2011), and more commonly observed in psychophysical investigations (Ratcliff, 2014; Vickers, 1979), drifts and discriminability

performance level off for extremely bright or dark stimuli. The drift rates change sigmoidally as a function of brightness level and are most informative for proportions from 0.2 to 0.8. In addition, light and bright stimuli lead to approximately symmetric performance, thus requiring proportions only above or below 0.5. Based on these considerations, I used 6 white-to-black pixel proportions for the speed condition, $\{0.505, 0.520, 0.540, 0.565, 0.605, 0.740\}$, and 6 proportions for the accuracy condition, $\{.496, .502, .510, .520, .528, .540\}$. The numbers were calibrated to produce accuracy rates of $\{0.55, 0.62, 0.69, 0.76, 0.83, 0.90\}$.

The other non-correlation parameters are listed in the Table 4.2. The values for thresholds α_i , drift rate standard deviation η , non-decision mean χ and standard deviation ϕ are taken directly from Vandekerckhove et al. (2011). Mean bias λ_i and standard deviation of bias γ are transformations of end point estimates of a uniform random variable because I assumed a beta density whereas Vandekerckhove et al. (2011) assumed a continuous uniform density.

Instruction	α	η	λ	γ	χ	ϕ
Accuracy	0.221	0.127	0.464	0.065	0.279	0.041
Speed	0.050	0.127	0.464	0.065	0.279	0.041

Table 4.2: Non-correlation parameters used to obtain predictions in study A. The values were reported by Vandekerckhove et al. (2011). Starting point values were transformed to decision bias values.

The remaining parameters are the degrees of freedom for the t copula and correlation parameters that characterize generalized diffusion models. I fixed the degrees of freedom ω to 3 for all simulations, which gives a t copula having all its moments

(not true for $\omega < 3$). In particular, $\omega = 3$ avoids the lack of Law of Large Numbers for the Cauchy distribution ($\omega = 1$) while ensuring that the t copula has maximal tail dependence, which distinguishes it from the normal copula (Joe, 1997; Nelsen, 2006).

Lastly, I used three correlation structures with low, medium and high levels of correlation between parameters, which also stayed the same throughout the experiment. The patterns are given in the Table 4.3. The structures reflect a few of many possible correlation patterns that could exist under typical experimental conditions. For example, based on the pattern of signs, sets 1 - 3 can be interpreted as a tendency of decision bias β and evidence δ to agree on the best response while the non-decision τ^{nd} reflects motor bias towards the other response. Thinking through the behavioral consequences of other correlation structures, similar interpretations can be proposed, but these are only suggestive and were used to select the three proposed correlation patterns out of the infinitely many possibilities without any theoretical commitment.

Set	$\rho_{\delta,\beta}$	$\rho_{\delta,t^{nd}}$	$\rho_{\beta,t^{nd}}$
1	0.15	-0.15	-0.15
2	0.50	-0.50	-0.50
3	0.85	-0.85	-0.85
4	0.15	0.15	0.15
5	0.50	0.50	0.50
6	0.85	0.85	0.85
7	-0.15	-0.15	0.15
8	-0.50	-0.50	0.50
9	-0.85	-0.85	0.85

Table 4.3: Correlation parameters used to obtain predictions in study A. The values were picked to range from low to high and represent three possible correlation patterns. For sample paths only column one matters, especially rows 4 - 6. Predicting response and response times depends on all the values.

This combination of the used parameters should lead to interpretable comparisons between the three models. All calculations were done under the same parameter values for all the non-correlation parameters, and correlation parameters were the same for the two copula-based models. The only differences between the models are the copula assumptions. The Ratcliff diffusion model has the independence copula and generalizations have the normal and t copulas.

In conclusion, the used set of parameters was intended to enable unambiguous conclusions about how the three models differ in their predictions based solely on their parameter dependence assumptions. The next section gives an overview of Study A1 before delving into the details of mean sample path predictions.

4.2 Study A1 - Mean Sample Paths

Study A1 examined predicted mean sample paths of the standard Ratcliff diffusion models and its generalizations, described in Section 3.3, as representations of the evidence accumulation process. The interest in the mean sample path is that it characterizes dynamics of the evidence accumulation process and could be related to neural data. This study also fits with a recent trend of combining process models with cognitive neuroscience data Forstmann and Wagenmakers (2015); Forstmann, Wagenmakers, Eichele, Brown, and Serences (2011); Mulder et al. (2014); Turner, Forstmann, et al. (2013).

Recent evidence from single-cell recordings of regions of monkey brains involved in rapid perceptual decisions suggests that accumulation process is represented by the spike rates of a weakly correlated ensemble of neurons (Bogacz, 2007; Gold & Shadlen, 2007; Heitz & Schall, 2013; Ratcliff et al., 2007; Shadlen & Newsome, 1998; Zandbelt

et al., 2014). Under the assumption that accumulated evidence is represented by the spike rate, examining how the mean trajectory of accumulation changes under different correlation regimes could suggest an explanation for features of spike rate functions observed in experiments with no explicit manipulation of correlations and give testable predictions for experiments manipulating correlations.

While connecting the dynamics of accumulation specified by the diffusion models is an exciting prospect, there are crucial limitations to the standard Ratcliff diffusion model, which carries over into its generalization explored in this thesis (Heitz & Schall, 2013). Only the decision process is described in detail while other processes are represented solely through a non-decision time constant. The modeling assumptions have been motivated by behavioral data, and are often sufficient for adequate fits of it. However, the models would require considerable expansion to be compared to the spike rate functions obtained by single cell recording in a complete, quantitative manner.

One problem of connecting the diffusion model to the time course of spike rate is temporal alignment. During the trial, several events of interest, among others, include stimulus presentation, accumulation initiation, reaching the peak of accumulation, and motor response. The implanted electrodes track activity in buildup neurons during the whole trial and their spike rate can be averaged by aligning the record of temporally variable activity on either of the three events. Without further assumptions, the diffusion models can be connected to the spike rate by alignment at accumulation initiation or peak of accumulation. The model says nothing about the rest of the trial, and so cannot be aligned on stimulus or response.

The second problem is aggregating sample paths generated by the bounded diffusion process across trials. During the trial, buildup neurons begin activity at the baseline, then undergo accumulation of activity until a threshold, and then diminish in activity back towards the baseline. The mean spiking rate, at a given time point, is based on activity that is either still climbing towards a peak or is declining towards the baseline.

In contrast, diffusion models have a baseline vary across trials, followed by accumulation, followed by absorption, followed by a reset to another initial state during the next trial. The mean accumulated evidence is based on still accumulating processes and absorbed processes, but not declining processes. This necessitates a distorted prediction, with distortion growing with time as more processes are absorbed and contribute to the average.

To sum up, diffusion models allow for decision or peak accumulation alignment, but with the mean sample paths becoming more distorted relative to spike rates of buildup activity because there is no decay to the baseline process. Even with these limitations, we can still obtain useful information from mean sample paths by considering which part of the accumulation paths are altered by inclusion of the correlation parameter relative to the independent diffusion model. As an example, consider explorations of the standard Ratcliff diffusion model by Ratcliff et al. (2003). By simulating and aggregating only evolving sample paths, authors obtained a predictions that the decision process tends to remain near the starting value for a substantial part of the decision until it rapidly moves towards a boundary, a pattern common across decision speeds. This qualitative prediction was consistent with recordings from buildup neurons in the superior colliculus.

Guided by the example of Ratcliff et al. (2003), study A1 intended to explore how different features of the predicted sample paths change with a correlation between starting point and drift rate. In the next subsection, I define mathematically the mean sample path, calculation of which is an analytically intractable problem that requires an approximate integration approach, presented in the same subsection. After, I present and describe predicted mean sample paths for the three models, and discuss their implications.

4.2.1 Mean Sample Paths Calculation

To motivate the exact calculations I obtained I restate two facts. From previous studies (Ratcliff & McKoon, 2008; Ratcliff & Rouder, 1998), we know that parameters assumed to vary across trials can induce asymmetries in the predicted response time distribution for the two responses, and that asymmetries may differ depending on the quantile. This suggests that accumulation trajectories underlying the reaction times for different choices are asymmetric and asymmetries depend on when the accumulation ends. Hence, to understand effects of correlated parameters on the mean sample path, it is important to condition the mean on the response and a finishing time cutoff. The functions I sought to obtain for each model $i \in \{1, 2, 3\}$ are

$$E_i[x(t) | r, \alpha, \boldsymbol{\theta}_i, t_c], \quad (4.2)$$

$t_c \in \{Q(0.35), Q(0.70)\}$ is the time cut-off by which a process has to reach the absorption time defined in terms of quantiles $Q(\cdot)$, and $\boldsymbol{\theta}_i$ are the parameters resulting from averaging across trial parameters with respect to the copula-based mixing density. Note that $\boldsymbol{\theta}_i$ characterizes only bias and rate of evidence accumulation because the non-decision time is irrelevant during the decision formation.

The mean sample paths implied by the three models are not available in explicit form. However, we can still obtain predictions by realizing that the mean sample path is an integral problem at each time point. Consider the value of the mean sample path with fixed parameters at some time t . Then, the mean sample path

$$\mathbb{E}_i[x(t) \mid r, \alpha, \boldsymbol{\theta}_i, t_c] = \int_{\mathcal{S}} x(t) f(x(t) \mid r, \alpha, \boldsymbol{\theta}_i, t_c) dx(t), \quad (4.3)$$

where $\mathcal{S} = [0, \alpha]$ is the bounded state space and $f(x(t) \mid r, \alpha, \boldsymbol{\theta}_i, t_c)$ is the density of $x(t)$ conditioned on its starting value. Because $f(x(t) \mid r, \alpha, \boldsymbol{\theta}_i, t_c)$ is unavailable, we cannot use numerical integration. However, the problem could be solved if we could sample from the density and then use a Monte Carlo estimator to approximate the sample path at time t .

The Monte Carlo estimator appropriate for this problem is a consequence of a law of large numbers for independent, but non-identically distributed random variables with finite variance (Casella & Berger, 2002). For a random sample $x_1(t), x_2(t), \dots, x_N(t) \sim f(x(t) \mid r, \alpha, \boldsymbol{\theta}_i, t_c)$ at time t , with expected value $\mathbb{E}[x(t)] < \infty$,

$$\frac{1}{N} \sum_{i=1}^N x_i(t) \cong \mathbb{E}[x(t) \mid r, \alpha, \boldsymbol{\theta}_i, t_c], \quad (4.4)$$

when N is large.

Hence, to obtain a Monte Carlo estimate of the mean sample path we need samples from the marginal density over a grid of time points. One way to obtain such samples is to simulate the Wiener process until it hits an absorbing boundary. Each simulated path provides a draw from the marginal density at each time point of the grid. Because sample paths are generated independently, values of the process at time t make an independent sample. Hence, assumptions for the Monte Carlo estimator hold for each time point.

To obtain the calculations I used a two-step Monte Carlo procedure. For the decision model $i = 1, 2, 3$,

1. Sample N draws of $(\delta, \beta) \sim f_i(\delta, \beta | \nu, \eta, \lambda, \gamma)$,
2. Sample N sample paths $x(t | \alpha, \delta, \beta)$,
3. Group the sample paths by the response r and cut-off t_c ,
4. Average across drifting and absorbed sample paths within each group.

Note that the aggregation procedure assumes alignment on the initiation of the accumulation process rather than the peak of accumulation. This concentrates attention on the segment of the mean sample path around the starting point where distortion from adding absorbed processes is least and effect of correlations are most interesting.

Computationally, the procedure for estimating mean sample paths is simple and fast. Simulation in Step 1 can be obtained with the base and copula packages in the R statistical environment (Hofert, Kojadinovic, Maechler, & Yan, 2013; R Core Team, 2014). Similarly for Step 2, there is a R wrapper RWiener for simulating a random walk approximation of the Wiener process using a C routine (Tuerlinckx, Maris, Ratcliff, & De Boeck, 2001; Wabersich, 2014).

To conclude, understanding of how correlated parameters affect the time course of a decision process can be revealed by comparing mean sample paths of the standard to generalized Ratcliff diffusion models. The exact calculation of $E[x(t) | r, \alpha, \theta_i, t_c]$ is intractable, but we can obtain it by simulating the bounded Wiener process with variable parameters and then averaging sample paths.

4.2.2 Results

Given the typical parameter values, I simulated 1×10^5 sample paths for each condition. Simulated samples paths are summarized as mean sample paths and organized in Figures 4.1 - 4.6 based on six values of correlation between decision bias β and rate of evidence accumulation δ . Each figure contains 12 subplots showing how decisions of different speed evolve on average under different experimental conditions. The left column corresponds to “speed” emphasis and the right column is for the “accuracy” emphasis. Each row is based on a brightness proportion that increases from top to bottom, with all proportions being above 0.5, hence biasing the “bright” decision. Within each subplot, mean sample paths absorbed at the upper boundary stand for “bright” decision and those absorbed at the lower boundary stand for “dark” decision. For each decision, mean sample paths are separated into fast, intermediate and slow groups. Lastly, colors distinguish the three copulas: red for independent, green for normal and blue for t.

Across the subplots of each figure, the mean sample paths of the three models show various complex regularities. The regularities arise under changes in several factors including speed-accuracy instructions, brightness level, correlation sign and magnitude, response and speed of a response. The data in the graphs represent a full-factorial design.

It is hard to discuss the trajectory as a whole and capture how it changes with adjustments in the independent variables, so to simplify, I concentrated on the initial states and curvature. Upon obtaining the results it turned out that absorption times can show large effects, so these are described, too. The choice of the three dependent variables is motivated by the fact that the correlation is between decision bias and

rate of accumulation, which respectively influence initial states and curvature of each trajectory that together determine when the process reaches the boundary.

To understand effects of correlated decision bias and rate of evidence accumulation, adjusted for other factors, we can use the standard Ratcliff model as a baseline condition to detect effects. Before describing the effects, I report an upper bound on the standard error of effects based on the obtained sample size, which is needed for judging reliability of the simulated results. Differences in rate of evidence accumulation is considered qualitatively, so standard errors have not been calculated. Using the selected parameters, the standard error of the mean decision bias, based on all sample paths, is $\sigma_\beta = 0.065/\sqrt{n}$. When sample paths are partitioned based on decision and speed, and brightness is highest (resulting in frequency of dark decisions to be about 0.1), standard errors of mean initial bias of all six paths are below $0.065/\sqrt{0.1n}$. Hence, with the simulated sample sizes, standard errors of mean initial states are below 0.001. The approximate standard error suggests that reliable difference between predictions of independent and elliptical copulas for the mean initial state are 0.002.

To understand effects of correlated parameters in the clearest form I concentrate detailed description of results on the highest correlation magnitude. Figure 4.3 summarizes results for $\rho_{\delta,\beta} = 0.85$. The independent copula, colored in red, provides the baseline. I present initial positions of the sample paths on the bias scale to remove the scale differences between speed-accuracy conditions. As a reference, the overall mean initial bias is 0.464.

Under “speed” and near-neutral stimuli, shown in subplot 31, fastest and intermediate bright decisions are greater than the mean bias, 0.483 and 0.465, respectively,

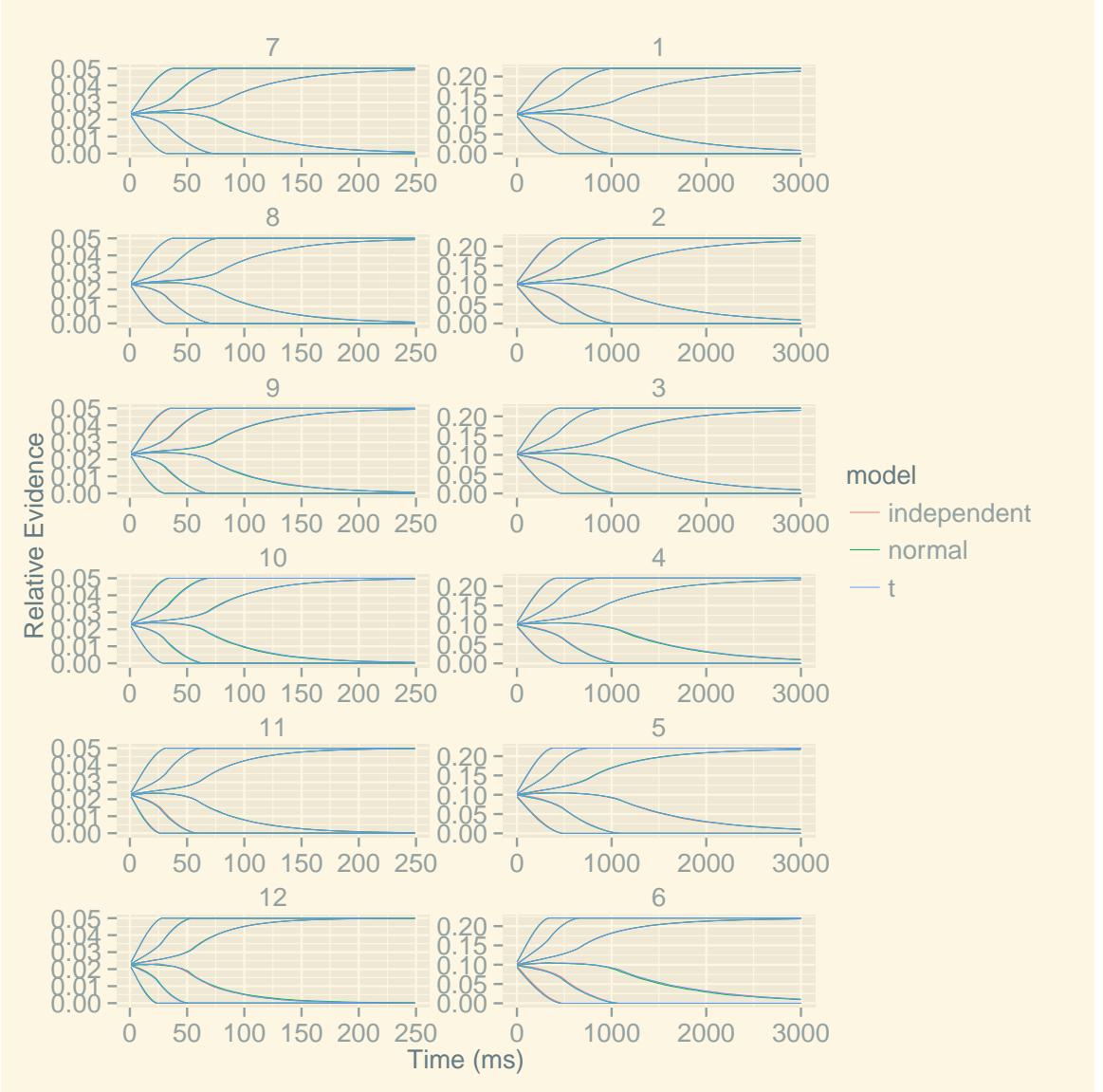


Figure 4.1: Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = 0.15$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions.

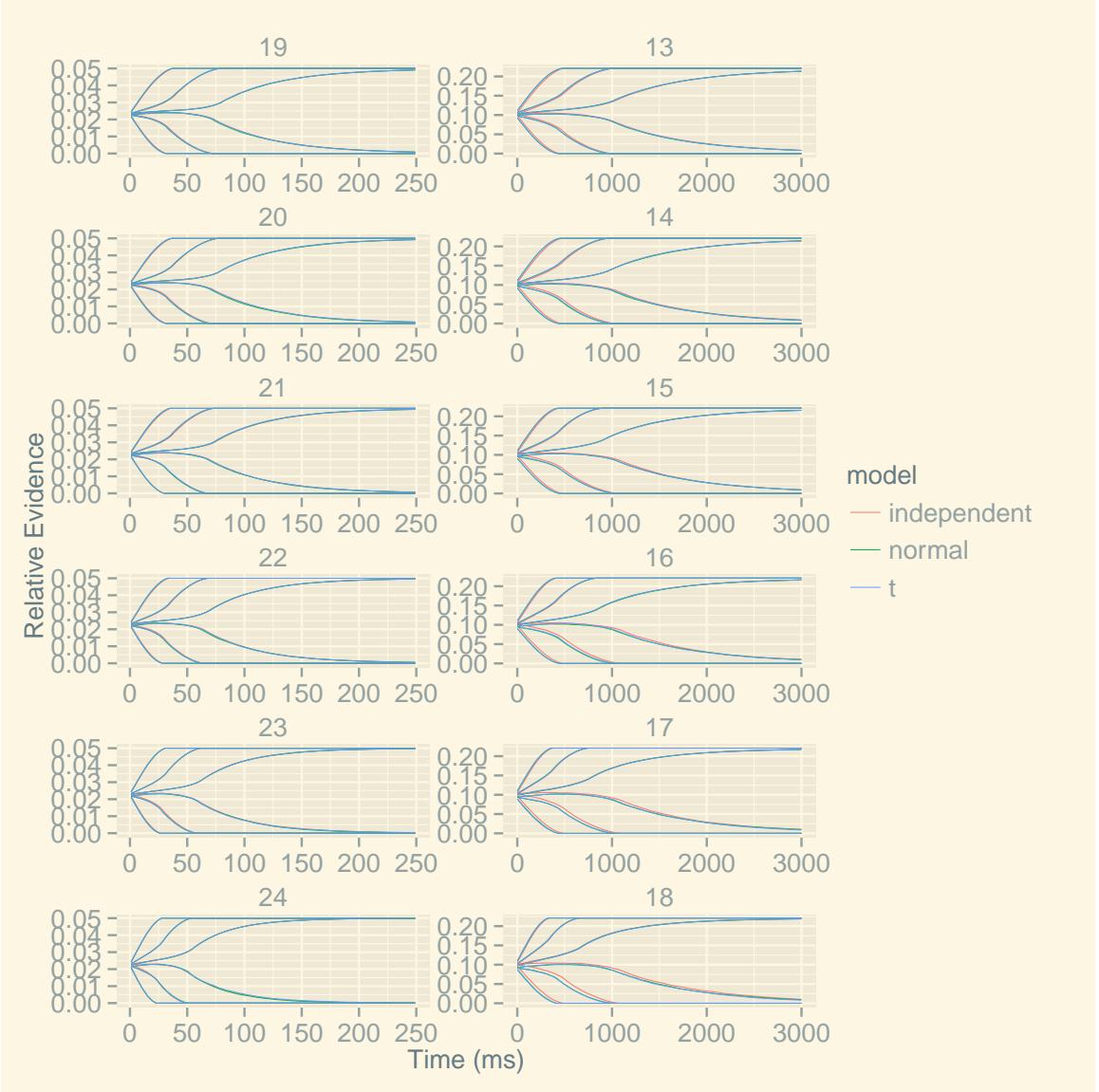


Figure 4.2: Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = 0.5$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions.

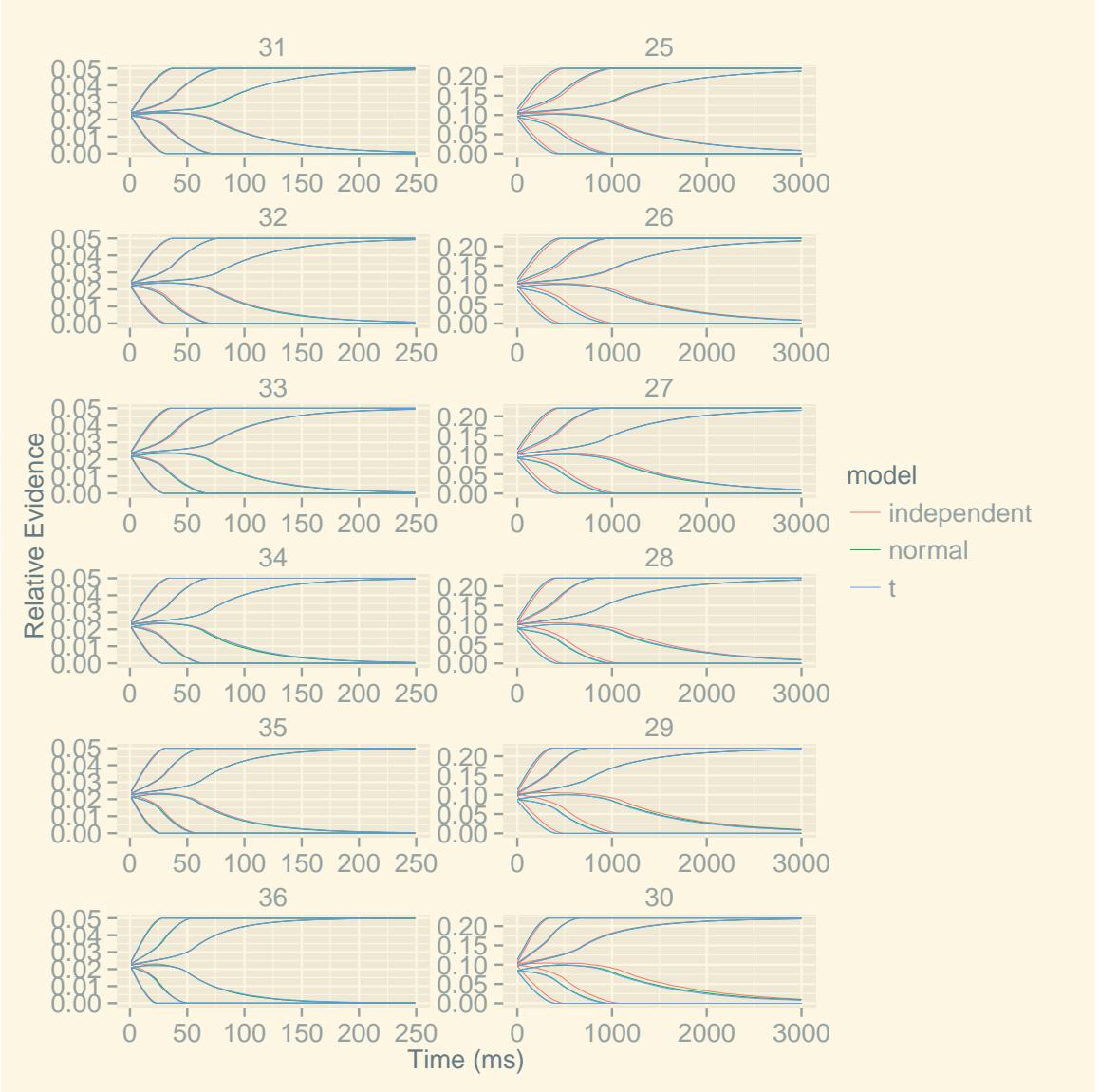


Figure 4.3: Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = 0.85$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions.

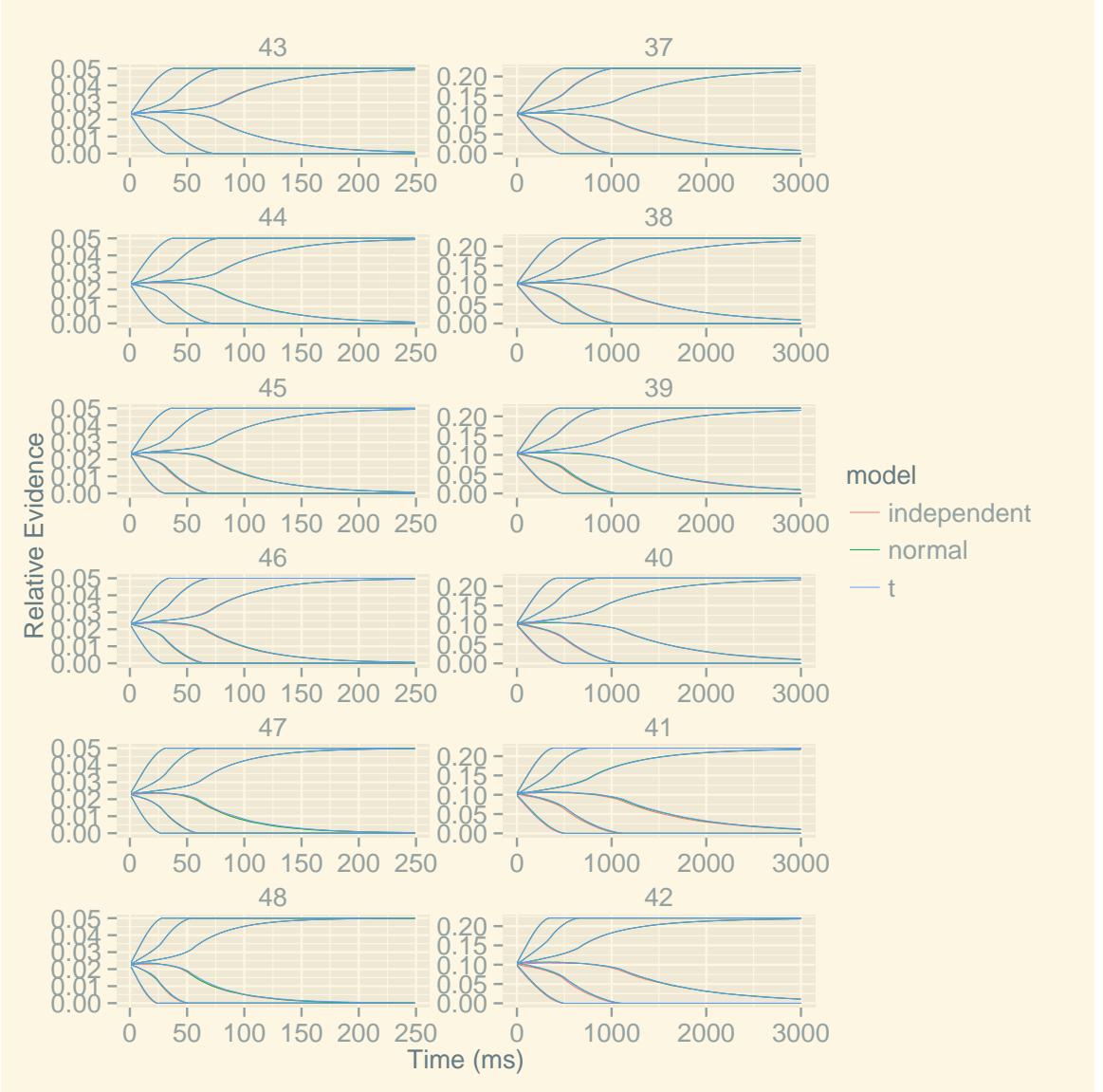


Figure 4.4: Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = -0.15$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions.

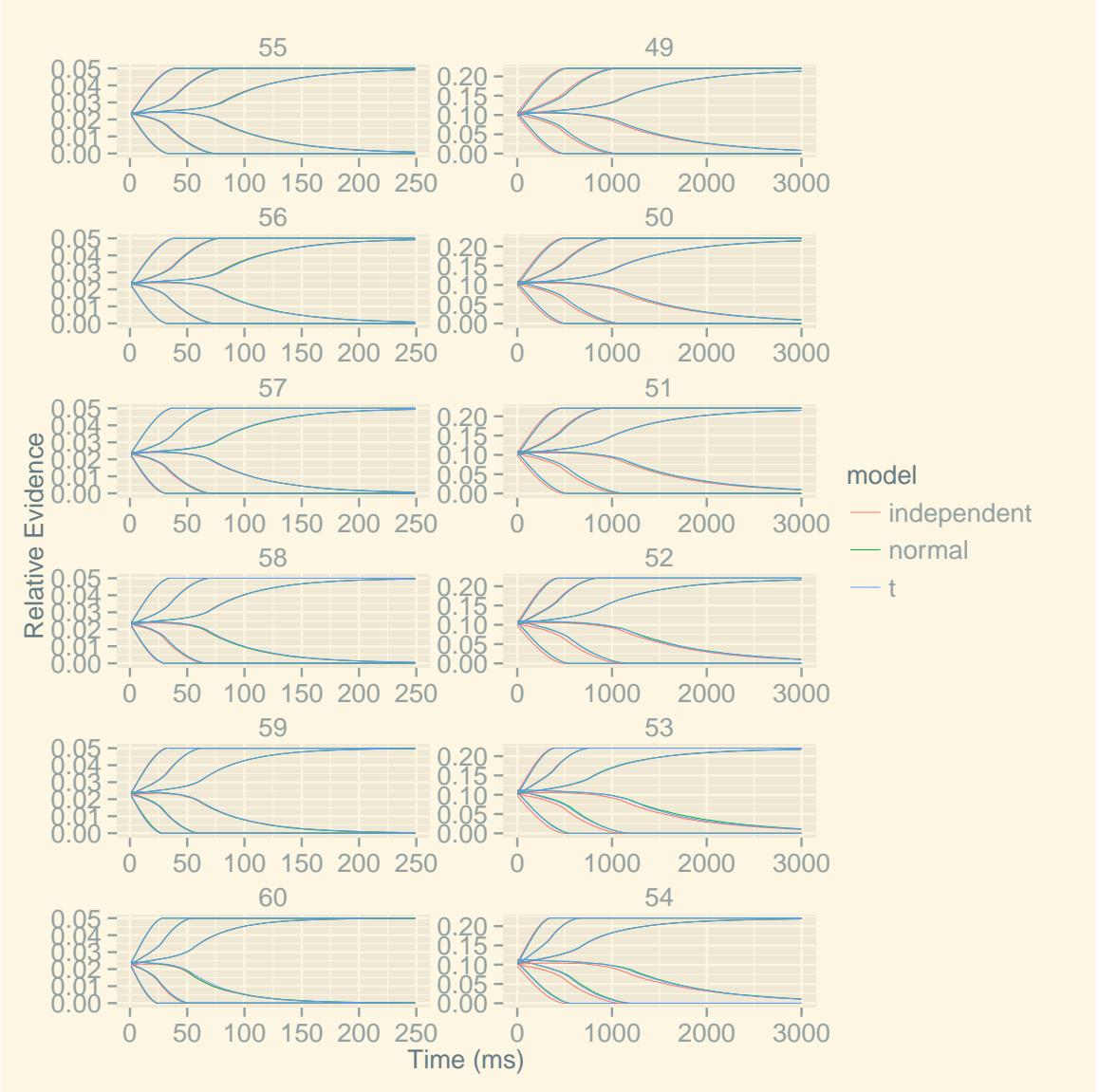


Figure 4.5: Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = -0.5$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions.

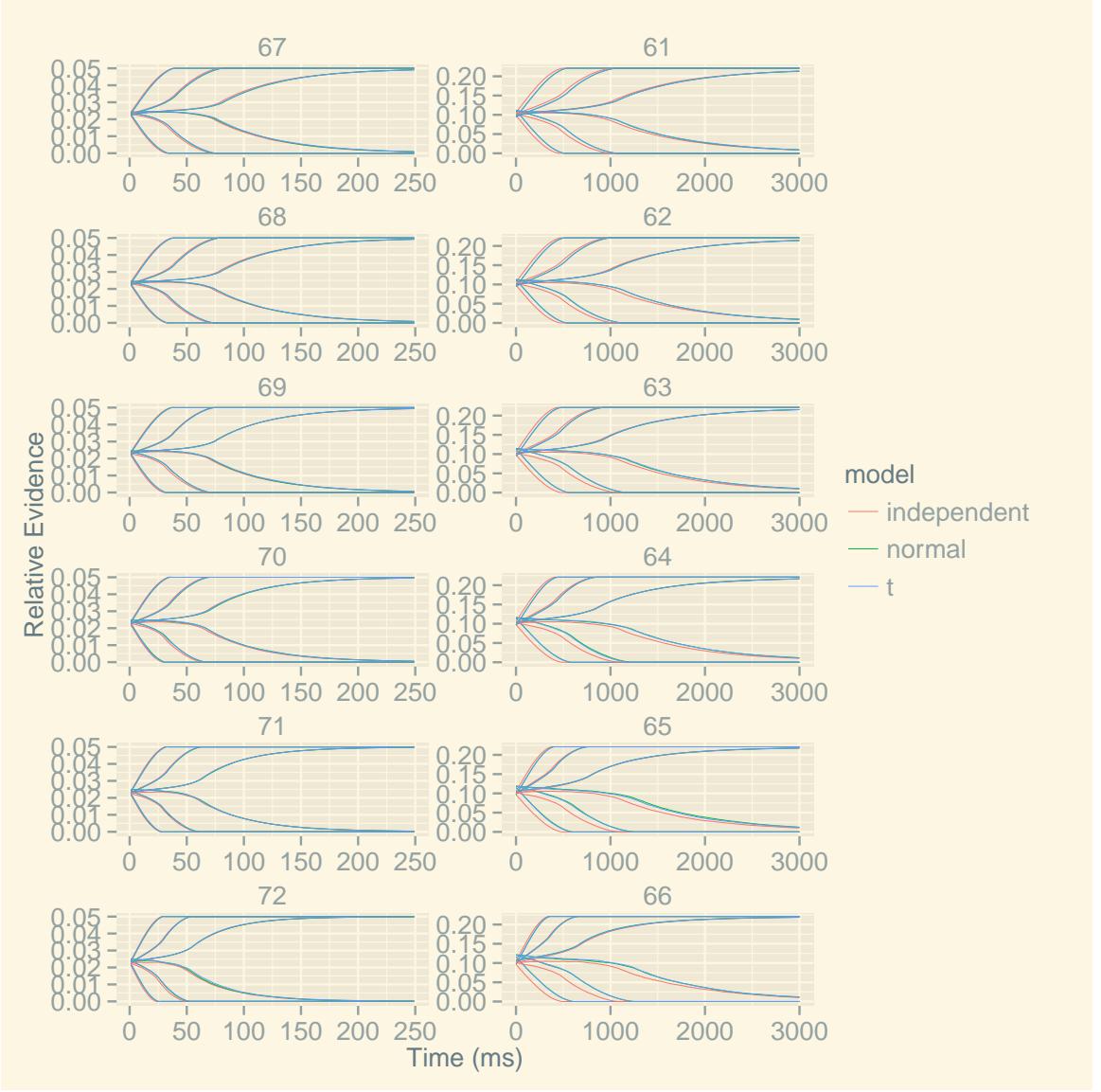


Figure 4.6: Predicted mean sample paths for the three models when $\rho_{\delta,\beta} = -0.85$. The left column are speed and the right column are accuracy conditions. The rows going from the top to the bottom represent increasing white-to-black pixel ratio. Within each subplot the mean sample paths are split into bright decision, mapped to the upper boundary, and dark decision, mapped to the lower boundary. Both bright and dark decisions are further split into fast, intermediate and slow decisions.

but slowest decisions are slightly less than the mean bias, or 0.463. Dark decisions show a similar, almost reversed pattern of departures from the mean bias. Fastest and intermediate dark decisions are less than the mean bias, being 0.442 and 0.460, respectively, while slowest dark decisions are not different from the mean bias. For both decisions, the fastest speed is obtained when responses are closest to their boundary, while intermediate and slowest responses arise from being farther from the boundary, sometimes crossing the mean bias line.

Increasing brightness to moderate levels, as shown in subplots 32 – 36, lowers the initial bias slightly to 0.480 for the fastest bright decisions, with higher brightness values producing no further effect. In contrast, increasing brightness reverses the initial bias of the intermediate bright decisions to less than the mean bias, and further lowers the initial bias of the slowest bright decisions. At the highest brightness, intermediate bright decisions begin with the initial bias of 0.461 and slowest bright decisions start at 0.456. Similarly, when brightness is increased, the fastest and intermediate dark decisions decrease further from the mean bias, and the slowest dark responses change from no difference to initial bias less than the mean bias. Under condition of maximum brightness, fastest dark decisions start from 0.425, intermediate dark decisions start from 0.450 and slowest dark decisions start from 0.457. Basically, under conditions of high brightness, any decision, besides the fastest bright decision, has initial positions biased away from the mean bias towards the dark decision. The qualitative pattern is stable, but quantitatively the maximum difference between any kind of decision and the mean bias does not exceed 8%.

Changing instructions to “accuracy” causes a similar pattern of initial biases, as shown in subplot 25. When stimuli are near-neutral, initial bias in the fastest bright

decisions is 0.480, higher than the mean bias, but biases in the intermediate and slowest bright decisions, 0.463 and 0.461, respectively, are at or lower than the mean bias. Conversely, the fastest dark decisions start below the mean bias at 0.444, the intermediate dark decisions show no difference from the mean bias and the slowest dark decisions start above the mean bias at 0.468.

Staying under the accuracy instruction, but increasing brightness enhances initial bias deviations for the bright decisions and partially reverses initial bias deviations for the dark responses, as portrayed in subplots 26 – 30. At the highest brightness level, fastest bright decisions show no change in the initial bias; intermediate and slowest bright decisions show larger decreases from the mean bias, 0.459 and 0.455. Alternatively, the fastest dark decisions increase in their departure from the mean bias to 0.437, while the intermediate and slowest dark decisions move below the mean bias to 0.456 and 0.462, respectively. As under speed condition, increasing brightness decreases initial bias for all the decisions, but the fastest bright decisions, towards the lower boundary. Quantitatively, initial bias never gets farther than 6% away from the mean bias.

Against the background of the independent copula, Figure 4.3 shows clear effects of elliptical copulas. However, there is negligible difference between the normal and t copulas. Green and blue lines nearly overlap. Thus, the following observations apply to both elliptical copulas.

Let's start with subplot 31. With “speed” emphasis and near-neutral stimuli, high positive correlation between decision bias and rate of evidence accumulation increases the distances of decisions of different speeds from the mean bias relative to the independent copula. Expressing the effect of copula in difference of independent

copula predictions from the normal copula, the fastest bright decisions start 0.015 bias units away, the intermediate bright decisions start 0.015 bias units away and the slowest bright decisions start at 0.011 bias units away, all closer to the upper boundary. Conversely, dark decisions of all speeds start closer to the lower boundary, with fastest decisions closest to it. Fastest dark decisions initiate -0.017 bias units away, intermediate dark decisions initiate -0.017 and slowest dark decisions initiate -0.019 bias units away.

Increasing brightness level causes an initial bias pattern of bright and dark decisions under elliptical copula similar to the independent copula, and with effects of correlation reduced under some conditions, as shown in subplots 32 – 36. At maximum brightness level, fast and intermediate bright decisions are still above the independent copula, with initial bias of 0.011 and 0.006 bias units away, respectively, but the slowest bright decisions drop below the independent copula predictions to -0.009 bias units away. On the other hand, dark decisions move even lower in their initial biases mostly causing larger differences between correlated and independent parameters. Fastest dark decisions start -0.015 bias units away, intermediate dark decisions start -0.027 bias units away and slow dark decisions start -0.041 .

Reducing correlation to medium or low values preserves the same pattern of effects, across brightness levels, but reduces their magnitude. This pattern is seen in Figures 4.1 - 4.2. For bright decisions under near-neutral stimuli, the highest difference between elliptical and independent copula is at the intermediate speed, where high correlation causes departures of 0.015 bias units away, medium correlation causes departures of 0.010 bias units away and low correlation causes departures of 0.004 bias units away. Other speeds show the same pattern, with only slightly different

magnitudes. Dark decisions, however, show the greatest departures at the slowest speed. With near-neutral stimuli, the highest correlation causes departures of -0.019 bias units away, the medium correlation causes departures of -0.011 bias units away, and the low correlation causes departures of -0.003 bias units away.

With increased brightness, the effect of reducing correlation is also to reduce the difference in initial bias between independent and elliptical copula sample paths. The effects are maximum for the fastest decisions. Bright decisions at the highest correlation cause differences of 0.011 bias units away, at the medium correlation the difference is 0.007 bias units away, and at the low correlation the difference is 0.003 bias units away. For dark decisions, however, the largest difference is for the slowest sample paths. When correlation is high the effect is -0.041 bias units, when correlation is medium the effect is -0.025 bias units and when correlation is low the effect is -0.003 bias units. At other speeds, both bright and dark decisions show the same trend of diminishing difference in initial bias predictions of the elliptical copula from the independent copula.

Switching to subplot 25 shows samples paths under “accuracy” instruction. Changing instructions increases effects of correlations. Near-neutral stimuli cause departures in initial bias from the mean bias in the same directions, but with larger magnitude with correlated than independent parameters. Fastest bright decisions initiate 0.045 bias units away, intermediate bright decisions initiate 0.033 bias units away, slowest bright decisions initiate 0.016 . Similarly, dark decisions under correlated parameters cause departures farther below the mean bias than when correlation is 0. Fastest dark decisions start -0.045 bias units away, intermediate dark decisions start -0.040 bias units away and slowest dark decisions start -0.031 bias units away.

Subplots 26 – 30 track increasing brightness level. The change in initial bias of sample paths is mostly similar to what independent copula causes, but magnitudes are greater and one class of decisions is qualitatively different. At the maximum brightness level, fastest and intermediate bright decisions initiate above the mean bias, with 0.032 and -0.012 bias units away, respectively, with the latter being on the opposite side from what the independent copula predicts. Slowest bright decisions, as for the independent copula, are below the mean bias, initiating -0.020 bias units away. For dark decisions, the pattern is the same for both independent and elliptical copulas, but the latter showing greater departures from the mean bias. Fastest dark decisions start -0.062 bias units away, intermediate dark decisions start -0.069 bias units away and slowest dark decisions start -0.078 bias units away.

Similar to the “speed” instruction, lowering correlation between decisions bias and rate of evidence accumulation reduces prediction differences. This is seen in the right column of subplots in Figures 4.1 - 4.2. Under near-neutral stimuli, the greatest effect for bright decisions is for fast sample paths. When correlation is high the initial bias effect is 0.045 bias units, when correlation is medium the initial bias effect is 0.027 bias units, and when correlation is low the initial bias effect is 0.009 bias units. Similarly, dark decisions show greatest effect in fastest sample paths. For high correlation the initial bias effect is -0.045 bias units, for medium correlation the initial bias effect is -0.028 bias units, and for low correlation the initial bias effect is -0.009 bias units.

At increased brightness levels, reducing correlation still reduces differences between independent and copula model’s predictions. Fastest bright decisions show the greatest effects of correlation. At the maximum brightness level, when correlation is high the initial bias is 0.032 bias units away, when correlation is medium the initial

bias is 0.020 bias units away, and when correlation is low the initial bias is 0.006 bias units away. For dark decisions, the greatest effects of correlation are mostly for slowest samples paths. With high correlation the initial bias effect is -0.078 bias units, with medium correlation the initial bias effect is -0.044 bias units, and with low correlation the initial bias effect is -0.012 bias units. For other speed, the pattern of declining effect of correlation is similar.

Next, lets consider Figure 4.6 , which shows how negative correlation between decision bias and rate of evidence accumulation impacts initial bias of decisions. Changing correlation from positive to negative produces a reversal in the trends where initial bias tends to be on the opposite side of the mean bias, away from the final decision. However, magnitudes of differences in predictions of elliptical and independent copula are close. Subplot 67 contains sample paths generated under “speed” instruction and near-neutral brightness; in it, the fastest, intermediate and slowest bright decisions start -0.019 , -0.014 , and -0.010 bias units away, respectively. Similarly, fastest, intermediate and slowest dark decisions start away from the boundary the sample paths eventually hit, with effects of 0.017 , 0.020 and 0.021 bias units away, respectively.

Increasing brightness levels, as shown in subplots 68 – 72, generally shifts all types of decisions downwards towards the dark decision. The reversed pattern is still visible at the highest brightness level, with somewhat adjusted magnitudes. For fastest bright decisions initial bias starts -0.012 bias units away, for intermediate bright decisions initial bias starts -0.005 bias units away, and for slowest bright decisions initial bias starts 0.01 bias units away. On the other hand, fastest, intermediate and slowest dark decisions start 0.023 , 0.028 , and 0.041 bias units away, respectively.

Lowering correlation to medium and low level, as shown in Figure 4.1 and 4.2 , preserves the reversed effects, but reduces their magnitude. Under near-neutral brightness, the largest effect for bright decisions is in the fastest sample paths. For high correlation effect is -0.019 bias units, for medium correlation effect is -0.012 bias units, and for low correlation effect is -0.004 bias units. In contrast, for dark decisions the largest effect is in the slowest sample paths. Under high correlation effect is 0.021 bias units, under medium correlation effect is 0.013 bias units, and under low correlation effect is 0.004 bias units.

Under increased brightness level, the effect of correlation on initial bias drops off similarly. The bright decisions still show the greatest correlation effect in the fastest sample paths. High correlation causes a difference of -0.012 bias units, medium correlation causes a difference of -0.006 bias units, and low correlation causes a difference of -0.001 bias units. Whereas dark decisions are still maximally impacted by correlation in the slowest sample paths. The effect is 0.041 bias units under high correlation, the effect is 0.023 bias units under medium correlation, and the effect is 0.006 bias units under low correlation.

Shifting to “accuracy” instructions, the reversed pattern can also be seen in subplots 61 – 66. For subplot 61, with stimuli brightness level set to near-neutral, bright decisions are biased towards the dark decision boundary. Fastest bright decisions initiate -0.056 bias units away, intermediate bright decisions initiate -0.033 bias units away, and slowest bright decisions initiate -0.056 bias units away. In contrast, dark decisions are biased towards the bright decision boundary. In comparison to the independent copula, the effect is 0.060 bias units for the fastest dark decisions, 0.044

bias units for the intermediate dark decisions, and 0.027 bias units for the slowest dark decisions.

Raising brightness level generally preserves the reversed pattern. The difference from the predictions of the independent copula somewhat diminish for the bright decisions and increase for the dark decisions. At the highest brightness level, fastest bright decisions start -0.039 bias units away, intermediate bright decisions start -0.009 bias units away, and slowest bright decisions start 0.027 bias units away. In contrast, mean sample path for fastest dark decisions is 0.091 bias units away, for intermediate bright decisions is 0.090 bias units away, and for slowest bright decisions is 0.089 bias units away.

The magnitude of the reversed pattern depends crucially on the correlation magnitude. Comparing Figures 4.4 - 4.6 shows how the effect reduces with smaller correlation level. For near-neutral stimuli, bright decisions show the greatest effect of correlation in the fastest sample paths. Predictions of independent and elliptical copula are different by -0.039 bias units under high correlation produces, by -0.031 bias units under medium correlation produces, and by -0.009 bias units under low correlation produces. Whereas dark decisions do not show a group of sample paths uniformly most affected by parameter correlation. As a concrete example, the slowest decisions show that at high correlation the effect is 0.060 bias units, at medium correlation the effect is 0.017 bias units, and at low correlation the effect is 0.005 bias units. Similar diminishing effect holds for sample paths at other speeds.

As shown under “speed” instruction, increasing brightness does not diminish the decline in initial bias difference between independent and elliptical model predictions. At the highest brightness level, bright decisions show largest effect of correlation in

the fastest sample paths. High correlation causes -0.039 bias units effect, medium correlation causes -0.022 bias units effect, and low correlation causes -0.006 bias units effect. In contrast, largest effect for dark decisions is in the slowest sample paths. The effect is 0.089 bias units under high correlation, the effect is 0.050 bias units under medium correlation, and the effect is 0.015 bias units under low correlation. Other speeds show a similar declining effect when correlation is lowered.

In addition to affecting initial bias, changing magnitude of correlation between initial evidence and rate of evidence accumulation may have an effect on the shapes of the mean sample paths. To examine this effect shapes generated under the independent copula again provide a baseline, which is the same across Figures 4.1 - 4.6

Let's consider predicted mean sample paths under the "speed" instruction first, the left column of subplots in Figure 4.1. In subplot 7, the brightness level is near-neutral. On average, sample paths corresponding to fastest bright decisions go straight to the upper boundary without apparent lingering around the starting point. The mean sample path moves mostly in a straight line, but bends closer to the boundary such that its rate of change decreases. Similarly, intermediate bright decisions, on average, move immediately from the starting point towards the upper boundary. However, the rate of change is at first small, then picks up near the middle of overall travel time, and then diminishes again closer to the boundary. Similarly, mean sample path for slowest bright decisions leaves the starting point straight for the boundary. The rate of change is near-flat at first, then increases noticeably, and finally declines gradually as the sample path comes closer to the boundary.

Dark decisions are mirror images of bright decisions for fast and intermediate speeds when it comes to shapes and patterns in rate of change. The slowest dark decisions are different only in the beginning when their mean sample path moves somewhat away from the lower boundary, but then changes its course towards the lower boundary. The rate of change away and then towards the boundary is small, but then the rate of change increases substantially and finally decreases again when the mean sample path gets closer to the boundary.

Increasing brightness level or switching to the “accuracy” instruction does not change the shapes of mean sample paths for either bright or dark decisions. At the highest brightness level, the qualitative features are constant, but quantitatively, rates of change in the mean sample paths increase to reflect the higher quality of stimulus. Likewise, switching instruction to accuracy emphasis produces similarly shaped mean sample paths, but the rates of change decrease.

Against the baseline provided by the independent copula model, predictions for normal and t copulas are portrayed with green and blue curves, respectively. The first observation is that predictions of the two copulas differ negligibly, as they did for the initial bias predictions. Thus the following conclusions about the shapes hold in general for both elliptical copulas.

Starting at the subplot 31 of Figure 4.3, under conditions of near-neutral stimuli and speed emphasis, the graph shows that predictions of the elliptical copulas are qualitatively the same as those under the independent copula. The qualitative identity holds at all speed for both bright and dark decisions. The only noticeable effect of correlated parameters is a slight quantitative change. The effect is visible in intermediate and slowest dark decisions in the initial segment of the sample path

when the former moves towards the lower boundary with low rate of change and the latter moves towards the upper boundary with the low rate of change.

Increasing brightness preserves the pattern of effects under the near-neutral condition, as seen in subplots 32 – 36. When brightness level is highest, qualitatively, the shapes of mean sample paths predicted by independent and elliptical copulas are identical. In contrast, the quantitative effect in intermediate and slowest dark decisions becomes more pronounced, but remains in the initial segment of the sample path.

Shifting attention to subplots 25 – 30 shows that changing emphasis from speed to accuracy preserves the qualitative and quantitative pattern of effects of correlated parameters. Predictions generated by independent and elliptical copulas show qualitative identity. Quantitatively, the rate of change in the intermediate and slowest mean sample path is different in the same way as under speed emphasis during the initial segment of the sample path. When parameters are correlated, intermediate dark decisions move towards the lower boundary at a smaller rate and slowest dark decisions move towards the upper boundary at a higher rate.

Under reduced correlation magnitude, the qualitative patterns remain and the quantitative effects reduce. Figure 4.2 shows predictions under intermediate correlation. Across the subplots, the shapes of the mean sample paths are the same for all speeds and both decisions, regardless of correlation level. However, the intermediate correlation is sufficient to produce a noticeable quantitative effect, as seen in intermediate and slowest dark decisions, especially with higher brightness levels. When correlation is small, predictions still show qualitative identity between independent and elliptical copula, as shown in Figure 4.1. However, the quantitative effect in

intermediate and slowest dark decisions is negligible, regardless of instruction and brightness level.

When correlation switches from positive to negative, the differences in predictions change both qualitatively and quantitatively. Lets concentrate on Figure 4.6 that summarizes mean sample paths under high negative correlation. Scanning subplots 67 – 72 shows that under speed emphasis the qualitative pattern of independent and elliptical mean sample paths are identical except for the slowest dark decisions. Independent copula predicts a slow rise towards the upper boundary in the initial segment of the slowest dark decisions, but elliptical copulas with negative correlation predict a slow decline towards the lower boundary. The same pattern holds across brightness levels. Quantitatively, negative correlation causes higher rates of change in intermediate and slowest sample paths corresponding to dark decisions. The effect on rate of change increases with brightness level.

Changing instruction to “accuracy” shows the same pattern qualitative pattern as under the “speed” instruction. However, quantitatively, the effect of negative correlation is higher under “accuracy” instruction. Intermediate and slowest sample paths under elliptical copula have higher rates of change in the initial segments, which increase with higher brightness level.

Similar to positive correlation, when the negative correlation decreases in magnitude, differences in predictions between independent and elliptical copulas are stable qualitatively, but change quantitatively. Looking at Figure 4.5, when correlation is medium the qualitative shift in shape of the slowest dark sample path is still noticeable. On average, slowest dark decisions decline towards the lower boundary from their initial point when parameters are correlated, but for the independent copula

slowest dark decisions first rise towards the upper boundary before declining towards the lower boundary. Quantitatively, changing correlation from high to medium negative correlation reduced prediction differences in intermediate and slowest dark decisions. Dark decisions at both speeds decline with lower rates.

Shifting to predictions under low correlation, as portrayed in Figure 4.4, shows that qualitative similarities and discrepancies between the elliptical and independent copulas remain, but quantitative difference become tiny. The slowest dark mean sample path declines from its initial position towards the lower boundary, with all other decision types showing the same qualitative pattern between the copulas. However, the difference in rate of change in both intermediate and slowest dark decisions declines relative to the independent copula decline to a tiny level.

To conclude the results for sample path predictions, I shift to describing absorption times. Absorption times were not initially planned, but their effects can be so large that this information may be important in understanding effects of correlations. Recall that absorption time is an instance when the mean sample path reaches one of the boundaries. The differences between the independent and elliptical copulas are a very visible feature of predictions, especially at higher correlations, as can be seen in Figures 4.1 - 4.6. As for the initial bias and shapes of sample paths, the absorption times for the normal and t copula models are similar, hence the following observations generalize across them.

Let's concentrate on Figure 4.3, where predictions are shown under high correlation. Predictions for the “speed” instruction are in the left column. Across brightness levels, fastest and intermediate decisions of either bright or dark type show negligible

difference between elliptical and the independent copula. Under near-neutral brightness, the effect ranges from -2 ms to 0 ms, and under highest brightness, there is no effect. However, slowest decisions can show large effects. When stimuli are near-neutral, slowest bright decisions show 45 ms effect while slowest dark decisions show 48 ms effect. Changing stimulus brightness to high causes slowest bright decisions to have -40 ms effect and slowest dark decisions to have 38 ms effect, a decline relative to the lower brightness.

When instruction is changed to accuracy emphasis, effects are much larger across decisions types. With near-neutral stimuli, fastest bright decisions show -39 ms difference, intermediate bright decisions show -35 ms difference, and slowest bright decisions show 813 ms difference. Similarly, fastest, intermediate and slowest dark decisions show -48 ms, -50 ms, and -1264 ms effect, respectively. Increasing brightness of stimuli tends to decrease effects in bright decisions and increase effects in dark decisions. At the highest brightness level, the effect in fastest bright decisions is -15 ms, in intermediate bright decisions is 0 ms, and slowest bright decisions is -62 ms. Whereas the effect in fastest dark decisions is -92 ms, intermediate dark decisions is -116 ms, and in slowest dark decisions is 1242 ms.

Medium level correlation magnitude still causes substantial differences under “accuracy”, but not under “speed” instruction. This pattern can be seen in Figures 4.2. Fixing instruction to “speed” emphasis gives negligible effects to fastest and intermediate decisions of both bright and dark type, ranging from -3 to 1 ms effects across brightness levels. At the slowest speed, bright decisions show effect of -3 ms and dark decisions show effect of 39 ms when brightness level is near-neutral, but at the

highest brightness level slowest bright decisions show effect of -69 ms and slowest dark decisions show effect of -83 ms.

When instructions are fixed to “accuracy” emphasis and stimuli have near-neutral brightness, the effect in fastest bright decisions is -22 ms, in intermediate bright decisions is -17 ms, and in slowest bright decisions is -99 ms. Similarly, the effect in fastest dark decisions is -26 ms, in intermediate dark decisions is -24 ms, and slowest dark decisions is -340 ms. Changing stimulus brightness to highest level causes reduction in effects for bright stimuli such that fastest decisions are -9 ms apart, intermediate decisions are 0 ms apart, and slowest decisions are -1672 ms apart. In contrast, dark decisions show an increased effect with fastest decisions being -65 ms away, intermediate decisions being -73 ms away, and slowest decisions being 1652 ms away.

Decreasing correlation to the low level causes almost completely removes effects for fastest and intermediate decisions of both types, but effects still persist under slowest decisions. This pattern is summarized in Figure 4.1. Under “speed” emphasis, fastest and intermediate decisions, either bright or dark, range in effects from -1 ms to 1 ms, regardless of brightness level. However, when brightness is near-neutral, slowest bright decisions show 0 ms effect and slowest dark decisions show 10 ms effect; when brightness is highest, slowest bright decisions show -136 ms effect and slowest dark decisions show -25 ms effect.

Changing instructions to “accuracy” emphasis also shows mostly small effects. When brightness is near-neutral, bright decisions show effects of -7 ms when fastest, 2 ms when intermediate, and 1559 when slowest. Likewise, dark decisions of fastest, intermediate and slowest speed show effects of -8 ms, 1 ms, and 905 ms, respectively.

Raising brightness level to highest causes -4 ms, 4 ms, and 382 ms effects for fastest, intermediate and slowest dark decisions, respectively. Highest brightness also causes effects of -23 ms, -17 ms, and 1050 ms for fastest, intermediate, and slowest dark decisions, respectively.

Turning to the negative correlation, the patterns shown in Figures 4.4 - 4.6 are qualitatively identical to the ones observed under positive correlation. Observing near-neutral stimuli under “speed” instructions causes little difference in absorption times of fastest and intermediate mean sample paths for both decisions, but large difference in slowest decisions. Bright decisions show effects of 1 ms, 3 ms, and -2 ms for fastest, intermediate, and slowest speeds, respectively. Dark decisions show effects of 2 ms, 2 ms, and 39 ms for fastest, intermediate, and slowest speeds, respectively. Increasing brightness leaves effects mostly unchanged such that stimuli with highest brightness cause bright decisions with 1 ms, 0 ms, and 77 ms effects for fastest, intermediate, and slowest types. Likewise, the effects are 2 ms, 2 ms, and -72 ms for fastest, intermediate, and slowest dark decisions.

Changing instruction to “accuracy” increases effects substantially for both decisions. Near-neutral brightness of stimuli causes effects of 47 ms, 44 ms, and 870 ms for fastest, intermediate, and slowest bright decisions, respectively. Similarly, fastest, intermediate, and slowest dark decisions show effect magnitudes of 61 ms, 73 ms, and -1180 ms, respectively. Increasing brightness reduces effects for bright decisions, but enhances effects for dark decisions. Fastest bright decisions are 16 ms away, intermediate bright decisions are -3 ms away, slowest bright decisions are -1000 ms away, but fastest dark decisions are 118 ms away, intermediate dark decisions are 170 ms away, and slowest dark decisions are 365 ms away.

When negative correlation is reduced to medium or low level the pattern of effects changes similarly to when correlation is positive. At medium level effects are still strong for both bright and dark decisions across different speed levels when instructions are fixed to “accuracy”, but are negligible except for the slowest decisions when instructions are set to “speed”. When correlation is further lowered to small level, effects for both bright and dark decisions become small for fastest and intermediate decisions, but are still large for slowest decisions, when instructions are set to “accuracy”. In contrast, effects are negligible under “speed” instructions for all types of decisions, except for slowest bright and slowest dark decisions.

Taking absorption results as a whole, we can see stability in fastest and intermediate decisions, but high variability in slowest decisions. The fastest and intermediate decisions occur around the center of the overall distribution while the slowest decisions are drawn from the long tail. Conditioning on different parts of the distribution leads to very different conditional standard deviations. The stability in absorption times for fastest and intermediate decisions indicates that sample size were adequate and reported differences real. However, for the slowest decisions the haphazard flipping in sign and large variation in magnitude suggest that the sample size of 1×10^5 was not adequate. Hence, absorption times for the slowest decisions should be viewed skeptically or disregarded as being to noisy.

4.2.3 Discussion

The objective of examining the mean sample paths was to explore how adding a correlation parameter may change dynamics of the accumulation process. The interest in the sample paths is motivated by the finding of neural populations in cortical and

subcortical areas with slowly building up activity predictive of later movement (Gold & Shadlen, 2007; Heitz & Schall, 2013; Smith & Ratcliff, 2004). The results obtained in Study A1 bear, in a tentative way, on whether models with dependent parameters could improve fit of the independent copula model, suggest an alternative explanation for some features of single-cell recordings and/or suggest a novel experiment for testing the overall evidence accumulation model of perceptual decisions.

To get a descriptive handle on the complex effects of correlations on the sample paths, the results concentrated on primary features of the mean sample paths: initial mean bias, shapes across several segments, and absorption times. The effects of the correlation are most visible in starting points and absorption times, starting with moderate correlation magnitude. The latter are driven by the combination of medium effect on initial bias and small effect on rate of change that build up over the duration of decision to give large differences in absorption times. The largest effect in absorption time suggests that correlations may also be important for the reaction times and that their flips in difference from the independence diffusion model may impact asymmetry of response time distributions.

Finding medium and large effects on initial decision bias and absorption times suggest that the correlation parameter may be important in fitting and explaining these two features. A quick test of usefulness of adding the correlation parameter can be based on the condition where independent and normal copula models showed greatest discrepancy. Greatest difference in predictions appeared in dark (incorrect) decisions when brightness levels were highest and instruction set to “accuracy”. The effects could reach around 10% differences in decision bias and around 170 ms in absorption times.

Interestingly, the results imply that the diffusion model predicts some positive correlation between drift rates and starting points, after sorting the sample paths by decision and speed. This is visible in all the figures where mean sample paths corresponding to the bright decisions tend to have starting points closer to the upper boundary and positive drift rates driving the mean sample paths upwards. Combining the prediction of the independent copula model with typically successful fits of the standard Ratcliff model suggests that data collected under standard experimental procedure, akin to the Ratcliff and Rouder (1998) experiment, is consistent with some positive correlation between bias and rate of evidence accumulation. If the model is expanded with correlation parameters, as explored in this thesis, we should estimate a positive correlation between the starting point and the drift rate. The mean sample paths, however, say nothing about the other two correlation parameters.

The independent copula model naturally predicts that starting points, especially in fastest and intermediate decisions, will be closer to the boundary at which the process will absorb. Adding a positive correlation enhances this effect in the fastest and intermediate decision, and causes it in the slowest decisions. Flipping the sign of correlation reverses the effect such that starting points are away from the absorbing boundary and produce criss-crossed paths. The pattern of effects suggests that the two models may be similar under the standard experimental settings, but will diverge under the negative correlation.

A qualitative divergence of the two models under negative correlation suggests a new experiment. The experiment would be based on a manipulation that correlates stimulus quality with a decision bias in a negative relation. Say, participants are signaled to expect a dark stimulus while receiving a bright stimulus. The target of

analysis would be the initial segments of the spike rate functions collected from the buildup neurons that should look reversed relative to the control condition where the manipulation is absent or induces a positive relation.

Besides differences between the independent and elliptical copulas, the negligible differences between the normal and t copulas are also important. In practice, which one should be used? Normal copula takes less function calls to evaluate and has one less parameter than the t copula. At least on the basis of the mean sample paths, it is more computationally efficient to use normal copula in theoretical explorations and data fitting. A similar conclusion will be shown to hold for the joint distribution of behavioral data in Study A2.

Lastly, the study highlights the limited nature of the Ratcliff diffusion model. It is a one stage model with other processes collapsed into a single non-decision time constant (Ratcliff & McKoon, 2008; Wagenmakers, 2009). In actual applications of the model to single cell recording from monkeys, ad-hoc equations based on exponential decay were added and the fit to neural data obtained by eyeballing (Ratcliff et al., 2011, 2007). The rising interest in model-based neuroscience suggests that lack of a decay process in the decision process, and lack of details of the anterior and posterior processes is an important modeling problem for the near future.

4.3 Study A2 - Response times and Responses

When a sample path hits one of the boundaries, the boundary and the absorption time combined with non-decision time are equivalent to responses and response time data. Study A2 examines what the three models predict for the joint distribution of the behavioral data. To better understand effects of experimental variables outlined

in Section 4.1, the joint distribution has to be reduced to several summary quantities. Similar to study A1, obtaining summary quantities involves integration problems. The next subsection covers the summary quantities and methods for their calculations, followed by results and discussion.

4.3.1 Joint Distribution Calculation

While studying mean sample paths under correlated parameters should suggest how the decision process dynamics change for different patterns of correlation, examining the joint density of response times and responses should show how the resulting behavior changes, especially the response asymmetries. The joint density of response times and responses has a long history of experimental investigation (Luce, 1986; Stone, 1960; Townsend & Ashby, 1983), and interpretation of its features may benefit from considering correlations among psychological parameters. Recall that the joint density is a mixture model averaged with respect to variability in parameters, such that for model $i = 1, 2, 3$

$$f_i(t^{rt}, r \mid \alpha, \boldsymbol{\theta}_i) = \int_A f(t^{rt}, r \mid \alpha, \delta, \beta, \tau^{nd}) f_i(\delta, \beta, \tau^{nd} \mid \boldsymbol{\theta}_i) dA, \quad (4.5)$$

where $\boldsymbol{\theta}_i$ will reflect the correlation structure and other parameters defined in Section 3.3.

For a fixed correlation structure, the joint density of behavior contains all the predictions of interest. We could look at its shape, which consists of two density components, one for each response. Or we could examine summary features like the vector of mean of response times and responses, and their covariance matrix. However, a more popular approach is to examine response time density function conditioned on choice (Luce, 1986; Vickers, 1979), both of which follow from the joint density of

t^{rt} and r . I adopted the standard approach because it emphasizes the often observed performance asymmetry between different responses and provides a systematic way of examining effects of a correlation structure.

We can obtain conditional densities by normalizing the joint density with the probability of response. For response r and model i , conditional density of response time is

$$f_i(t^{rt} | r, \alpha, \boldsymbol{\theta}_i) = f_i(t^{rt}, r | \alpha, \boldsymbol{\theta}_i) / P\{r = c | \alpha, \boldsymbol{\theta}_i\}, \quad (4.6)$$

where $c \in \{0, 1\}$ is a choice. As was done with mean sample paths, to enable analysis we need to reduce a function, an infinite-dimensional object, to a vector or a matrix, finite dimensional objects. To provide an informative account of the effects of correlated parameters, it is necessary to capture location, scale, shape and the corresponding response probabilities (normalizing constants) of each conditional density of response time. The quintile-probability (QP) plot provides a concise graphical way of summarizing all these data features (Ratcliff & Rouder, 1998). The QP plot combines the conditional reaction time quantiles and their corresponding choice probabilities to show how their relation changes across stimulus quality conditions.

The points of the QP plot are pairs of choice probabilities and reaction time quantiles. The probability of hitting the upper boundary for the models with variable parameters is

$$\begin{aligned} P_i\{r = 1 | \alpha, \boldsymbol{\theta}_i\} = \\ \int_{\delta \times \beta} \left[\exp\left(\frac{-2\delta\alpha\beta}{\sigma^2}\right) - 1 \right] / \left[\exp\left(\frac{-2\delta\alpha}{\sigma^2}\right) - 1 \right] f_i(\delta, \beta | \boldsymbol{\theta}_i) d\delta d\beta, \end{aligned} \quad (4.7)$$

where the probability of hitting the lower boundary is $P_i\{r = 0 | \alpha, \boldsymbol{\theta}_i\} = 1 - P_i\{r = 1 | \alpha, \boldsymbol{\theta}_i\}$. And the quantiles for the conditional density $f_i(t^{rt} | r, \alpha, \boldsymbol{\theta}_i)$ are defined

as

$$Q_i(p \mid \alpha, \boldsymbol{\theta}_i) = \int_{\delta \times \beta \times \tau^{nd}} \min\{t^{rt} : F_i(t^{rt} \mid \alpha, \delta, \beta, \tau^{nd}) \geq p\} f_i(\delta, \beta, \tau^{nd} \mid \boldsymbol{\theta}_i) d\delta d\beta d\tau^{nd}, \quad (4.8)$$

where Q_i is the quantile function and F_i is the cumulative distribution function for the i^{th} model.

Notice that computing the conditional densities and calculating all the summary features again requires integration. Similar to the problem of finding mean sample paths, the integrals are intractable and we can use the Monte Carlo method for both reaction time quantiles and choice probabilities. Both quantities can be estimated from the samples generated by the random walk algorithm for simulating approximate Wiener process by recording a boundary at which the process is absorbed and adding the non-decision time τ^{nd} to the absorption time (Tuerlinckx et al., 2001). Based on the generated sample, I used the distribution-free, median-unbiased quantile estimator (Hyndman & Fan, 1996) implemented in R (optional parameter type = 8) and the Monte Carlo estimator for the probabilities.

4.3.2 Results

To contrast the models' predictions with reasonable precision I simulated $n = 1 \times 10^7$ observations under each condition. The sample size was motivated by bootstrap estimates of standard errors of reaction time quantile and choice probability estimators. Estimates of choice probabilities use the same number of observations across conditions, and with the given sample size, standard errors in response probability is around 0.0005. So differences in predicted probabilities exceeding 0.001 are reliable.

In contrast, estimates of quantiles depend on the experimental condition, with the highest variation expected in the 0.90 quantile under highest brightness proportion for “dark” responses. Under highest brightness proportion, the expected proportion of “dark” response is around 10%, leading to an approximate sample size of 1×10^6 . With the expected sample size, bootstrap estimates indicate that 0.1 quantile will have approximately $\hat{\sigma}_{0.1} = 0.3$ ms, 0.3 quantile will have $\hat{\sigma}_{0.3} = 0.5$ ms, 0.5 quantile will have $\hat{\sigma}_{0.5} = 0.6$ ms, 0.7 quantile will have $\hat{\sigma}_{0.7} = 1.0$ ms and 0.9 quantile will have $\hat{\sigma}_{0.9} = 2.2$ ms. Hence, comparisons in 0.1, .3, .5, .7, .9 quantile predictions between models are reliable when differences exceed 0.6 ms, 1 ms, 1.2 ms, 2 ms, 4.4 ms, under all experimental conditions.

Simulations of behavioral data under different correlation patterns among parameters are summarized in Figures 4.7 - 4.9 with quantile-probability plots. Different rows stand for varying levels of correlation - with correlation increasing in magnitude from top to bottom. Each row of subplots describes performance during a reduced version of the Ratcliff and Rouder (1998) brightness discrimination task predicted by the three models. The left and right subplots divide predictions by “accuracy” and “speed” instruction, respectively. Within each subplot, the abscissa represents response probabilities, with values above 0.5 corresponding to bright response and below 0.5 representing dark response. The ordinate represents reaction time quantiles conditional on the response. Moving along the lines within each subplot shows how performance varies for different brightness levels with instructions fixed.

Of primary interest are effects of correlation structure and magnitude on reaction times and choice probabilities, accounting for quantile, brightness level, instruction, choice and type of elliptical copula. The red lines in any row of the Figures 4.7 - 4.9

represent behavioral predictions under the independent copula model, which can serve as a baseline for understanding correlation effects. Several trends are predicted by the independent model. Within each row subplot, the gaps of varying length between the quantiles indicate a positive skew in the reaction time distributions. The effect of brightness level is mostly to rescale the distribution by stretching the right tail towards ∞ and rapidly reduce bright response probability with brightness closer to 0.5. Reaction times and responses show correlations. Under “accuracy” instruction, the bright responses are slower under lower response probabilities and faster under higher response probabilities. Under “speed” instruction, the bright responses are slower across response probabilities. Across subplots in each row, when instructions are switched from “speed” to “accuracy”, the conditional reaction time distributions show shift and scale effects in hundreds of milliseconds (ms). The probability of bright response increases slightly. In summary, the quantiles and probabilities follow regularly observed patterns and form a baseline to gauge effects of parameter correlations.

As can be seen across Figures 4.7 - 4.9, one major trend is that differences between independent copula and elliptical copula models are substantial, but differences between normal and t copula models are negligible. The green and blue lines have almost perfect overlaps. The overlaps arise because predictions based on the t copula are mostly 2 ms to 4 ms smaller than the normal copula predictions. Such discrepancies are of no practical consequence with standard sample sizes, so the following conclusions are generalized across elliptical copulas.

Recall that predictions of behavior were obtained under three patterns of correlation of $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$: $[+ - -]$, $[+ + +]$, $[- - +]$. I first describe effects of

correlations that follow $[+ - -]$ pattern, summarized in Figure 4.7. For the sake of clarity, let's examine the bottom row of the figure, where predictions of elliptical copulas are based on highest correlation of 0.85. The reaction time distributions remain positively skewed. Effect of instructions and brightness proportions on reaction time shapes and response probability do not change from the independent copula. The reaction time distributions for the two responses also show asymmetries. Hence, qualitative features of the behavioral data and how it responds to experimental manipulations is similar, with or without correlations. This suggests that correlations act as an independent source of variation.

Parameter correlations, however, do introduce departures from the independent copula in the quantitative details of the behavioral data. One major effect of correlation is to shift and rescale the reaction time distributions for each response relative to the independent copula. Under “speed” instruction in the subplot 6, the bright responses speed up, with effects starting at -15 ms for 0.1 quantile and reducing to -11 ms for 0.9 quantile, when brightness is near neutral. As brightness level increases, the speed up effect reduces, giving effects starting at -9 ms for 0.1 quantile and reversing with 3 ms for 0.9 quantile, at the highest brightness level. In contrast, the dark responses slow down. When brightness is near neutral, difference from the independent copula starts at 19 ms for 0.1 quantile and decreases to 8 ms for 0.9 quantile. The higher levels of brightness increase the slow down effect, changing 0.1 quantile difference to 27 ms and 0.9 quantile to 26 ms, at the highest brightness.

Switching instructions to “accuracy” emphasis, portrayed in subplot 5, changes prediction differences between elliptical and independent copulas. For least bright stimuli, dark responses slow down only at 0.1 quantile by 12 ms, but speed up across

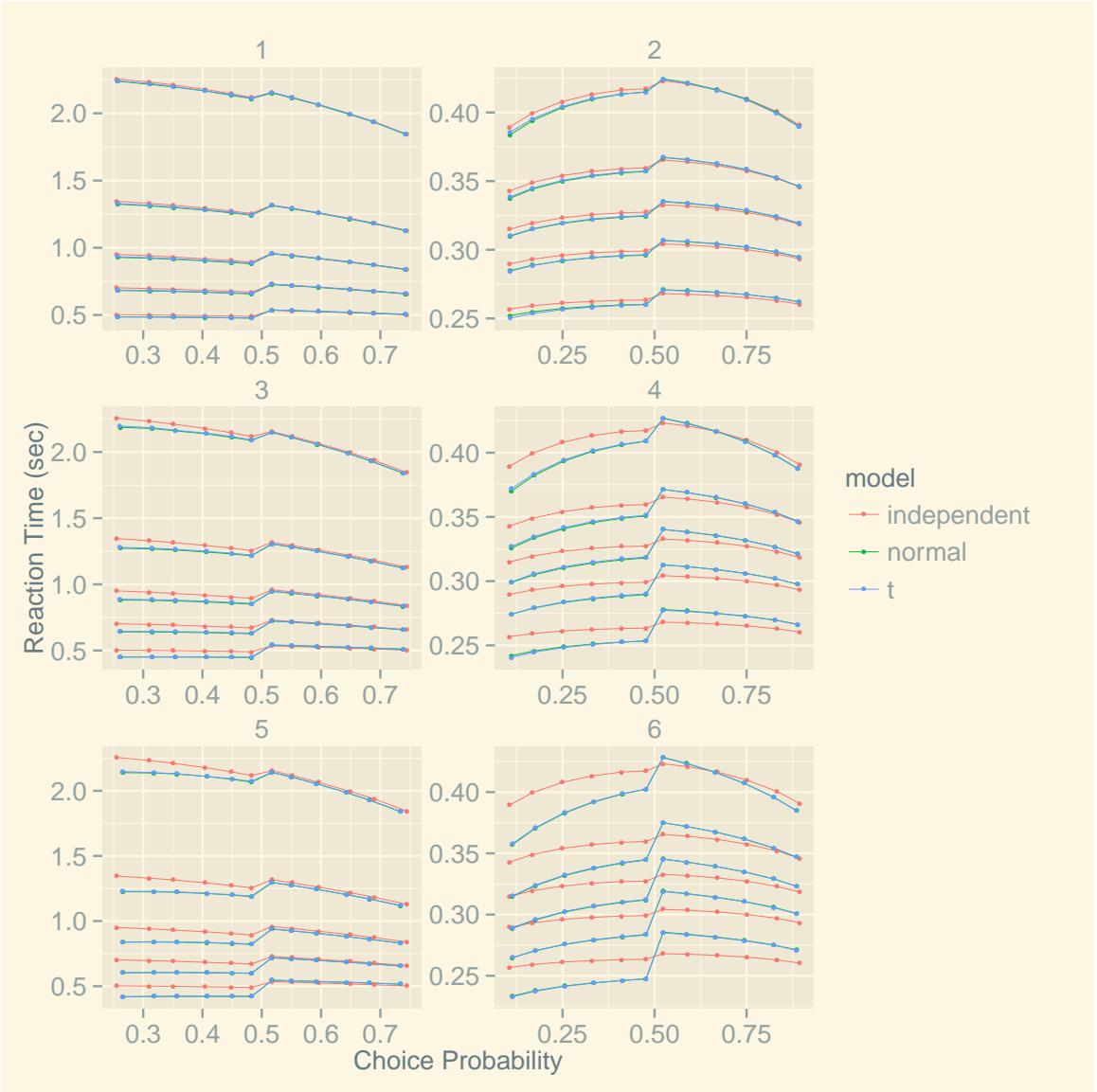


Figure 4.7: Behavioral predictions for the brightness discrimination task under $[+--]$ pattern of correlations among $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Correlations increase magnitude from top to bottom. Left column and right column correspond to “accuracy” and “speed”, respectively. Within each subplot, the lines connect reaction time quantiles conditional on response, and show how behavior changes from high to intermediate luminance.

the other quantiles, starting with -13 ms difference at 0.3 quantile, peaking with -34 ms at 0.7 quantile and dropping off with -26 at 0.9 quantile. As brightness increases, the first quantile continues to show a slow down effect, maxing out at 13 ms with the brightest stimulus. At other quantiles correlations increase the speed up effect, and with increasing brightness the peak speed up of -104 ms moves from 0.7 to 0.9 quantile. Similarly, bright responses mostly speed up. At near-neutral brightness level, the difference at 0.1 quantile is -68 ms, which reduces to -23 ms for quantile 0.9. As brightness increases, the speed up effect reduces across quantiles, with highest brightness leading to a speed up of -47 ms at 0.1 quantile, and slow down of 41 ms at 0.9 quantile.

The pattern of quantitative effects is complex, but there is an overall effect of correlations that ties them together. Correlated parameters control correlation between response and response time. The predicted differences between elliptical and independent copula under “speed” condition together amount to different asymmetry patterns, as can be seen in the subplot 6. Overall, independent copula leads to slightly slower bright response. The asymmetries for the independent copula range from 5 ms across quantiles at the lowest brightness level to a $4 - 1$ ms range across 0.1 – 0.9 quantiles at the highest brightness level. In contrast, elliptical copula reverses and magnifies the asymmetries under the independent copula. The bright responses are faster. For lowest brightness level, the 0.1 quantile has an asymmetry of -29 ms, which decreases to -14 ms for the 0.9 quantile. Increasing brightness level magnifies asymmetries to -33 ms at the 0.1 quantile and -21 ms at the 0.9 quantile.

Under “accuracy” instruction in the subplot 5, the independent copula shows a more complex asymmetry pattern. Under the lowest brightness level, the bright

responses are mostly slower than dark responses. At the 0.1 quantile the difference is 36 ms, but it reduces to 13 ms at the 0.7 quantile and flips to -39 ms at the 0.9 quantile. As brightness level increases bright responses switch to being faster than dark responses. The highest brightness level causes asymmetries of -32 ms at the 0.1 quantile and -785 ms at the 0.9 quantile. When parameters are strongly correlated the relative slow down of bright responses disappears. At the 0.1 quantile the asymmetry is -44 ms and at the 0.9 quantile the difference is -36 ms. With brighter stimuli the same asymmetry remains, but increases in magnitude. Highest brightness level causes a difference of -92 ms at the 0.1 quantile and -640 ms at the 0.9 quantile.

The overall effect across instructions and brightness levels of correlated parameters with $[+--]$ pattern is to speed up bright responses relative to the dark responses. Depending on the experimental conditions, the asymmetry either reverses the prediction of independent copula under speed instruction from slow to fast bright responses, or reduces asymmetry to a homogeneous pattern under “accuracy” of only faster bright responses.

In addition to reaction time quantiles, the probability of “bright” response is equal or smaller under all conditions, but the effects are minor. Under “accuracy” instruction in the subplot 5, the difference in response probability between elliptical and independent copula decreases gradually from 0.000 to -0.010 , as brightness increases from neutral to very bright, while under “speed” instruction, located in the subplot 6, the difference decreases from 0.000 to -0.080 . In combination, effect of correlated parameters is to slightly reduce accuracy in brightness discrimination.

Moving up to the middle row of Figure 4.7, the correlation changes to 0.5. The same differences in behavioral predictions between elliptical and independent copula are evident, but of lesser magnitude. Instructed to emphasize “speed”, the near-neutral brightness stimulus will cause a speed up of bright responses, as shown in the subplot 4. At the 0.1 quantile the speed up is -9 ms and the effect decreases to -6 ms at the 0.9 quantile. With higher brightness level the speed up diminishes such that at the 0.1 quantile the difference is -6 ms and at the 0.9 quantile the effect may reverse to 2 ms. In contrast, dark responses under elliptical copula slow down relative to the independent copula. At the lowest brightness level the 0.1 quantile is 10 ms slower and the 0.9 quantile is 6 ms slower. Increasing brightness level increases the slow down effect to 16 ms across all the quantiles.

Changing instructions to “accuracy” emphasis, reaction times for both responses mostly speed up relative to the independent copula, as seen in the subplot 3. Looking at the “bright” responses, when brightness is near-neutral, reaction times speed up maximally by -42 ms for the 0.1 quantile and the effect declines to -15 ms for the 0.9 quantile. Increasing brightness diminishes the speed up such that at the highest brightness level the reaction time difference is -29 ms for the 0.1 quantile and reverses to 26 ms for the 0.9 quantile. Similarly, the dark responses show overall speed up for the elliptical copula. The lowest brightness level causes a slight slow down of 6 ms for the 0.1 quantile, but causes a speed up for all the other quantiles, with -17 ms for the 0.9 quantile. With higher brightness level the 0.1 quantile slow down remains at 6 ms, but the other quantiles show a growing speed up, with -65 ms effect for the 0.9 quantile at the highest brightness level.

The asymmetry arising from the shifting and rescaling of the reaction time distributions is also detectable under 0.5 correlations. With “speed” emphasis instruction, the independent copula predicts slightly slower bright responses, shown in subplot 4. However, the elliptical copula causes the asymmetry to reverse, making bright responses faster. At near-neutral brightness the speed up effect is -15 ms for the 0.1 quantile, which slowly decreases to -7 ms for the 0.9 quantile. Increasing brightness level enhances the asymmetry. With brightest stimuli the predicted speed up effect is -18 ms for the 0.1 quantile and -12 ms for the 0.9 quantile.

Switching to the subplot 3, we can see the independent copula predicts slower bright responses for low brightness levels and then faster bright responses for higher brightness levels. When parameters are correlated at 0.5 the switch in asymmetry is almost replaced with only faster bright responses. At the lowest brightness level, the bright responses become faster at the 0.1 quantile with -12 ms effect, remain slower at quantiles $0.3 - 0.7$ with 9 ms, 16 ms, 5 ms, and again become faster at the 0.9 quantile with -38 ms effect. For all the other brightness levels the bright responses are faster for all quantiles. Brightest stimuli cause a speed up of -67 ms for the 0.1 quantile and -694 ms for the 0.9 quantile.

Unlike reaction times, the effect of correlated parameters on response probabilities is negligible. Under “speed” emphasis, elliptical copula predicts a decline in bright response ranging from -0.001 at the near-neutral brightness to -0.008 at the highest brightness. When given “accuracy” instructions, elliptical copula causes declines in bright response probability of 0.000 at the near-neutral brightness, which decrease to -0.005 at the brightest level. Overall, correlated parameters with $[+ - -]$ pattern cause a tiny decline in accuracy.

The final, upper, row of Figure 4.7 shows effects of 0.15 magnitude correlations on behavioral predictions. Let's concentrate on subplot 2, which shows behavior under "speed" instruction. When stimulus has near-neutral brightness, bright responses speed up by -3 ms at the 0.1 quantile and -2 ms at the 0.9 quantile. Increasing brightness level decreases the speed up effect on such that brightest stimuli cause -2 ms speed up at the 0.1 quantile and appear to reverse the effect at the 0.9 quantile to 1 ms (note: it is below the reliable difference with the given sample size). Oppositely, the dark responses slow down with correlation. At the near-neutral brightness, the slow down effect is 3 ms at the 0.1 quantile and 2 ms at the 0.9 quantile. Increasing brightness level increases the slow down, resulting in 4 ms at the 0.1 quantile and 5 ms at the 0.9 quantile.

Moving to subplot 1, with instruction changed to "accuracy" emphasis, the elliptical copula causes overall speed up relative to the independent copula. For bright responses, when stimulus is near-neutral, produce speed of -13 ms at the 0.1 quantile and -4 at the 0.9 quantile. With higher brightness level the speed up effect diminishes. When brightness reaches the highest level the speed up effect is -9 ms at the 0.1 quantile and reverses to 9 ms at the 0.9 quantile. Likewise, correlated parameters cause the dark responses to mostly speed up. At the near-neutral brightness there is a slight slow down of 2 ms at the 0.1 quantile, which gradually turns into speed up of -6 ms at the 0.9 quantile. Increasing brightness enhances the overall speed up of dark responses. With highest brightness level, there is still a slight slow down of 1 ms at the 0.1 quantile, and a larger speed up of -19 ms at the 0.9 quantile.

With small shifting and scaling effects of parameter correlations, the asymmetry caused by the elliptical copula is slightly different or similar to the independent copula.

The asymmetry is slightly different when “speed” is emphasized, as in the subplot 2. There is a weak reversal of the bright responses from slower to faster relative to the dark responses. Near-neutral brightness stimulus causes speed up of -1 ms at the 0.1 quantile and reduces the 0.9 quantile to slow down of 1 ms. Increasing brightness level enhances the speed up effect leading to speed up of -3 ms at the 0.1 quantile and -3 ms at the 0.9 quantile when brightness is highest.

In contrast, subplot 1 shows that under “accuracy” instruction small parameter correlation produces similar asymmetry predicted by the independent copula. For stimuli of near-neutral brightness the bright responses are overall slower. At the 0.1 quantile the slow down is 21 ms, which persists through other quantiles but reverses at the 0.9 quantile to -38 ms. Highest brightness levels overall cause faster bright responses, with highest level resulting in speed of -42 ms at the 0.1 quantile and -757 at the 0.9 quantile.

Lastly, correlation magnitude of 0.15 produces negligible effect on bright response probability. Overall, correlated parameters reduce probability relative to the independent copula. Difference in predictions under “accuracy” emphasis, shown in the subplot 1, starts at 0.000 for the near-neutral stimuli and drops to -0.002 for the brightest stimuli. Shifting to the subplot 2, when instructions emphasize “speed”, prediction difference is 0.000 for the lowest brightness and -0.001 for the highest brightness.

Figures 4.8 - 4.9 show similar effects of correlation magnitude, so I skip the detailed descriptions. For the remaining figures, I concentrate on the results that bear on the question of how the correlation pattern affects behavioral predictions. Correlation pattern [++], shown in Figure 4.8, leads to similar manipulation of response time

and response relation as $[+--]$, but there are noticeable differences. Let's concentrate on the bottom row where maximal correlations make the trends most clear. In the subplot 6, under “speed” instruction, elliptical copula causes faster bright responses, but the asymmetry is weaker than for $[+ --]$ pattern. The asymmetry is mainly weakened by considerably smaller speed up of correlated bright responses, but the effect on times of dark responses is also somewhat moderated by switching patterns.

Changing to subplot 5, with data generated under “accuracy” instruction, underscores the same conclusion that asymmetries are qualitatively the same, but there are quantitative differences. Similar to $[+ --]$ correlated parameters replace the more complicated asymmetry pattern that includes a switch from slower to faster under low brightness levels with overall faster bright responses. However, change in asymmetry under $[+++]$ pattern is driven by overall slow down of both dark and bright responses whereas the previous pattern showed overall speed up.

The effect of correlated parameters with $[+++]$ pattern, however, is opposite, but still negligible. The probability of bright responses increases with similar magnitudes as under $[+ --]$ pattern. In other words, switching the pattern slightly increases accuracy relative to the independent copula.

Relative to the other two patterns, setting correlations to $[--+]$ pattern produces the opposite asymmetry, as shown in Figure 4.9. Again, let's concentrate on the bottom row because effects are most pronounced at the highest correlation. In the subplot 6, with participant's emphasis on “speed”, slower bright responses under the elliptical copula are strongly enhanced relative to the independent copula. The effect arises from simultaneous slow down of bright responses and speed up of dark

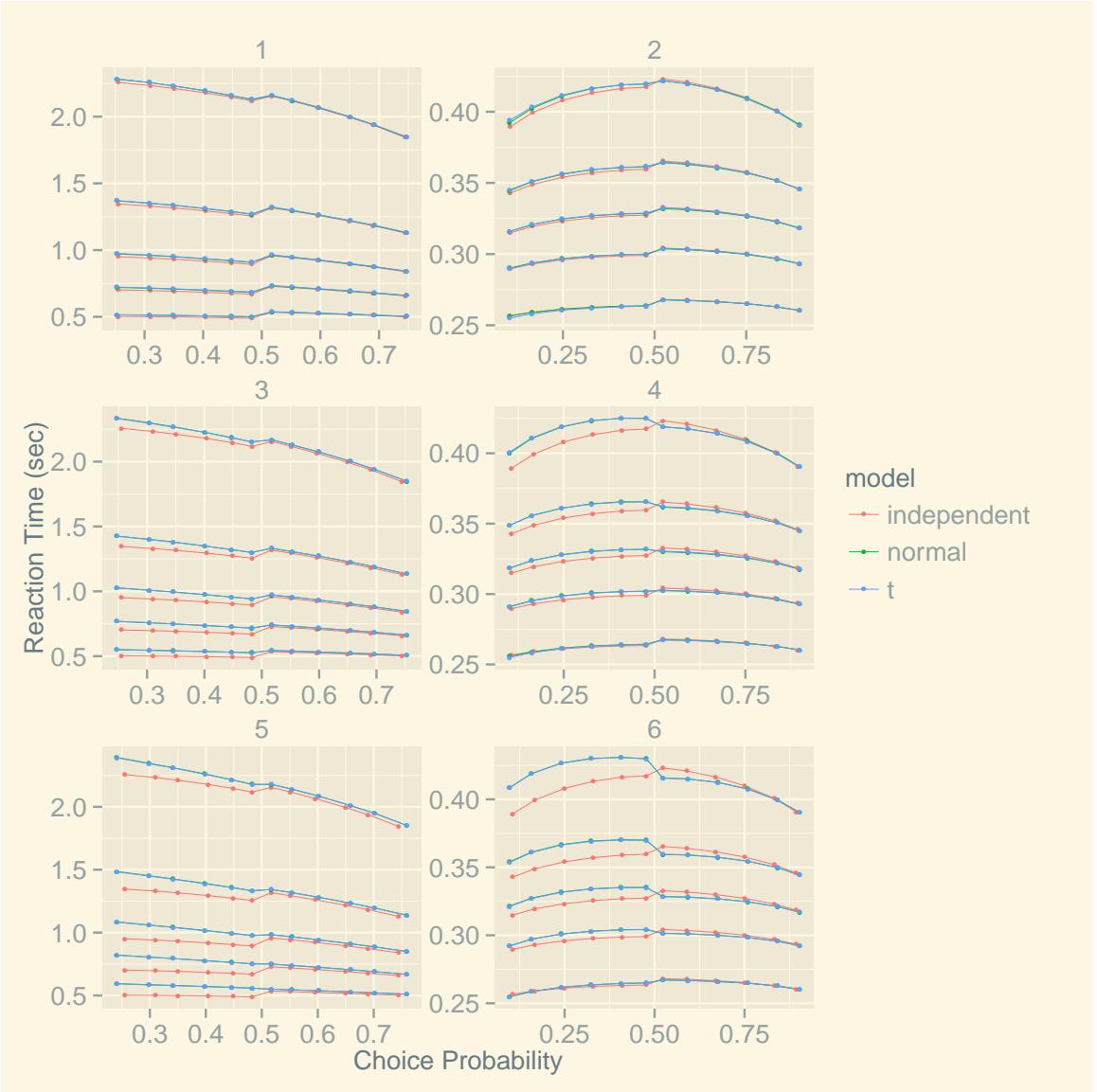


Figure 4.8: Behavioral predictions for the brightness discrimination task under [+++] pattern of correlations among $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Correlations increase magnitude from top to bottom. Left column and right column correspond to “accuracy” and “speed”, respectively. Within each subplot, the lines connect reaction time quantiles conditional on response, and show how behavior changes from high to intermediate luminance.

responses, which is the opposite set of effects relative to the other two correlation patterns.

Shifting instructions to “accuracy”, as shown in the subplot 5, also shows opposite effects of $[--+]$ pattern relative to the other two. As opposed to replacing the shift from slower to faster bright responses with uniformly faster trend, $[--+]$ pattern preserves the slow down effect at the near-neutral brightness level and extends it to the middle brightness levels.

Overall, correlated parameters can have substantial effects on reaction time at moderate to high magnitude, but negligible effects on choice probability even at high magnitude . The difference between the normal and t copulas are negligible, but the pattern of correlations and their magnitude have large, even opposite effects. The effects, however, interact with experimental conditions, response and quantile. Effects of correlations depend strongly on instructions and choice, moderately on the quantile and weakly on brightness level.

4.3.3 Discussion

Correlations between parameters can matter for performance of a simple perceptual discrimination task. Though unclear a priori, Figures 4.7, 4.8, 4.9 show clear evidence of potentially strong impact of correlated parameters. The primary effect of replacing independent copula with either elliptical copula is to adjust the correlation of response and response time. The correlation is evident from the speed asymmetries between dark and bright responses, sometimes one being faster, sometimes the other. Hence, adding correlation parameters can provide a useful modification of the standard Wiener process model of simple decisions to better fit or explain typically

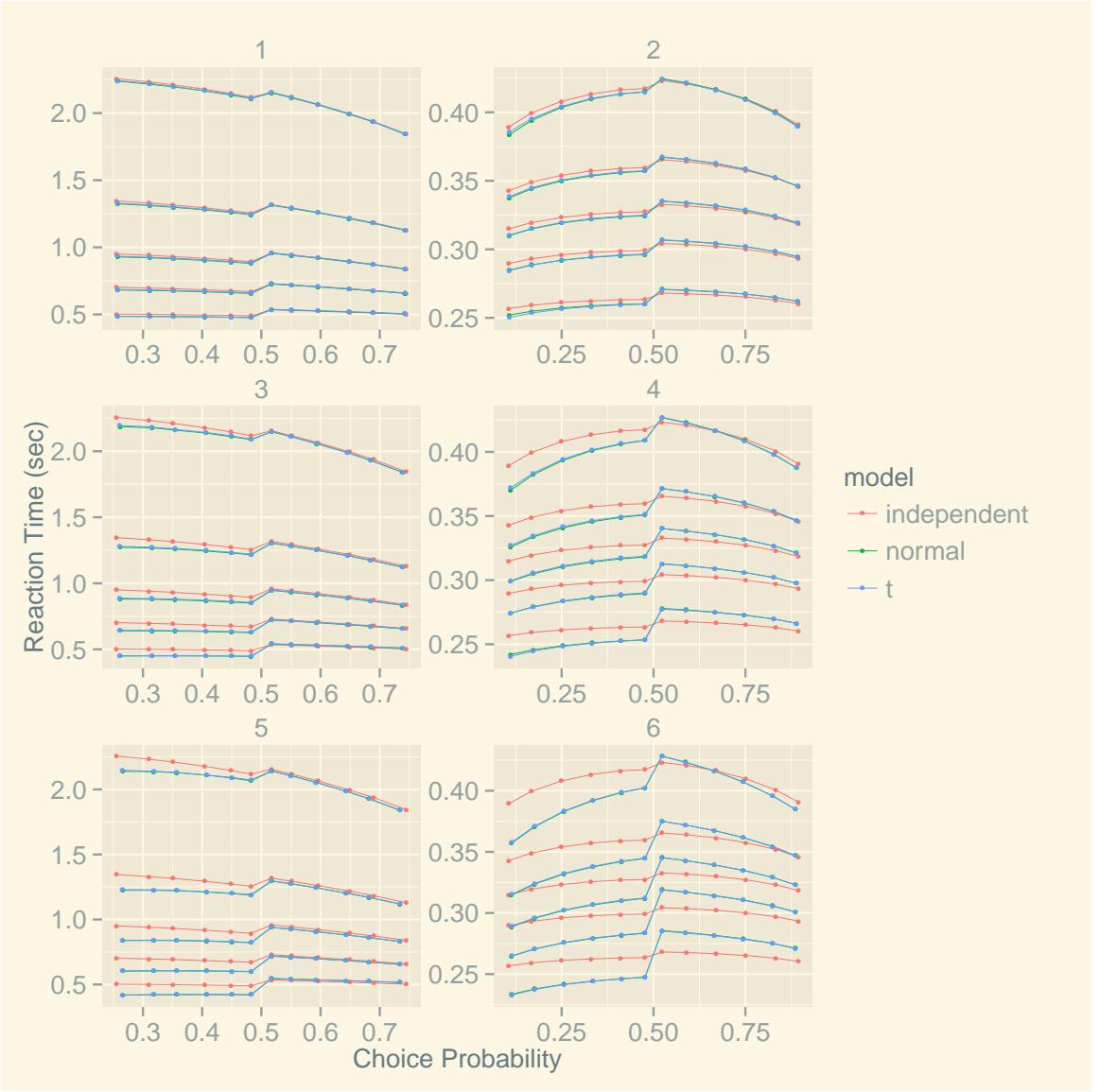


Figure 4.9: Behavioral predictions for the brightness discrimination task under $[--+]$ pattern of correlations among $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Correlations increase magnitude from top to bottom. Left column and right column correspond to “accuracy” and “speed”, respectively. Within each subplot, the lines connect reaction time quantiles conditional on response, and show how behavior changes from high to intermediate luminance.

observed asymmetry of response times (Ratcliff & McKoon, 2008; Ratcliff & Rouder, 1998).

However, results suggest that the asymmetry effect is not driven equally by all the parameters. In the discussion the parameters are arranged as follows: $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$. Comparing different patterns shows that $[+ - -]$ and $[+ + +]$ make bright responses faster, but $[- - +]$ reverses the effect to make bright responses slower. The first two patterns share positive correlation between rate of evidence accumulation δ and decision bias β , but correlation flips to negative in the third pattern. The magnitudes of asymmetries do differ from pattern to pattern, and are driven by relations between rate of evidence accumulation and decision bias to non-decision time. Hence, if asymmetry needs to be accounted for, the most consequential parameter for discrimination performance is $\rho_{\delta,\beta}$. This parameter was the one manipulated in Study A1 and showed large effects on absorption times relative to the independent copula model.

Predictive consequences of the correlation parameters depend strongly on their magnitude and available sample size. Small magnitudes, around 0.15, start showing effects which can be detected with large sample sizes (Dutilh et al., 2012, 2009; Ratcliff & Rouder, 1998), and in practice mark the threshold of detection. However, out of simplicity, if correlation magnitudes have small magnitudes, then their impact may be ignored and the standard model with independent copula used for analysis. However, if correlations increase to medium magnitudes around 0.50, the performance changes noticeably. The impact of correlations increases further with high correlations, like 0.85, which can produce really strong effects, noticeable even with small sample sizes. While the results above are not final, they strongly suggest

that ignoring the correlations with moderate or high values will lead to serious misfits and/or inaccurate estimates of all the other parameters when the data is fit with the standard model with independent copula.

The results, however, do suggest that not all data features and experimental conditions are better for inferring existence of parameter correlations. Across Figures 4.7, 4.8, 4.9, the most clear qualitative signs of asymmetry manipulation relative to the independent copula are reaction time quantiles under “speed” instruction and brightness levels are from near-neutral to moderate. However, quantitatively “accuracy” instructions lead to very large deviations, especially at higher quantiles and brightness levels for bright responses. The set of median quantiles may be useful statistics to gauge whether parameter correlations have been affected across conditions. Similarly, the quantiles could be used for posterior predictive checks to determine adequacy of the model’s assumptions.

Based on existence of strong effects in specific features of the data suggests that we can expand the range of data that sequential sampling models have to account for by explicitly manipulating them. In particular, inducing positive and negative correlations between starting points and drift rates can modulate the asymmetry between the two responses. To my knowledge, such a manipulation has not been done. One way to approach it would be to give bias instructions before the trial commences and then to pair it with stimulus quality in a positively or negatively correlated manner. Such an experiment is under current development to determine the extent of misfit if no correlation parameters are included.

Finally, if parameter correlations are important for explaining data, then selecting a distribution function other than the independent copula is required. Similar to

Study A1, we considered normal and t copulas, both belonging to the elliptical class, and found that their effects are very similar. Figures 4.7, 4.8, 4.9 show a strong overlap in green (normal) and blue (t) lines mostly based on a few millisecond difference. Besides similar predictive consequences, both copulas also provide a similar, flexible representation of parameter dependencies by allowing either direction and full range of correlation values. However, the normal copula should be preferred because by omitting the degrees of freedom parameter and having less mathematical operations, it is the more parsimonious and computationally efficient copula.

4.4 Study B - Benchmark Dataset Analysis

While Study A explored the theoretical predictions of the generalized diffusion models, Study B concentrates on their application. The purpose of Study B is to determine where the benchmark dataset collected by Ratcliff and Rouder (1998) has evidence for a correlation structure of decision and non-decision parameters under standard experimental conditions. The outline of the section is as follows. First, I describe an outlier filtering procedures to deal with extreme response times that can strongly bias estimates if not pre-processed. Next, I define a general Bayesian model, covering both independent and normal copula models, and a variant of a Metropolis-Hastings algorithm that was used to fit. There is also presentation of convergence diagnostics that were used on MCMC draws, and statistical methods to extract information about correlation structure of parameters from them. The section ends with results and their discussion.

4.4.1 Outlier filtering

An important feature of any dataset involving response times is presence of contaminants. Contaminants are observations arising from processes unrelated to research questions. Some of the recognized types of contaminants are fast guesses, slow guesses and delayed responses (Craigmile et al., 2010; Ratcliff, 1993; Ratcliff & Tuerlinckx, 2002; Vandekerckhove & Tuerlinckx, 2007, 2008; Vandekerckhove et al., 2011). Contaminants could arise due to non-compliance with experimental instructions or occasional loss of attention, among other processes.

To obtain accurate inferences about the process of interest, contaminants need to be either excluded or modeled. The proposed models are complex and initial testing may be more successful without including additional structure required to capture contaminants (Craigmile et al., 2010; Ratcliff & Tuerlinckx, 2002). Therefore, I used a set of common procedures to clean the response time data before fitting the models (Ratcliff, 1993).

Because contaminants mixed with regular response times cannot be identified, the most that is possible to do is to remove outliers. An outlier is defined as an extreme observation relative to the mean of the sample, specified depending on the type of outlier. There are fast and slow outliers that appear in a typical dataset. The fast outliers are usually to be understood as fast guesses, thus a useful heuristic to detect them is chance performance (Ratcliff & Tuerlinckx, 2002).

One way to use chance performance to remove fast outliers is to bin data into non-overlapping, continuous intervals. Starting at the lowest multiple of the bin size that includes the fastest response times, we can scan the ordered sequence of observations one interval at a time. For each interval, we calculate the accuracy rate. If accuracy is

at chance, then responses are fast guesses and they should be excluded. I will define “at chance” as an interval between 0.50 and 0.53, taking into account sampling error and the fact that the probability of exactly chance performance is 0. When accuracy rises above the chance interval, then responses in the bin will be assumed to be due to the process of interest and the lower bound of the bin forms the cut-off for excluding fast responses.

For the benchmark dataset, I used a window of 33 ms. The individuals and conditions vary a lot, so the method is more accurate if it is applied to all the participant-by-condition combinations as opposed to applying it to the pooled data. Applying the method resulted in a cut-off value of 200 ms for all three participants. The obtained cutoff is similar to previous studies analyzing the dataset (Ratcliff & Rouder, 1998). The total number of observations excluded with this cutoff is 491.

On the other hand, dealing with extremely long reaction times is different because it includes both slow guesses and delayed responses. In this case, the accuracy signature is not appropriate because delayed responses can be highly accurate because they are generated based on the decision process. A practical method to deal with slow outliers is to use a multiple of standard deviations from the mean as a cut-off.

To remove the slow outliers I used four standard deviations from the mean as a cut-off. As for the fast outliers, I applied the procedure to all participant-by-condition combinations. The resulting cut-offs are also similar, but somewhat less stringent than in prior studies, leaving a few slower outliers in the processed sample. Application of the cut-off removed 372 observations. Overall, applying the two outlier procedures brought the total number of excluded observations to 3.2% of the full dataset.

4.4.2 Bayesian Models

Given that elliptical copulas differed little in their predictions, I decided to fit only independent and normal copula models. The normal copula is superior to t copula because it has one less parameter and is computationally faster to evaluate. With the data to be modeled established, next I describe Bayesian models that I fit to estimate correlation parameters.

Given a rich data set and up to free 10 parameters, multiple variations of the two Bayesian models could be constructed by making different assumptions about how subject, stimulus quality and speed-accuracy instructions influence cognitive processing. I relied on results from previous modeling and an EEG study to hypothesize how experimental manipulations affect processing. Particularly, I assumed that effects of instructions on processing are complex and effect multiple parameters while stimulus quality changes affect only drift rates. Based on this assumption, I left several more parameters free than is usual with applications of the standard Ratcliff model. Finally, while a hierarchical version of each model can be defined and fitted, I fitted separate one-level models for each participant because of computational demands in obtaining posterior values and having a large dataset that makes information sharing across subjects unnecessary for precise estimates.

I use the following variables to denote features of the experimental design. The dataset contains $l = 1, 2, 3$ participants. Each participant received a block-wise instruction $k = 0, 1$, where 0 is speed and 1 is accuracy, combined with a trial-wise brightness level $j = 0, 1, \dots, 32$. Therefore, for each condition, a participant would have $N_{j,k,l}$ observations. The observations come from a model $M = 1, 2$, referring to independent and normal diffusion model, respectively.

Recall that the joint density of response times and responses is a mixture of the Wiener first-passage density and a copula-based density. The joint density for model M of a single observation $(t_{i,j,k,l}^{rt}, r_{i,j,k,l})$ conditioned on instruction k and brightness proportion p_j is

$$f_W^M(t_{i,j,k,l}^{rt}, r_{i,j,k,l} \mid p_j, \alpha_{k,l}, \nu_{j,k,l}, \eta_{k,l}, \lambda_{k,l}, \gamma_{k,l}, \chi_{k,l}, \phi_{k,l}, \rho_l^{\delta, \tau^{er}}, \rho_l^{\beta, \delta}, \rho_l^{\beta, \tau^{er}}), \quad (4.9)$$

a mixture of Wiener densities. The indices of parameters reflect individual differences across all parameters. The instructions are assumed to influence thresholds, means of drift rates and bias, as with previous successful model fits of the benchmark data and similar experiments (Ratcliff & McKoon, 2008; Ratcliff & Rouder, 1998; Vandekerckhove & Tuerlinckx, 2007; Vandekerckhove et al., 2011; Wagenmakers, 2009). The instructions are also assumed to influence non-decision times, as suggested by an increase in the lateralized readiness potential under accuracy instructions (Rinkenauer et al., 2004), and single cell recording data that shows effect of instructions on several nodes in the neural network underlying task performance (Heitz & Schall, 2013). The brightness level only influences drift rates (Ratcliff & McKoon, 2008; Vandekerckhove et al., 2011; Voss et al., 2004). When $M = 1$ correlation parameters are set to 0.

Similar to Study A, I modeled effects of brightness proportion and instruction on drift rates with a regression function. I assumed that mean drift rate regresses on brightness proportion through a sigmoidal-shaped Weibull function

$$\nu_{j,k,l} = -\nu_{k,l}^{lo} + (\nu_{k,l}^{hi} - \nu_{k,l}^{lo}) \left(1 - \exp \left(-[p_j / \nu_{k,l}^{sc}]^{\nu_{k,l}^{sh}} \right) \right), \quad (4.10)$$

where $\nu_{j,k,l}$ is the drift rate, p_j is the brightness level, $\nu_{k,l}^{lo}$ is the lower asymptote, $\nu_{k,l}^{up}$ is the upper asymptote, $\nu_{k,l}^{sc}$ is the scale parameter and $\nu_{k,l}^{sh}$ is the shape parameter. The parameter indices express individual differences and effect of speed/accuracy

manipulation on drift rates through Weibull parameters (Vandekerckhove et al., 2011).

Assuming a functional relation replaced estimation of 66 drift rates with 8 Weibull parameters for each subject.

Notice that the joint density of behavioral data defined above already assumes the integration of the Wiener density with respect to the distribution of parameters, with or without dependencies. Instead of using a Monte Carlo estimate as was done in Study A, I used a deterministic approximation method called adaptive quadrature because it is much faster in three dimensions while being very accurate (Berntsen, Espelid, & Genz, 1991; Genz & Malik, 1980). Similar to the Monte Carlo estimator, quadrature methods approximate an integral with a weighted average, such that

$$\int_{\mathcal{X}} g(x)dx \approx \sum_{i=1}^N w_i g(x_i), \quad (4.11)$$

but weights w_i can be other than $1/n$ and nodes x_i are set deterministically. When evaluating the likelihood value I borrowed a routine from a C package cubature developed by S. G. Johnson and Narasimhan (2013) to link with a custom C++ program that evaluates the full Bayesian model.

The remaining part of the Bayesian models are the priors for all the parameters. All parameters except for the correlation parameters are shared by the two models. My approach to specifying the priors was driven by the practical reason of obtaining fast and accurate posterior values because of the considerable computational cost of evaluating the likelihood for the whole dataset for each subject at each iteration of MCMC. To compliment the proposal density of the sampling algorithm, defined over the full real space (next section), and let the data dominate the inferences, I reparameterized non-correlation parameters and assigned diffuse priors.

Except for lower and upper boundaries, all the non-correlation parameters take values on the positive real line without restrictions, I transformed them with a log function to the full real line. The boundaries range over the whole line, with a restriction that lower bounda is smaller than the upper bound, so no log transformation is needed. Each of the non-correlation parameters were assumed to come from a multi-variate normal distribution with no correlation and large variances. The correlation parameters, however, are restricted to a convex set included in a 3-dimensional cube with sides ranging from -1 to 1 , and cannot be reparameterized to remove the restrictions. The restrictions guarantee proper correlation matrices, which have to be positive definite. An appropriate prior for correlation parameters is the LKJ density, defined up to a normalizing constant as

$$f_{lkj}(\rho_1, \rho_2, \rho_3) \propto \left(1 - \sum_{i=1}^3 (\rho_i^2) + 2 * \prod_{i=1}^3 (\rho_i) \right)^{(\psi-1)}, \quad (4.12)$$

where ρ_i are random correlation parameters and $\psi > 0$ is the free parameter (Lewandowski, Kurowicka, & Joe, 2009). Notice that if $\psi = 1$, the LKJ density is uniform over the convex set, if $\psi < 1$ density concentrates on zero, and if $\psi > 1$ density concentrates around higher values

Given specifications of the likelihood and prior, assumptions of independence for observations and parameters allows to formulate the full Bayesian model as a product of densities. In symbolic form, the overall Bayesian model for model M is

$$\begin{aligned} f_W^M(\boldsymbol{\alpha}^*, \boldsymbol{\nu}^*, \boldsymbol{\eta}^*, \boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*, \boldsymbol{\chi}^*, \boldsymbol{\phi}^*, \boldsymbol{\rho}^{\delta, \tau^{nd}}, \boldsymbol{\rho}^{\beta, \delta}, \boldsymbol{\rho}^{\beta, \tau^{nd}} | \mathbf{t}^{rt}, \mathbf{r}, \mathbf{p}) &\propto \\ \prod_{l=1}^3 \prod_{k=1}^2 \prod_{j=0}^{32} \prod_{i=1}^{n_l} f_W^M(t_{i,j,k,l}^{rt}, r_{i,j,k,l} | p_j, \alpha_{k,l}^*, \nu_{j,k,l}^*, \eta_{k,l}^*, \lambda_{k,l}^*, \gamma_{k,l}^*, \chi_{k,l}^*, \phi_{k,l}^*, \rho_l^{\delta, \tau^{nd}}, \rho_l^{\beta, \delta}, \rho_l^{\beta, \tau^{er}}) &\times \\ f_{MVN}(\alpha_{k,l}^*, \nu_{j,k,l}^*, \eta_{k,l}^*, \lambda_{k,l}^*, \gamma_{k,l}^*, \chi_{k,l}^*, \phi_{k,l}^* | \boldsymbol{\mu}, \boldsymbol{\Sigma}) &\times \\ f_{LKJ}^M(\rho_l^{\delta, \tau^{er}}, \rho_l^{\beta, \delta}, \rho_l^{\beta, \tau^{er}} | \psi), \end{aligned} \quad (4.13)$$

where f_W^M is the mixture Wiener density, f_{MVN} is the multivariate normal, f_{LKJ} is the LKJ prior, and starred parameters indicate log transformations. For the independent copula model, when $M = 1$, $f_{LKJ} = 1$ and correlation parameters are 0. (μ, Σ, ψ) are free parameters of the overall prior. To express little information about non-correlation parameters and no information about correlation parameters, the free parameters were set such that $\mu = \mathbf{0}, \Sigma = 100\mathbf{I}, \psi = 1$.

The Bayesian model describes the data-generating process for observations for the three subjects and their parameters. Lack of hierarchy and independence assumptions imply that posterior density can be factored into three separate densities, one for each subject. Hence, it is possible to sample the posteriors for each subject separately. In the next subsection, I describe a variant of the Metropolis-Hastings algorithm that fits well with the properties of the Bayesian model and was used to sample from individual posterior densities.

4.4.3 MCMC Sampler

Selecting a specific Metropolis-Hastings sampler requires specifying a sampling scheme and a proposal density. A sampling scheme defines an iteration of a sampler in terms of how the dimensions of a Markov chain are updated. It is possible to update all dimensions simultaneously or cycle through them sequentially in a deterministic or random fashion. Once a sampling scheme is chosen, a proposal density will define how we update the position of the Markov chain during an iteration. Both choices are best motivated by considering properties of the model to be fit.

Two central features of the fitted Bayesian models that constrain sampler choice are an expensive likelihood evaluation and a correlated parameter space. At each

iteration of the MCMC chain, a joint likelihood value has to be computed for thousands of observations, with each requiring a triple integral of the Wiener probability density function (itself approximated). Even after writing the likelihood calculation in compiled language and parallelizing the program across 60 cores, the time can still take around 5 seconds at a reasonable accuracy level. Such durations multiplied by thousands of iterations needed to find and explore the equilibrium density carry a substantial computational cost. Hence, samplers that minimize likelihood evaluations, and hence overall computational time, are preferred.

Another constraint on the sampler choice is the correlated parameter space. Previous fits of the Ratcliff model and other sequential sampling models show that estimated parameters are correlated Ratcliff and Tuerlinckx (2002); Turner, Sederberg, Brown, and Steyvers (2013). In our case, posterior distribution has a covariance matrix with non-zero off-diagonal values. To enable efficient movement in the parameter space the proposal density underlying a sampler has to be tuned to the variances and especially covariances of the parameters. Thus, samplers that can be easily hand-tuned or that automatically tune the covariance matrix of the proposal density are preferred.

Both constraints are matched well with adaptive Metropolis-Hastings (AMH) algorithm (Haario, Saksman, & Tamminen, 2001). AMH uses global updates and requires a single chain. The proposal density is multivariate normal centered at the current value and scaled covariance matrix calculated based on the full history of sampled states. The adaptive covariance helps the algorithm to propose new values efficiently by taking into consideration the geometry of the high density region, and does so

automatically. The basic algorithm proposed by Haario et al. (2001) works well, but I added two features to improve sampling from the Bayesian diffusion models.

One modification involves making the scaling factor of the covariance matrix adaptive, too. The covariance of the proposal density is defined as $\sigma\hat{\Sigma}$, and under restrictive conditions the optimal value for σ is $2.34^2/|\boldsymbol{\theta}|$ (Gelman et al., 2014). The conditions include normality of the posterior and independence of the components. The first condition is likely to hold, but the second condition is not. Instead of manually searching for a value of σ to achieve a good acceptance rate and efficient sampling, we can make the scaling factor adaptive, too.

Adaptation of the scaling factor can be guided by the desired acceptance rate (Roberts & Rosenthal, 2009). For 20 or more parameters, acceptance rates ranging from 10% – 25% have been found to work well in practice, hence providing a target for the scaling factor (Gelman et al., 2014; Roberts & Rosenthal, 2009). In sampling from the posteriors, I adapted the scaling factor at iteration n such that

$$\sigma^* = \exp(\log(\sigma^{(n)}) + 1/n^{(1/2)}(\exp(\log(\alpha^{(n)}) - \alpha)),$$

where $\sigma^{(n)}$ is an estimate of the scaling factor at iteration n , $\alpha^{(n)}$ is the Metropolis-Hastings probability at iteration n , and α is the target acceptance rate. When fitting models, I set $\alpha = 0.15$ for both independent and normal copula models.

The second feature to adapt AMH to the Bayesian diffusion models are reflective boundaries for the correlation parameters. Recall that AMH proposal density is multivariate normal, so it can propose values outside the $[-1, 1]$ domain over which correlations vary. When encountered with a proposal outside of the sample space, the standard AMH is guaranteed to reject the proposal (0 probability event) and keep the chain in place for that iteration. To lower the substantial overall computational

time of sampling from the Bayesian diffusion models, it would help to avoid proposing beyond the boundaries.

A simple solution is to reflect the proposed values back into the sample space by the same amount as they fall out. Suppose a parameter θ has boundaries a and b , where $a < b$. Then if the algorithm proposes $x < a$ replace it with $x^* = 2a - x$. If the proposal is $x > b$ replace it with $x^* = 2b - x$. Most of the time a single reflection is sufficient, but the step can be applied recursively until success. Reflection is a symmetric operation, so the Metropolis-Hastings ratio is not changed (Yang & Rodríguez, 2013).

Even with reflection step added to the standard AMH, in principle, one more inefficiency remains. Correlation parameters do not only take values in a restricted range, but only in a convex subset of the cube. Hence, AMH can still propose 0 probability values that are guaranteed to be rejected. This problem may get severe if the high density region is around the boundaries of the convex set, but in the fits of the benchmark dataset the sampler remained almost exclusively in the convex set. Thus, at least in this application, AMH is free from the inefficiency of sampling outside the positive definiteness region.

The pseudocode of the modified AMH is listed under Algorithm 2. $\boldsymbol{\theta}$ marks parameters of either the standard or generalized Ratcliff model. The algorithm is initiated at some vector with positive density. Before adaptation can begin K values are sampled with a scaled identity matrix used for the covariance matrix to build a sufficient sample. Each sample is reflected if it falls out of boundaries. σ^2 is set to something small to ensure high acceptance rate. In the fits reported below, $\sigma^2 = 0.1^2 / |\boldsymbol{\theta}|$ and $M = 200$.

After K iterations, algorithm begins to adapt. $\hat{\sigma}^2$ is calculated based on the formula defined above. Covariance is updated using an unbiased estimator such that for $j - 1$ iterations

$$\hat{\Sigma}^{j-1} = \frac{1}{j-2} \sum_{i=0}^{j-1} (\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}})^T. \quad (4.14)$$

As in the initial period, proposed values are reflected if necessary. The final part of the code is the Metropolis-Hastings acceptance mechanism.

Algorithm 2 Modified AMH Pseudocode

Require: Initiate at $\boldsymbol{\theta}^0 \in \pi(\boldsymbol{\theta}^0 | \mathbf{x}) > 0$

```

for  $j = 1, 2, \dots, N$  do
    if  $j < K$  then
        Sample  $\boldsymbol{\theta}^* \sim \mathcal{N}(\boldsymbol{\theta}^{j-1}, \sigma^2 \mathbf{I})$ 
        Reflect  $\boldsymbol{\theta}' = \text{reflect}(\boldsymbol{\theta}^*)$ 
    else
        Update  $\hat{\Sigma}$ 
        Sample  $\boldsymbol{\theta}^* \sim \mathcal{N}(\boldsymbol{\theta}^{j-1}, \hat{\sigma}_1^2 \hat{\Sigma}^{j-1})$ 
        Reflect  $\boldsymbol{\theta}' = \text{reflect}(\boldsymbol{\theta}^*)$ 
    end if
    Sample  $\alpha \sim \mathcal{U}(0, 1)$ 
    Update  $\hat{\sigma}^2$ 
    if  $\alpha < \pi(\boldsymbol{\theta}^* | \mathbf{x}) / \pi(\boldsymbol{\theta}^{j-1} | \mathbf{x})$  then
        Set  $\boldsymbol{\theta}^j = \boldsymbol{\theta}^*$ 
    else
        Set  $\boldsymbol{\theta}^j = \boldsymbol{\theta}^{j-1}$ 
    end if
end for

```

4.4.4 Convergence Diagnostics

A Metropolis-Hastings algorithm enable us, in principle, to sample from posterior densities of the proposed models. The Markov chain theory combined with the structure of Metropolis-Hastings ensures that, given an arbitrary vector of initial values

from the parameter space, Markov chains will converge to the equilibrium density. However, the basic theory says nothing about when convergence will take place, information crucial to having accurate inferences.

In a limited number of cases, which are often not useful for real-world applications, there is some formal theory that provides a lower bound on the number of iterations for a given approximation error (Gamerman & Lopes, 2006; Robert & Casella, 2005). More typically, the decision about when convergence occurred are determined by heuristic methods, often called convergence diagnostics, applied post-hoc to a MCMC sample (Gelman et al., 2014). The conditions checked by heuristic methods are necessary for convergence, so if they hold it does not mean that convergence holds, but their failure does mean that convergence has not occurred. Working with any difficult sampling problem requires heuristic methods.

We can never be sure about the MCMC output, but we can do graphical and statistical tests that suggest convergence. One simple method involves plotting likelihood or log-likelihood values, $L(\boldsymbol{\theta}_1 \mid \mathbf{x}), \dots, L(\boldsymbol{\theta}_N \mid \mathbf{x})$, against iteration index, $i = 1, 2, \dots, N$. As the chain approaches the point of convergence the iteration series should stabilize and fluctuate within a limited range as the chain continues to explore the parameter space. This method is a useful heuristic for convergence, but it does not reveal when all the parameters have stabilized or whether there are multiple stable points.

To address the limited information of the likelihood values, we can use a trace plot, which plots the ordered sequence of draws $\{x_1, x_2, \dots, x_N\}$ for each parameter against the iteration index $i = 1, 2, \dots, N$. The signature of convergence is when the MCMC chain of all the parameters stabilizes around some value, in case of a unimodal

posterior density, or a set of values, in case of a multimodal posterior density, and has a stable variance. Stable variance is especially important in case of adaptive algorithms, that take some time to obtain an accurate estimate of the covariance matrix of the posterior density. All the draws prior to stable point are treated as a so-called burn-in period and discarded because they are draws from non-equilibrium distributions. The remaining draws can be used for posterior inference.

In addition to a graphical tests, it is common to do some sort of formal statistical test of convergence. A popular convergence test is Gelman-Rubin's test (Gelman & Rubin, 1992), but it requires multiple chains and would be too computationally intensive for the current application. An alternative is Geweke's test based on a difference between means of the initial and final part of the chain (Geweke, 1991). Under the stationarity hypothesis, the mean difference is approximately normally distributed with expected value of 0. Applied after the burn-in draws are dropped, parameters that have means within two standard errors of the mean difference are consistent with convergence.

When examining the chains for the Bayesian diffusion models, I used the graphical and statistical tests described in this subsection. All methods have been implemented in the R package CODA (Plummer, Best, Cowles, & Vines, 2006), and can be easily applied to out from custom Metropolis-Hastings algorithms by converting the sampling result into a MCMC S3 object.

4.4.5 Characterizing Parameter Dependencies

The overall question motivating this study is the existence and nature of correlations among cognitive parameters important for simple decisions. One way to

characterize the correlation structure is to quantify probabilities of correlations in some direction. Using posterior samples, I estimated probabilities of positive (or negative) correlation for each parameter using the Monte Carlo estimator, which is still appropriate even though samples are dependent (Givens & Hoeting, 2013; Robert & Casella, 2005). For example, the probability that initial position bias and drift rate are positively correlated is

$$P(\rho^{\beta,\delta} > 0) \approx \frac{1}{N} \sum_{i=1}^N 1_{\{\rho_i^{\beta,\delta} > 0\}}, \quad (4.15)$$

where N is the number of MCMC draws kept after the burn-in. The probability of correlation is a rough measure of evidence for the existence of a correlation structure.

In addition to quantifying probabilities of correlations, we want to understand plausible correlation values suggested by the data. I estimated plausible values with a highest posterior density (HPD) interval (Casella & Berger, 2002; Gelman et al., 2014). An HPD interval is a set of parameter values $\{\theta : f(\theta | \mathbf{x}) \geq k\}$, where k satisfies

$$\int_{\theta: f(\theta | \mathbf{x}) \geq k} f(\theta | \mathbf{x}) d\theta = 1 - \alpha. \quad (4.16)$$

Estimating this interval from a posterior MCMC sample requires solving an optimization problem

$$p^* = \operatorname{argmin}_p [F^{-1}(1 - \alpha + p) - F^{-1}(p)], \quad (4.17)$$

where p is the lower tail probability, α controls the credibility level and F^{-1} is a quantile function. I used the R package CODA to find p (Plummer et al., 2006).

4.4.6 Results

Before presenting posterior estimates I describe convergence diagnostics based on data from subject “kr” and fits of the normal copula model. The other two subjects

and fits with the independent copula model show similar patterns. Convergence diagnostics provide evidence that sampled values can be used for inference and also reveals how well AMH algorithm works with the fitted Bayesian models. Figure 4.10 contains log-likelihood values for each accepted vector of parameters for the normal copula diffusion model. The rise and stabilization of log-likelihoods provides evidence for successful convergence.

The initial run where the chain moves to the high density region and additional iterations used to estimate the covariance matrix of AMH have been dropped as burn-in. As an example, Figure 4.11 shows traceplots of correlation parameters for the normal copula model. The chains show movement within a restricted range around a single mean. The chain movement is consistent with uniform, approximately symmetric posterior densities. Other parameters, for both models, show similar traceplots.

In addition to graphical methods, Figure 4.12 shows evidence for convergence using the Geweke statistic for the normal copula model. Applied after the burn-in values have been dropped, the statistic is equal to a difference of means for the early and later parts of the chain, calculated independently for each parameter. I used the initial 10% of samples for the first mean and the latter 50% for the second mean. The no mean difference hypothesis is supported when Geweke statistic is within two standard deviations from 0. Points in Figure 4.12 are all within the null-hypothesis range, hence the chains are behaving as if equilibrium density has been reached. Based on Geweke statistics and graphical checks, sampled parameters can be used for inference.

Before presenting estimation results, I mention two features of the simulated chains relevant to the choice of algorithm for the fitted Bayesian models. Figure 4.13 shows

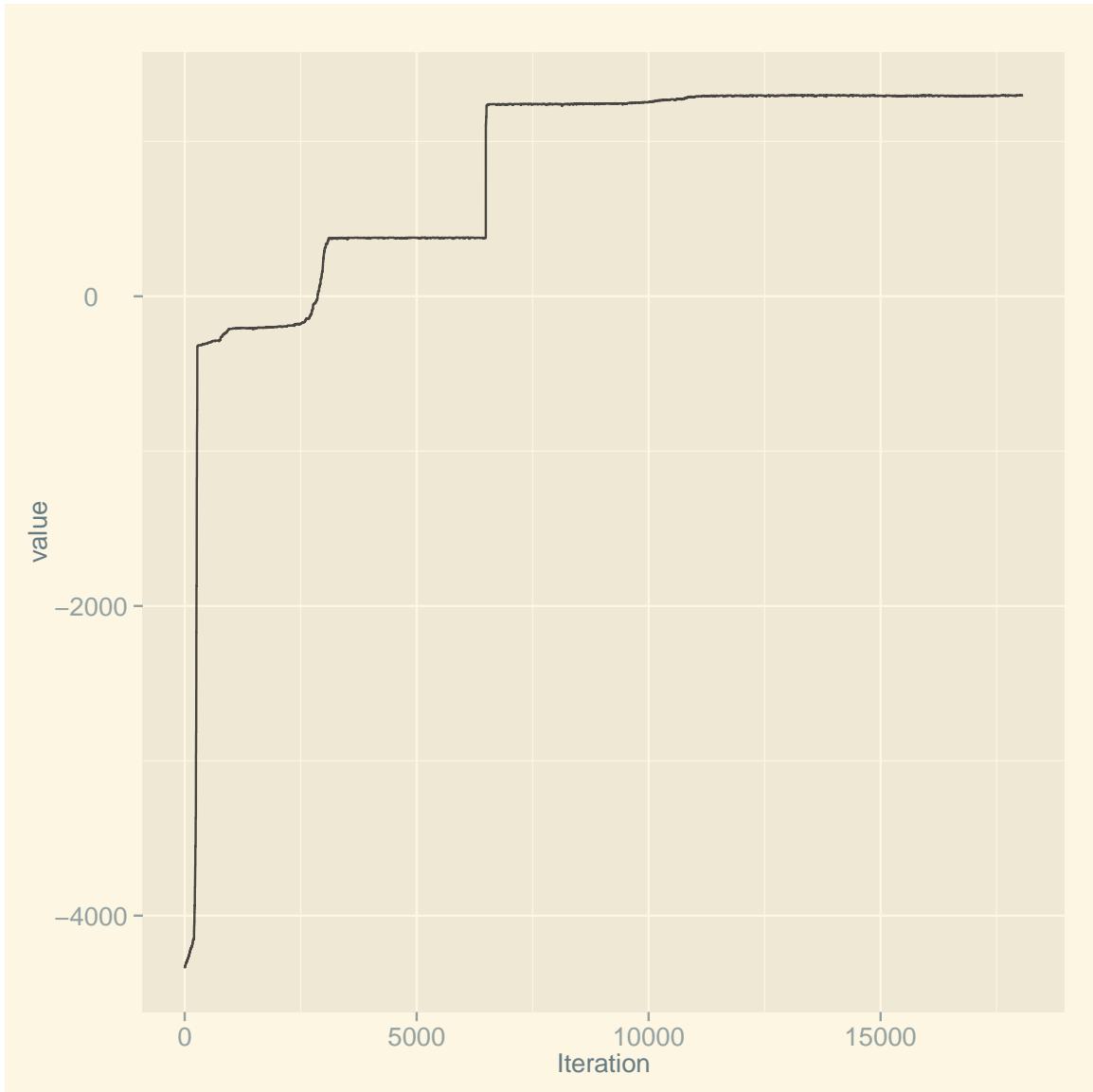


Figure 4.10: Log-likelihood values corresponding to points in the MCMC chains for the normal copula model. Initial run of the chain leads to several increases in the value of log-likelihood followed by a sequence of values in a stable range. The pattern is consistent with successful convergence.

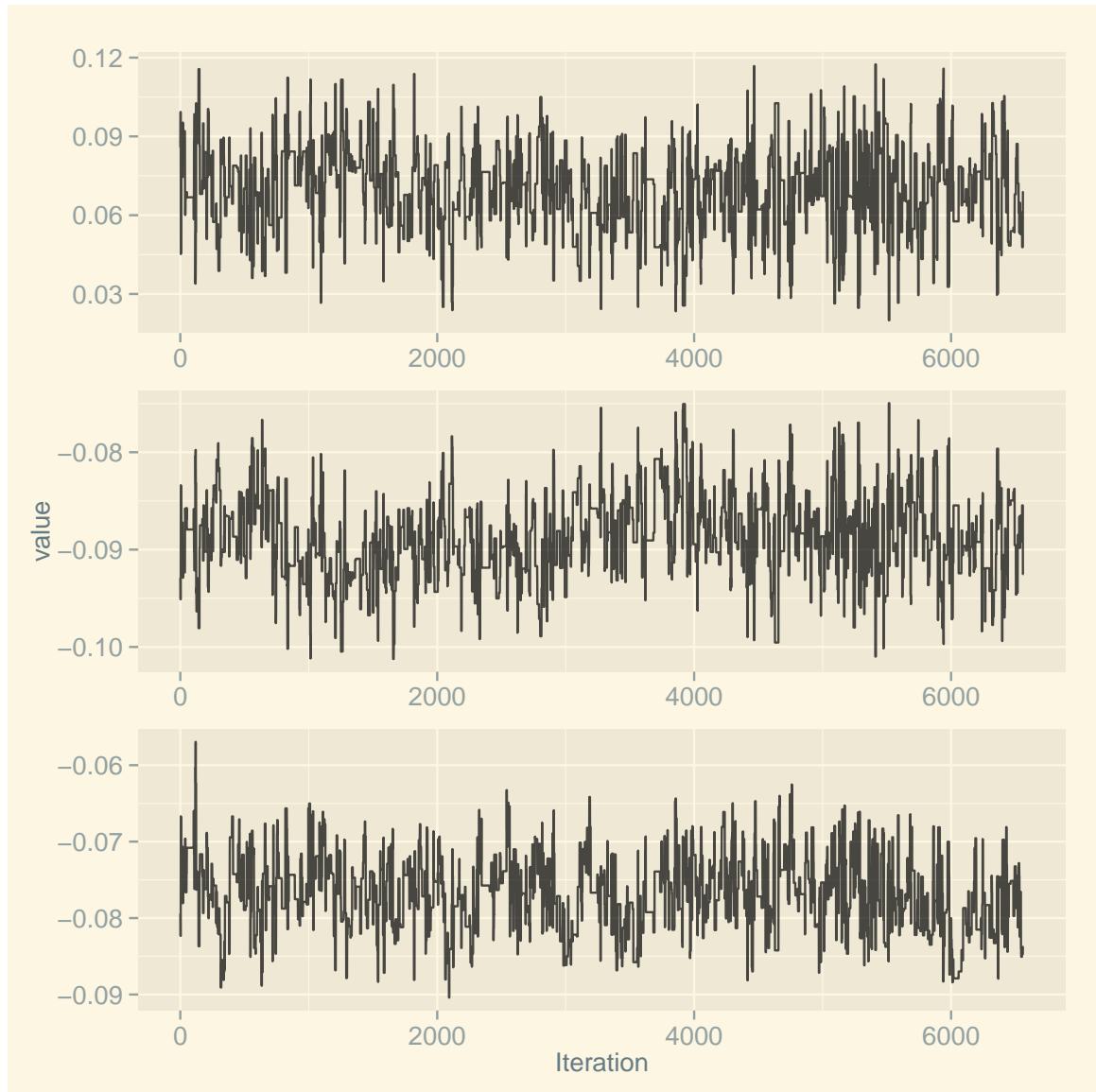


Figure 4.11: MCMC chains for $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$ without the burn-in iterations for the independent copula model. The stability of traces indicates convergence and rapid movement shows that AMH algorithm is appropriate for the model.

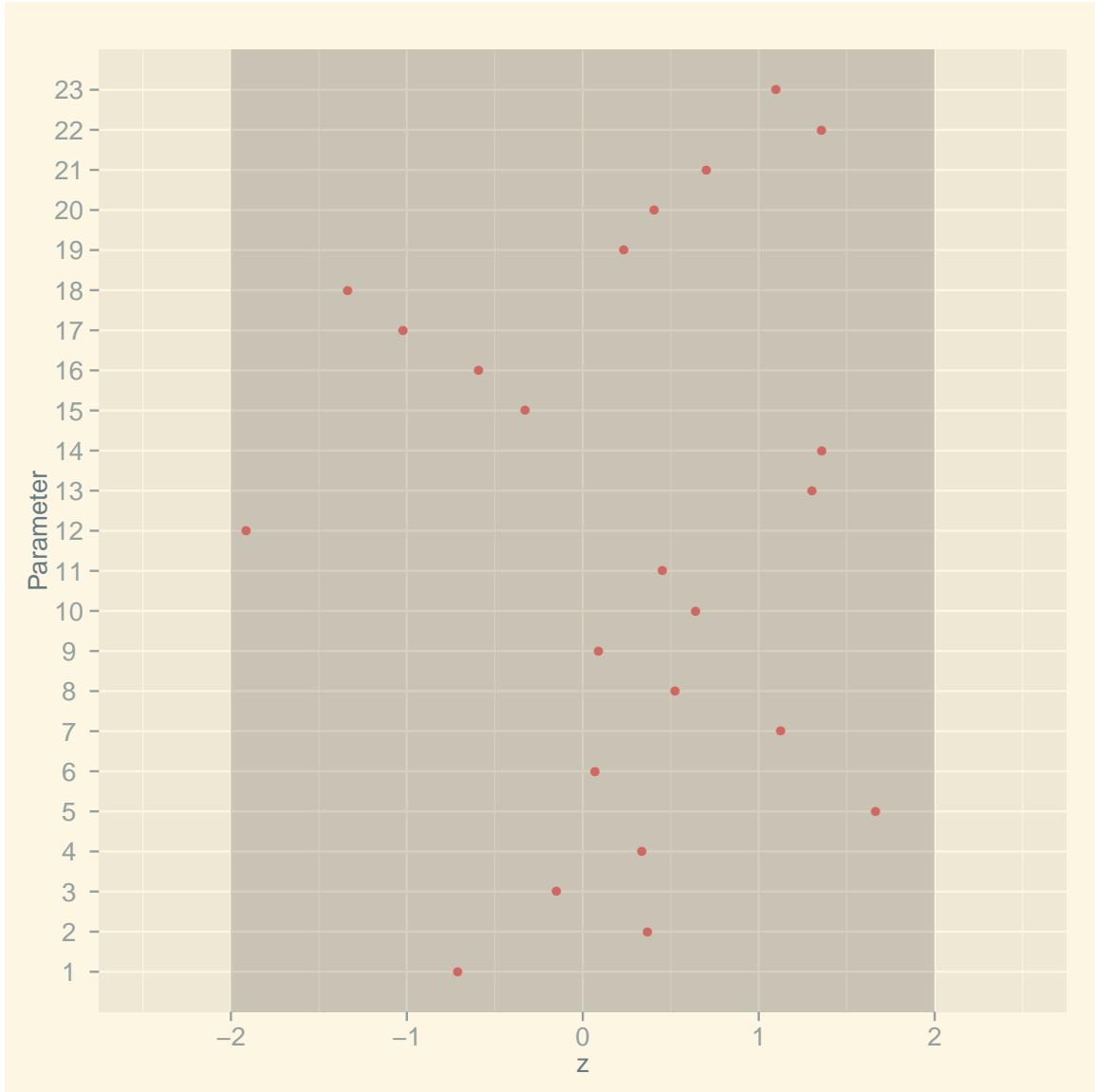


Figure 4.12: Values of Geweke statistic for all 23 parameters of the normal copula model. Geweke statistic is a difference in means of later minus the early part of a chain. All values are within two standard deviations indicating convergence to the equilibrium density.

the cross-correlations matrix for the 23 fitted parameters under the normal copula model. The parameters show a lot of positive and negative correlations, with values ranging from -0.965 to 0.961 . For example, lower and upper bounds are due to correlation between standard deviation parameters of decision bias and non-decision time, and drift rate and decision bias, respectively. Similar range holds for the independent copula model. Correlated parameter space has been reported for other sequential sampling models and confirms one of the reasons for choosing AMH algorithm (Ratcliff & Tuerlinckx, 2002; Turner, Sederberg, et al., 2013).

Another feature of the chains relevant to sampler choice is serial dependencies across iterations. As an example, auto-correlation levels for the three correlations parameters, as shown in Figure 4.14, show acceptable chain mixing. Auto-correlation varies across parameters, but remains around or below lag of 100. Similar auto-correlation levels hold for other parameters, of both models. At the obtained auto-correlation levels, chain of a few thousand iterations is sufficient for accurate inferences without thinning. With more iterations AMH would get even better, but in this particular application the practical constraints and computational time did not permit longer runs. Thus, AMH algorithm is a useful choice in fitting Bayesian models with Wiener density, with or without correlated parameters.

The estimates of non-correlations parameters, based on the normal copula, are summarized in Table 4.4. For each parameter, there is a mean and HDI bounds. The estimates are consistent with estimates reported in previous studies in their pattern and magnitudes even though half the dataset was used (Vandekerckhove et al., 2011). Data for the other two subjects lead to mostly a similar posterior distribution of the non-correlation parameters, with exceptions discussed below.

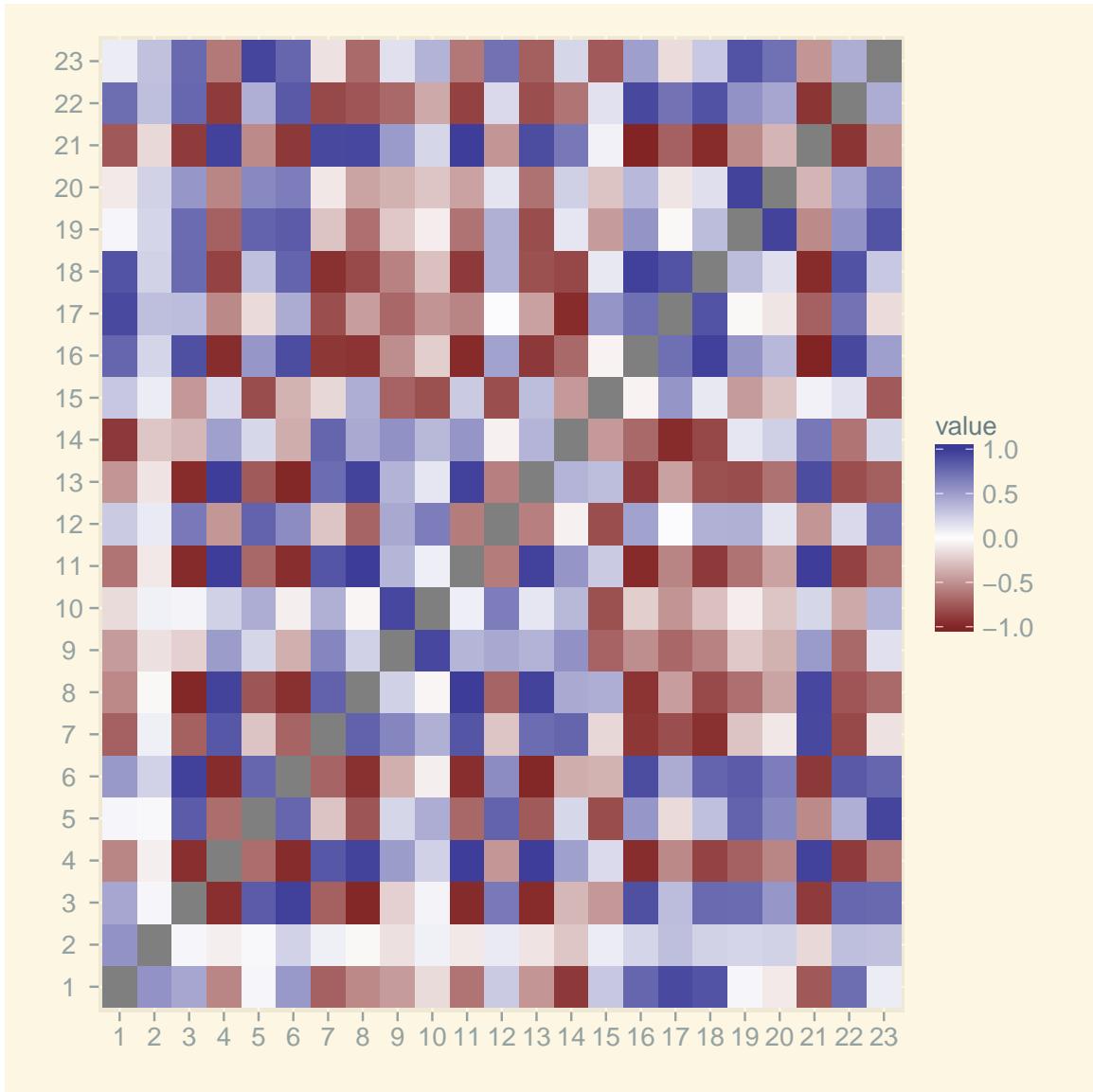


Figure 4.13: Cross-correlations of all 23 parameters based on MCMC chains without the burn-in period. Several pairs of parameters show small to high correlations consistent with other sequential sampling models, and support using adaptive algorithms for efficient sampling.

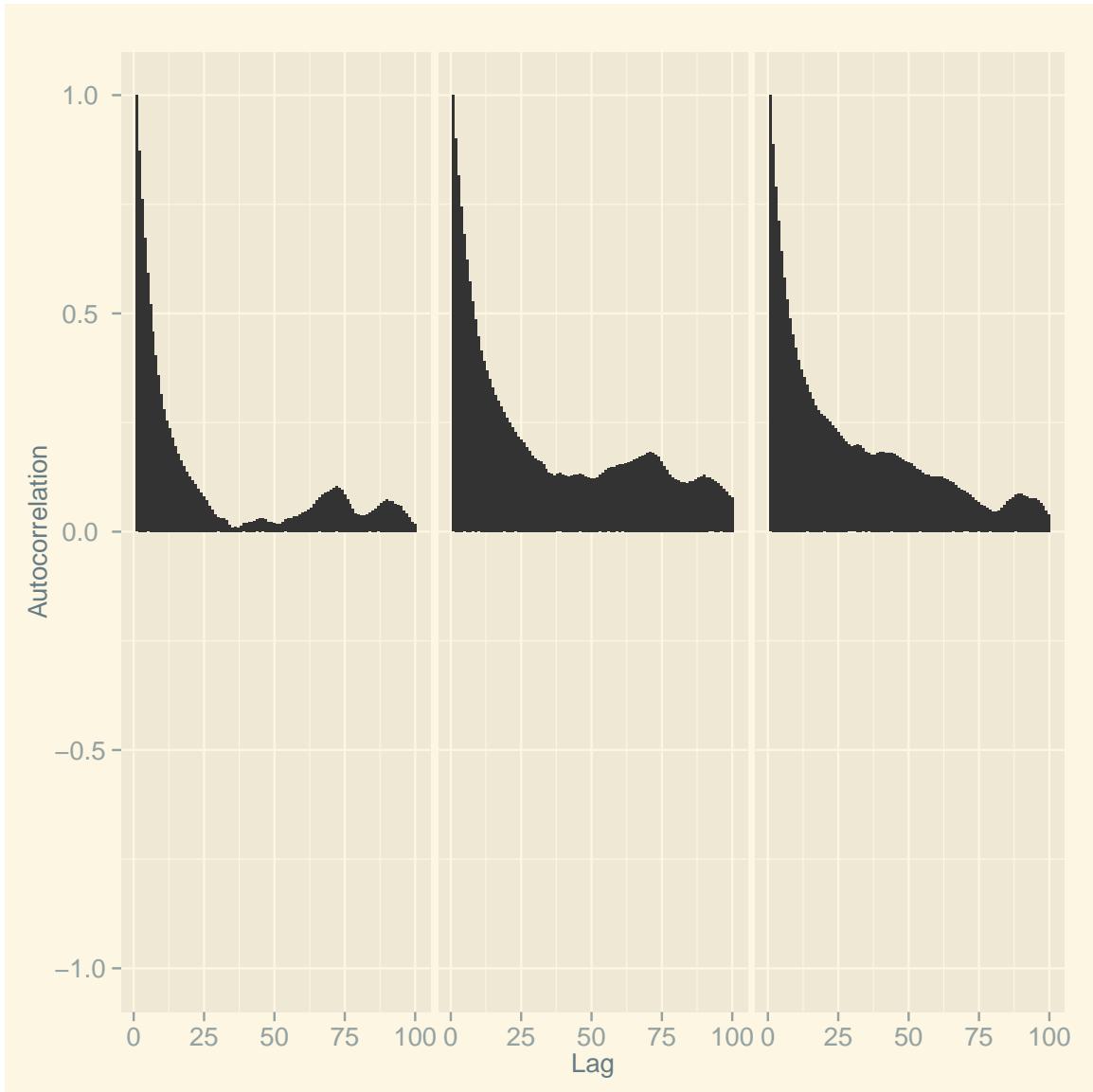


Figure 4.14: Autocorrelation in MCMC chains of $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$ based on several thousand iterations after burn-in. The levels are similar for other parameters. The levels are sufficiently small for accurate estimations, and would drop further with longer runs of the AMH sampler.

Parameter	Lower	Mean	Upper	Parameter Code
α^{ACC}	0.168	0.169	0.171	upper threshold
ν_{lo}^{ACC}	-0.241	-0.229	-0.217	smallest drift rate
ν_{hi}^{ACC}	0.408	0.424	0.443	largest drift rate
ν_{sh}^{ACC}	4.157	4.210	4.265	shape of drift rates
ν_{sc}^{ACC}	0.556	0.563	0.568	scale of drift rates
η^{ACC}	0.123	0.124	0.125	standard deviation of drift rates
λ^{ACC}	0.396	0.403	0.409	mean decision bias
γ^{ACC}	0.061	0.062	0.062	standard deviation of decision bias
χ^{ACC}	0.293	0.297	0.301	mean non-decision time
ϕ^{ACC}	0.039	0.039	0.040	standard deviation of non-decision time
α^{SPD}	0.045	0.047	0.048	upper threshold
ν_{lo}^{SPD}	-0.571	-0.565	-0.559	smallest drift rate
ν_{hi}^{SPD}	0.488	0.498	0.506	largest drift rate
ν_{sh}^{SPD}	5.575	5.615	5.640	shape of drift rates
ν_{sc}^{SPD}	0.459	0.466	0.472	scale of drift rates
η^{SPD}	0.111	0.114	0.117	standard deviation of drift rates
λ^{SPD}	0.364	0.369	0.376	mean decision bias
γ^{SPD}	0.062	0.063	0.063	standard deviation of decision bias
χ^{SPD}	0.270	0.273	0.275	mean non-decision time
ϕ^{SPD}	0.040	0.040	0.040	standard deviation of non-decision time

Table 4.4: Summary of the posterior distribution of the non-correlation parameters of the normal copula model for subject “kr”. Each parameter is summarized by a mean and lower/upper boundaries of its HDI. Note: ACC is accuracy condition and SPD is speed condition.

Table 4.4 shows several recurrent trends in analyses of two choice tasks (Ratcliff & McKoon, 2008; Wagenmakers, 2009). One effect of speed-accuracy manipulation is to manipulate how cautious people are as expressed by the evidence threshold, with lower value for “speed” and higher value for “accuracy” emphasis. However, in line with the assumption that speed-accuracy instruction is a complex manipulation (Heitz & Schall, 2013), there are changes in shapes of drift rate function and non-decision time. Under “accuracy” drift rates have a narrower range and lower rate

of change as brightness level increases than under “speed”. The time taken by non-decision processes is lower under “accuracy” than under “speed” by about 25 ms for “kr”, but is reversed for “nh” and “jf”, if measured in terms of means, and HDIs for the two conditions effectively do not overlap.

In contrast, other parameters show negligible or no effects, of speed-accuracy instruction. Non-decision time standard deviations have well separated HDIs, but the effect of the different is negligible. Weibull scale parameters, bias means and standard deviations show overlapping HDIs. The mean bias estimate, however, is consistent with a slight dark decision bias because HDIs are below 0.5.

Parameter	Lower	Mean	Upper	Subject
$\rho_{\delta,\beta}$	0.037	0.069	0.103	“kr”
$\rho_{\delta,\tau^{nd}}$	-0.097	-0.089	-0.079	“kr”
$\rho_{\beta,\tau^{nd}}$	-0.086	-0.077	-0.067	“kr”
$\rho_{\delta,\beta}$	-0.195	-0.140	-0.061	“jf”
$\rho_{\delta,\tau^{nd}}$	0.119	0.218	0.324	“jf”
$\rho_{\beta,\tau^{nd}}$	0.304	0.396	0.505	“jf”
$\rho_{\delta,\beta}$	0.274	0.307	0.577	“nh”
$\rho_{\delta,\tau^{nd}}$	-0.520	-0.389	-0.278	“nh”
$\rho_{\beta,\tau^{nd}}$	0.062	0.187	0.321	“nh”

Table 4.5: Summary of the posterior distribution of the correlation parameters of the normal copula model for all three subjects. Each parameter is summarized by a mean and lower/upper boundaries of its HDI. Note: ACC is accuracy condition and SPD is speed condition.

The central result of study B is finding small to moderate values of the correlation parameters, shown in Table 4.5. The pattern of correlations varies across subjects as judged by HDI estimates. “kr” shows small correlation, but “jf” and “nh” show moderate correlations. The probabilities of dependent relations are all close to 1.00

for all subjects. However, there is no consistent pattern across subjects. Estimates of the posterior densities of the correlation parameters are shown in Figure 4.15 for subject “kr”, with other subjects showing similar unimodal and approximately normal shapes, consistent with traceplots.

4.4.7 Discussion

Before discussing the primary result of study B - finding evidence for parameter correlations in the benchmark dataset - a few general conclusions can be drawn from the estimated parameters and the used sampler. The choices of marginal distributions in the two Bayesian models were of different statistical families than has been used previously with the diffusion model. The standard Ratcliff diffusion model assigns a continuous uniform distribution for both starting point and non-decision time constant while Vandekerckhove et al. (2011) used uniform for decision bias and normal for non-decision time constant in their analyses of the benchmark dataset. In my application, I used a beta distribution for the decision bias and gamma distribution for the non-decision time constant. While different families have been used, estimates across studies are very similar. The similarity is in agreement with the study by Ratcliff showing that a considerable latitude in distributions is possible with predictions of the various models being very similar (Ratcliff, 2013).

The latitude in distributional assumptions may suggest that various diffusion models are unfalsifiable, and hence useless for interpreting data (Heathcote, Wagenmakers, & Brown, 2014; Jones & Dzhafarov, 2014a, 2014b; Smith, Ratcliff, & McKoon, 2014). However, the latitude is limited Ratcliff (2013), and the models can only fit patterns

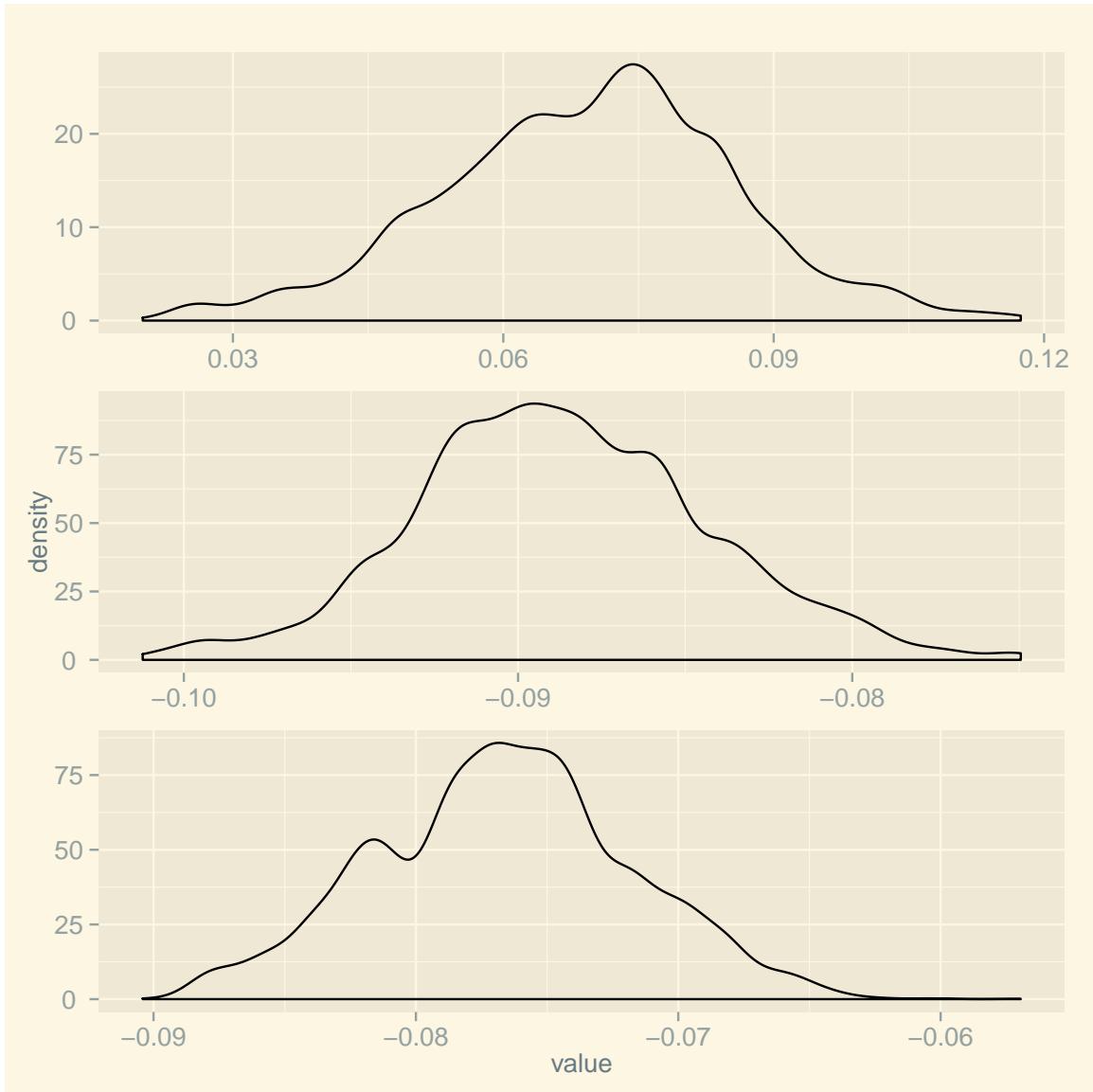


Figure 4.15: Estimated probability density functions for $(\rho_{\delta,\beta}, \rho_{\delta,t^{nd}}, \rho_{\beta,t^{nd}})$ based on the MCMC samples after the burn-in period. The posterior shapes are unimodal and approximately normal. The mean values give evidence for little or no correlations between parameters of the normal diffusion model.

of data similar to the observed data (Ratcliff, 2002). The value of the latitude is allowing two interpretable parameterizations, in terms of starting point or decision bias, which increases the chances of finding distributional assumption that give a better understanding of the across-trial variation in decision and non-decision parameters.

Another conclusion stemming from the parameter estimates is that correlation parameters can be estimated separately from non-correlation parameters. This conclusions is based on similarity of parameters recovered from the independent and normal copula models. This means that correlation parameters correspond to an independent source of variation in the data, as was also suggested by Study A. It suggests that correlations may be targeted experimentally while keeping stimulus quality and instructions fixed. The effects will interact with other independent variables, as shown in Study A, but the benefit of manipulating correlations would be new behavioral patterns against which sequential sampling models have not been tested, yet.

The estimates also show that speed-accuracy instructions effect multiple parameters of the models. This provides further support that speed-accuracy manipulation has complex effects that extend beyond shifting participants' caution, modeled with changes in evidence threshold (Heitz, 2014). As shown in the previous analyses of the (Vandekerckhove et al., 2011) data, drift rates change strongly depending on the instruction, as shown in Table 4.4, too. However, a simplifying assumption about the non-decision time means was made by Vandekerckhove et al. (2011), in contradiction to EEG data obtained by Rinkenauer et al. (2004) that shows strong effects of instructions on the motor activity in the neocortex and single cell recording data that shows widespread effect of instructions (Heitz & Schall, 2013). In agreement with

physiological evidence, the fitted model shows a 25 ms effect of instruction on the non-decision parameter, supporting the claim that instruction manipulation leads to complex effects on processing.

The primary finding in the third study is existence of small to moderate correlations between the decision process parameters. Figure 4.15 shows their posterior densities for subject “kr” that had the smallest correlations. A priori, their pattern or values were not obvious, but large datasets for the three subjects in Ratcliff and Rouder (1998) experiment support their existence under standard experimental settings. However, the lack of consistency in pattern suggests several post-hoc interpretations of what the correlation matrices may imply.

One possible explanation for the inconsistencies in magnitudes and the pattern is that insufficient number of MCMC draws has been collected. For each subject less than 30 thousand draws has been obtained using AMH to estimate the posteriors. The estimate of the covariance matrix needed for AMH proposals contains 190 free parameters, so it may require a much larger number of iterations to obtain accurate estimates. It is possible that the apparent stability of chains only shows a temporary plateau and that chains will move towards a higher density region with more iterations that will cause an agreement in the correlation estimates across subjects. The convergence diagnostics are only necessary, but not sufficient conditions for declaring convergence. This raises issues of developing or importing methods for Bayesian analysis that are applicable to models in psychology and that provide approximate, but definite answers, as for example the method of variational Bayes (MacKay, 2003).

Variational Bayes is based on finding an exact expression for an approximation to the posterior. This contrasts with MCMC methods that find an approximation

to the exact posterior. The advantages of variational Bayes are the speed of calculations and no requirement of necessary, but insufficient convergence checks. The approximations are guaranteed to be close. However, the difficulty with the method relative to MCMC is high level of mathematical manipulation skill required to obtain an approximation of the posterior. This is further complicated by the need for a separate derivation for each model, and hence can become a major part of a research project. At the same time, successful use of variation Bayes motivates development of standard mathematical models for which standard solutions can be obtained and reused across datasets.

If convergence did occur, then the lack of agreement may be just another manifestation of individual differences. The commonality between participants is existence of dependencies among processing parameters. However, their exact pattern of relations may reflect the different strategies or interpretations of the task, consistent with no pattern of parameter relations common to all subjects. The lack of agreement highlights once again the importance of not aggregating data across participants because the underlying processing architecture that enables them to do the task may differ (S. Brown & Heathcote, 2003; McClelland, 1979; Newall, 1973; Schwarz, 2003; Townsend & Ashby, 1983).

The primary source of such qualitative individual differences may be the control processes, such as task instruction processing, task representation, error monitoring, selective attention, that organize the processes, and their relations, required for task completion. Studies of control processes show that their manipulation can result in rapid (on the order of seconds) modulation of task-evoked brain network in line with task demands and statistical structure of stimuli (Collins & Frank, 2013; Jones,

Kinoshita, & Mozer, 2009; Mozer, Kinoshita, & Shettel, 2007). The neural basis of control processes has been generally assigned to prefrontal cortex and basal ganglia (Van Schouwenburg, Aarts, & Cools, 2010), regions which have two-way projections to almost all the other structures in the limbic system and neocortex (Calabresi, Picconi, Tozzi, Ghiglieri, & Di Filippo, 2014; Miller & Cohen, 2001). Hence, current understanding of neural architecture of the control processes provides an opportunity for simultaneous modulation of activity in all components of a task-evoked network supporting particular speeded decisions. In other words, the pattern of parameter dependencies reflects the simultaneous modulation induced by the control processes whose operation varies across individuals. In this formulation, subjects once again support the view of highly flexible capacity of humans for information processing.

4.5 Summary

The two studies reported in this thesis show that using theory of copulas can be a fruitful way of adding a correlation structure to the trial-varying parameters of sequential sampling or other models to explore implications of data more closely. Not only are there a wide selection of standard copulas (Joe, 1997; Nelsen, 2006), but there are also general methods for developing new copulas by considering two variables at a time (Czado, 2010). With a copula at hand, we can define a novel joint distribution of parameters that characterizes across-trial variation in a psychological state of participants while keeping familiar, interpretable scales and assigning flexible distributions to each of the parameters.

Exploration of predictions and data fitting of diffusion models with correlated parameters yielded several conclusions. One conclusion, based on comparing dependent

and independent parameter models, is that correlations may be important for data fitting and providing novel explanations for experimental effects. The conclusion is based on large discrepancies that correlations can produce relative to predictions of the independent copula model while other parameters are matched.

Considering predictions for the mean sample paths during a response signal task, correlations, even at high magnitudes, produce small effects shapes of the paths, medium effects on initial decision bias, and large effects on absorption times. In line with large effects on absorption times, correlated parameters can produce strong effects on reaction time distributions, overall impacting the asymmetry of the two conditional distributions, capable of enhancing or reversing it. At the same time, there is little effect on choice probabilities. Strong effects of correlation parameters, especially on reaction times, indicate that they may be useful for fitting and explaining variation in behavioral data not possible with the standard Ratcliff model.

The primary conclusion of study B is that data collected by Ratcliff and Rouder (1998), given the normal copula diffusion model, implies non-zero values for the three correlations. The correlations were obtained with other parameters shifting marginally relative to the fit of the independent copula model. Stability in parameters across models suggests that adding correlations to the diffusion models may provide new information about cognitive processes from behavioral data.

However, the patterns of correlations differed in magnitudes and signs across the three participants. One possible explanation is that convergence diagnostics failed and that chains only appear to have converged. If the obtained chains have not converged, then further simulation may bring out a uniform pattern across subjects.

Alternatively, the difference in correlation patterns is another expression of individual differences. The variation could arise from control processes, like task representation or pre-activation of muscles, that organize the basic processes and their relations to enable the participant to complete the task. The individual differences are a sign of a flexible information processing system that the human brain is.

The conclusions obtained in Study A are robust because they were based on controlled simulations. However, the conclusions of Study B are tentative. Assuming the computational resources are available, the lack of convergence can be easily checked by running longer chains. However, the risk of lack of convergence would persist as an integral part of the MCMC method for obtaining posterior densities. This raises issues of developing methods for Bayesian analysis that provide approximate, but definite answers, as for example the method of variational Bayes (MacKay, 2003).

Another way to clarify the issue of discordant correlation patterns is to analyze more datasets and conduct experiments. There are other large datasets of behavioral data obtained from a two-choice task that can be used to obtain precise estimates of the correlation matrix to establish whether subjects do show consistent correlation patterns without explicit experimental manipulation (Dutilh et al., 2013, 2012, 2009). In addition, an experiment where relations between parameters are manipulated can show whether all subjects can be brought to show various consistent patterns. One possibility is to set initial decision bias with an instruction on a trial level while matching or mismatching the induced bias with stimulus quality. This way one can set positive or negative relations between rate of information accumulation and initial bias. Such an experiment is being currently developed by the author.

References

- Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036–1060.
- Ashby, F. G., & Helie, S. (2011). A tutorial on computational cognitive neuroscience: Modeling the neurodynamics of cognition. *Journal of Mathematical Psychology*, 55(4), 273–289.
- Berger, J. O. (1997). *Statistical decision theory and Bayesian analysis* (2nd ed.). New York, NY: Springer-Verlag.
- Berkes, P., Wood, F., & Pillow, J. W. (2009). Characterizing neural dependencies with copula models. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems 21* (pp. 129–136). Red Hook, NY: Curran Associates, Inc.
- Berntsen, J., Espelid, T., & Genz, A. (1991). An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software*, 17(4), 437–451.
- Bogacz, R. (2007). Optimal decision-making theories: Linking neurobiology with behavior. *Trends in Cognitive Sciences*, 11(3), 118–125.
- Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, 113(4), 700–765.
- Boring, E. G. (1937). A psychological function is the relation of successive differentiations of events in the organism. *Psychological Review*, 44(6), 445–461.
- Braeken, J., Kuppens, P., De Boeck, P., & Tuerlinckx, F. (2013). Contextualized personality questionnaires: A case for copulas in structural equation models for categorical data. *Multivariate Behavioral Research*, 48(6), 845–870.
- Braeken, J., Tuerlinckx, F., & De Boeck, P. (2007). Copula functions for residual dependency. *Psychometrika*, 72(3), 393–411.
- Brown, J. W. (2014). The tale of the neuroscientists and the computer: Why mechanistic theory matters. *Frontiers in Neuroscience*, 8. doi: 10.3389/fnins.2014.00349
- Brown, S., & Heathcote, A. (2003). Averaging learning curves across and within participants. *Behavior Research Methods, Instruments, & Computers*, 35(1), 11–21.

- Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, 57(3), 153–178.
- Brown, S. D., Marley, A., Donkin, C., & Heathcote, A. (2008). An integrated model of choices and response times in absolute identification. *Psychological Review*, 115(2), 396–425.
- Busemeyer, J. R., & Diederich, A. (2010). *Cognitive modeling*. New York, NY: Sage.
- Calabresi, P., Picconi, B., Tozzi, A., Ghiglieri, V., & Di Filippo, M. (2014). Direct and indirect pathways of basal ganglia: A critical reappraisal. *Nature Neuroscience*, 17(8), 1022–1030.
- Camerer, C. (2003). *Behavioral game theory: Experiments in strategic interaction*. Princeton, NJ: Princeton University Press.
- Casella, G., & Berger, R. L. (2002). *Statistical inference* (2nd ed.). Pacific Grove, CA: Duxbury Press.
- Cassey, P., Heathcote, A., & Brown, S. D. (2014). Brain and behavior in decision-making. *PLoS Computational Biology*, 10(7). doi: 10.1371/journal.pcbi.1003700
- Cavagnaro, D. R., Myung, J. I., & Pitt, M. A. (2013). Mathematical modeling. In T. Little (Ed.), *The Oxford handbook of quantitative methods* (Vol. 1, pp. 438–453). New York, NY: Oxford University Press.
- Cho, R. Y., Nystrom, L. E., Brown, E. T., Jones, A. D., Braver, T. S., Holmes, P. J., & Cohen, J. D. (2002). Mechanisms underlying dependencies of performance on stimulus history in a two-alternative forced-choice task. *Cognitive, Affective, & Behavioral Neuroscience*, 2(4), 283–299.
- Cole, M. W., Bassett, D. S., Power, J. D., Braver, T. S., & Petersen, S. E. (2014). Intrinsic and task-evoked network architectures of the human brain. *Neuron*, 83(1), 238–251.
- Collins, A. G. E., & Frank, M. J. (2013). Cognitive control over learning: Creating, clustering, and generalizing task-set structure. *Psychological Review*, 120(1), 190–229.
- Coombs, C. H. (1983). *Psychology and mathematics: An essay on theory*. Ann Arbor, MI: University of Michigan Press.
- Craigmile, P. F., Peruggia, M., & Van Zandt, T. (2010). Hierarchical Bayes models for response time data. *Psychometrika*, 75(4), 613–632.
- Czado, C. (2010). Pair-copula constructions of multivariate copulas. In P. Jaworski, F. Durante, W. K. Hardle, & T. Rychlik (Eds.), *Copula theory and its applications* (pp. 93–109). Berlin, Germany: Springer-Verlag.
- Donkin, C., Brown, S., Heathcote, A., & Wagenmakers, E. (2011). Diffusion versus linear ballistic accumulation: Different models for response time, same conclusions about psychological mechanisms? *Psychonomic Bulletin & Review*, 18(1), 61–69.
- Donkin, C., Brown, S. D., & Heathcote, A. (2009). The overconstraint of response time models: Rethinking the scaling problem. *Psychonomic Bulletin & Review*, 16(6), 1129–1135.

- Dutilh, G., Forstmann, B. U., Vandekerckhove, J., & Wagenmakers, E.-J. (2013). A diffusion model account of age differences in posterror slowing. *Psychology and Aging*, 28(1), 64–76.
- Dutilh, G., Vandekerckhove, J., Forstmann, B. U., Keuleers, E., Brysbaert, M., & Wagenmakers, E.-J. (2012). Testing theories of post-error slowing. *Attention, Perception, & Psychophysics*, 74(2), 454–465.
- Dutilh, G., Vandekerckhove, J., Tuerlinckx, F., & Wagenmakers, E.-J. (2009). A diffusion model decomposition of the practice effect. *Psychonomic Bulletin & Review*, 16(6), 1026–1036.
- Edwards, W. (1965). Optimal strategies for seeking information: Models for statistics, choice reaction times, and human information processing. *Journal of Mathematical Psychology*, 2(2), 312–329.
- Edwards, W., Lindman, H., & Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychological Review*, 70(3), 193–242.
- Forstmann, B. U., & Wagenmakers, E.-J. (Eds.). (2015). *An introduction to model-based cognitive neuroscience*. New York, NY: Springer-Verlag.
- Forstmann, B. U., Wagenmakers, E.-J., Eichele, T., Brown, S., & Serences, J. T. (2011). Reciprocal relations between cognitive neuroscience and formal cognitive models: Opposites attract? *Trends in Cognitive Sciences*, 15(6), 272–279.
- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference* (2nd ed.). Boca Raton, FL: Chapman and Hall/CRC.
- Gao, J., Wong-Lin, K., Holmes, P., Simen, P., & Cohen, J. D. (2009). Sequential effects in two-choice reaction time tasks: Decomposition and synthesis of mechanisms. *Neural Computation*, 21(9), 2407–2436.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian data analysis* (3rd ed.). Boca Raton, FL: Chapman and Hall/CRC.
- Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4), 457–472.
- Genest, C., & MacKay, J. (1986). The joy of copulas: Bivariate distributions with uniform marginals. *The American Statistician*, 40(4), 280–283.
- Genz, A. C., & Malik, A. (1980). Remarks on algorithm 006: An adaptive algorithm for numerical integration over an N-dimensional rectangular region. *Journal of Computational and Applied Mathematics*, 6(4), 295–302.
- Geweke, J. (1991). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In J. M. Bernardo, J. O. Berger, A. P. Dawid, & A. F. M. Smith (Eds.), *Bayesian statistics 4* (pp. 169–194). Oxford, UK: Oxford University Press.
- Givens, G. H., & Hoeting, J. A. (2013). *Computational statistics*. Hoboken, NJ: John Wiley & Sons, Inc.
- Gold, J. I., & Shadlen, M. N. (2007). The neural basis of decision making. *Annual Review of Neuroscience*, 30, 535–574.

- Goldfarb, S., Wong-Lin, K., Schwemmer, M., Leonard, N. E., & Holmes, P. (2012). Can post-error dynamics explain sequential reaction time patterns? *Frontiers in Psychology*, 3. doi: 10.3389/fpsyg.2012.00213
- Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2), 223–242.
- Hare, T. A., Schultz, W., Camerer, C. F., O'Doherty, J. P., & Rangel, A. (2011). Transformation of stimulus value signals into motor commands during simple choice. *Proceedings of the National Academy of Sciences*, 108(44), 18120–18125.
- Heathcote, A., & Hayes, B. (2012). Diffusion versus linear ballistic accumulation: Different models for response time with different conclusions about psychological mechanisms? *Canadian Journal of Experimental Psychology*, 66(2), 125–136.
- Heathcote, A., Wagenmakers, E.-J., & Brown, S. D. (2014). The falsifiability of actual decision-making models. *Psychological Review*, 121(4), 676–676.
- Hebb, D. O. (1980). *Essay on mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Heitz, R. P. (2014). The speed-accuracy tradeoff: History, physiology, methodology, and behavior. *Frontiers in Neuroscience*, 8, 150. doi: 10.3389/fnins.2014.00150
- Heitz, R. P., & Schall, J. D. (2013). Neural chronometry and coherency across speed-accuracy demands reveal lack of homomorphism between computational and neural mechanisms of evidence accumulation. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 368(1628). doi: 10.1098/rstb.2013.0071
- Hofert, M., Kojadinovic, I., Maechler, M., & Yan, J. (2013). Copula: Multivariate dependence with copulas [Computer software manual]. Retrieved from <http://cran.fyxm.net/web/packages/copula/> (R package version 0.999-13)
- Hyndman, R. J., & Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician*, 50(4), 361–365.
- Joe, H. (1997). *Multivariate models and multivariate dependence concepts*. London, UK: Chapman and Hall/CRC.
- Johnson, N. L., Kotz, S., & Balakrishnan, N. (1994). *Continuous univariate distributions* (Vol. 1). New York, NY: John Wiley and Sons, Inc.
- Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions* (Vol. 2). New York, NY: John Wiley and Sons.
- Johnson, S. G., & Narasimhan, B. (2013). Cubature: Adaptive multivariate integration over hypercubes [Computer software manual]. Retrieved from <http://CRAN.R-project.org/package=cubature> (R package version 1.1-2)
- Jones, M., Curran, T., Mozer, M. C., & Wilder, M. H. (2013). Sequential effects in response time reveal learning mechanisms and event representations. *Psychological Review*, 120(3), 628–666.
- Jones, M., & Dzhafarov, E. N. (2014a). Analyzability, ad hoc restrictions, and excessive flexibility of evidence-accumulation models: reply to two critical commentaries. *Psychological Review*, 121(4), 689–695.

- Jones, M., & Dzhafarov, E. N. (2014b). Unfalsifiability and mutual translatability of major modeling schemes for choice reaction time. *Psychological Review*, 121(1), 1–32.
- Jones, M., Kinoshita, S., & Mozer, M. C. (2009). Optimal response initiation: Why recent experience matters. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems 21* (pp. 785–792). Red Hook, NY: Curran Associates, Inc.
- Karlin, S., & Taylor, H. M. (1975). *A first course in stochastic processes* (2nd ed.). New York, NY: Academic Press.
- Karlin, S., & Taylor, H. M. (1981). *A second course in stochastic processes* (2nd ed.). New York, NY: Academic Press.
- Kim, W., Pitt, M. A., Lu, Z.-L., Steyvers, M., & Myung, J. I. (2014). A hierarchical adaptive approach to optimal experimental design. *Neural Computation*, 26(11), 2465–2492.
- Kotz, S., Balakrishnan, N., & Johnson, N. L. (2000). *Continuous multivariate distributions, models and applications* (2nd ed., Vol. 1). New York, NY: John Wiley and Sons, Inc.
- Kruschke, J. K. (2010). What to believe: Bayesian methods for data analysis. *Trends in Cognitive Sciences*, 14(7), 293–300.
- Laming, D. R. J. (1968). *Information theory of choice-reaction times*. London, UK: Academic Press.
- Lee, M. D., & Wagenmakers, E.-J. (2013). *Bayesian cognitive modeling: A practical course*. New York, NY: Cambridge University Press.
- Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9), 1989–2001.
- Lewandowsky, S., & Farrell, S. (2011). *Computational modeling in cognition: Principles and practice*. New York, NY: Sage.
- Link, S. W., & Heath, R. A. (1975). A sequential theory of psychological discrimination. *Psychometrika*, 40(1), 77–105.
- Ljung, L. (1999). *System identification: Theory for the user*. Englewood, NJ: Prentice Hall.
- Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, 34(1), 1–12.
- Luce, R. D. (1986). *Response times: Their role in inferring elementary mental organization*. New York, NY: Oxford University Press.
- MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge, UK: Cambridge University Press.
- Macmillan, N. A., & Creelman, C. D. (2004). *Detection theory: A user's guide* (2nd ed.). New York, NY: Cambridge University Press.
- Matzke, D., & Wagenmakers, E.-J. (2009). Psychological interpretation of the ex-Gaussian and shifted Wald parameters: A diffusion model analysis. *Psychonomic Bulletin & Review*, 16(5), 798–817.

- McClelland, J. L. (1979). On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review*, 86(4), 287–330.
- Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24, 167–202.
- Mozer, M. C., Kinoshita, S., & Shettel, M. (2007). Sequential dependencies in human behavior offer insights into cognitive control. In W. D. Gray (Ed.), *Integrated models of cognitive systems* (pp. 180–193). Oxford, UK: Oxford University Press.
- Mulder, M. J., van Maanen, L., & Forstmann, B. U. (2014). Perceptual decision neurosciences - a model-based review. *Neuroscience*, 277, 872884.
- Myung, I. J., & Pitt, M. A. (1997). Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin & Review*, 4(1), 79–95.
- Myung, J. I., Karabatsos, G., & Iverson, G. J. (2008). A statistician's view on Bayesian evaluation of informative hypotheses. In H. Hoijtink, I. Klugkist, & P. A. Boelen (Eds.), *Bayesian evaluation of informative hypotheses* (pp. 309–327). New York, NY: Springer.
- Nelsen, R. B. (2006). *An introduction to copulas* (2nd ed.). New York, NY: Springer.
- Newall, A. (1973). You can't play 20 questions with nature and win. In W. G. Chase (Ed.), *Visual information processing* (pp. 1–26). New York, NY: Academic Press.
- Peruggia, M., Van Zandt, T., & Chen, M. (2002). Was it a car or a cat I saw? An analysis of response times for word recognition. In C. Gatsonis et al. (Eds.), *Case studies in Bayesian statistics* (pp. 319–334). New York, NY: Springer-Verlag.
- Peters, E., Hart, P. S., Tusler, M., & Fraenkel, L. (2014). Numbers matter to informed patient choices: A randomized design across age and numeracy levels. *Medical Decision Making*, 34(4), 430–442.
- Pike, R. (1973). Response latency models for signal detection. *Psychological Review*, 80(1), 53–68.
- Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1), 7–11. Retrieved from <http://CRAN.R-project.org/doc/Rnews/>
- Polanía, R., Krajbich, I., Grueschow, M., & Ruff, C. C. (2014). Neural oscillations and synchronization differentially support evidence accumulation in perceptual and value-based decision making. *Neuron*, 82(3), 709–720.
- Purcell, B. A., Heitz, R. P., Cohen, J. Y., Schall, J. D., Logan, G. D., & Palmeri, T. J. (2010). Neurally constrained modeling of perceptual decision making. *Psychological Review*, 117(4), 1113–1143.
- R Core Team. (2014). R: a language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <http://www.R-project.org/>
- Rabbitt, P. (1979). How old and young subjects monitor and control responses for accuracy and speed. *British Journal of Psychology*, 70(2), 305–311.

- Rae, B., Heathcote, A., Donkin, C., Averell, L., & Brown, S. (2014). The hare and the tortoise: Emphasizing speed can change the evidence used to make decisions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 40(5), 1226–1243.
- Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85(2), 59–108.
- Ratcliff, R. (1993). Methods for dealing with reaction time outliers. *Psychological Bulletin*, 114(3), 510–532.
- Ratcliff, R. (2002). A diffusion model account of response time and accuracy in a brightness discrimination task: Fitting real data and failing to fit fake but plausible data. *Psychonomic Bulletin & Review*, 9(2), 278–291.
- Ratcliff, R. (2013). Parameter variability and distributional assumptions in the diffusion model. *Psychological Review*, 120(1), 281–292.
- Ratcliff, R. (2014). Measuring psychometric functions with the diffusion model. *Journal of Experimental Psychology: Human Perception and Performance*, 40(2), 870–888.
- Ratcliff, R., Cherian, A., & Segraves, M. (2003). A comparison of macaque behavior and superior colliculus neuronal activity to predictions from models of simple two-choice decisions. *Journal of Neurophysiology*, 90, 1392–1407.
- Ratcliff, R., Hasegawa, Y. T., Hasegawa, R. P., Childers, R., Smith, P. L., & Segraves, M. A. (2011). Inhibition in superior colliculus neurons in a brightness discrimination task? *Neural Computation*, 23(7), 1790–1820.
- Ratcliff, R., Hasegawa, Y. T., Hasegawa, R. P., Smith, P. L., & Segraves, M. A. (2007). Dual diffusion model for single-cell recording data from the superior colliculus in a brightness-discrimination task. *Journal of Neurophysiology*, 97(2), 1756–1774.
- Ratcliff, R., Love, J., Thompson, C. A., & Opfer, J. E. (2012). Children are not like older adults: A diffusion model analysis of developmental changes in speeded responses. *Child Development*, 83(1), 367–381.
- Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, 20(4), 873–922.
- Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, 9(5), 347–356.
- Ratcliff, R., & Smith, P. L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review*, 111(2), 333–367.
- Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review*, 9(3), 438–481.
- Ratcliff, R., Van Zandt, T., & McKoon, G. (1999). Connectionist and diffusion models of reaction time. *Psychological Review*, 106(2), 261–300.
- Rinkenauer, G., Osman, A., Ulrich, R., Müller-Gethmann, H., & Mattes, S. (2004). On the locus of speed-accuracy trade-off in reaction time: Inferences from the lateralized readiness potential. *Journal of Experimental Psychology: General*,

- 133(2), 261–282.
- Robert, C. P., & Casella, G. (2005). *Monte Carlo statistical methods* (2nd ed.). New York, NY: Springer.
- Roberts, G., & Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2), 349–367.
- Ross, S. M. (2014). *Introduction to probability models*. New York, NY: Academic Press.
- Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin and Review*, 12(4), 573–604.
- Rouder, J. N., Lu, J., Speckman, P., Sun, D., & Jiang, Y. (2005). A hierarchical model for estimating response time distributions. *Psychonomic Bulletin & Review*, 12(2), 195–223.
- Rouder, J. N., Morey, R. D., & Pratte, M. S. (2014). Bayesian hierarchical models. In W. H. Batchelder, H. Colonius, E. Dzhafarov, & J. I. Myung (Eds.), *The new handbook of mathematical psychology* (Vol. 1). London, UK: Oxford University Press.
- Schwarz, W. (2003). Stochastic cascade processes as a model of multi-stage concurrent information processing. *Acta Psychologica*, 113(3), 231–261.
- Shadlen, M. N., & Kiani, R. (2013). Decision making as a window on cognition. *Neuron*, 80(3), 791–806.
- Shadlen, M. N., & Newsome, W. T. (1996). Motion perception: Seeing and deciding. *Proceedings of the National Academy of Sciences*, 93(2), 628–633.
- Shadlen, M. N., & Newsome, W. T. (1998). The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *The Journal of Neuroscience*, 18(10), 3870–3896.
- Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8, 229–231.
- Smith, P. L. (1995). Psychophysically principled models of visual simple reaction time. *Psychological Review*, 102(3), 567–593.
- Smith, P. L. (2000). Stochastic dynamic models of response time and accuracy: A foundational primer. *Journal of Mathematical Psychology*, 44(3), 408–463.
- Smith, P. L., & Ratcliff, R. (2004). Psychology and neurobiology of simple decisions. *Trends in Neurosciences*, 27(3), 161–168.
- Smith, P. L., & Ratcliff, R. (2009). An integrated theory of attention and decision making in visual signal detection. *Psychological Review*, 116(2), 283–317.
- Smith, P. L., Ratcliff, R., & McKoon, G. (2014). The diffusion model is not a deterministic growth model: Comment on Jones and Dzhafarov (2014). *Psychological Review*, 121(4), 679–688.
- Stone, M. (1960). Models for choice-reaction time. *Psychometrika*, 25(3), 251–260.
- Swensson, R. G. (1972). The elusive tradeoff: Speed vs accuracy in visual discrimination tasks. *Perception & Psychophysics*, 12(1), 16–32.

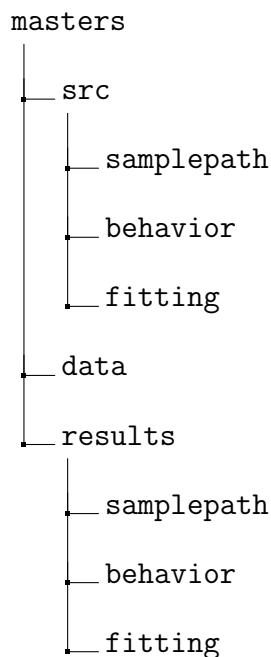
- Teodorescu, A. R., & Usher, M. (2013). Disentangling decision models: From independence to competition. *Psychological Review*, 120(1), 1–38.
- Townsend, J. T., & Ashby, F. G. (1983). *The stochastic modeling of elementary psychological processes*. New York, NY: Cambridge University Press.
- Tuerlinckx, F. (2004). The efficient computation of the cumulative distribution and probability density functions in the diffusion model. *Behavior Research Methods, Instruments, & Computers*, 36(4), 702–716.
- Tuerlinckx, F., Maris, E., Ratcliff, R., & De Boeck, P. (2001). A comparison of four methods for simulating the diffusion process. *Behavior Research Methods, Instruments, & Computers*, 33(4), 443–456.
- Turner, B. M., Forstmann, B. U., Wagenmakers, E.-J., Brown, S. D., Sederberg, P. B., & Steyvers, M. (2013). A Bayesian framework for simultaneously modeling neural and behavioral data. *NeuroImage*, 72, 193–206.
- Turner, B. M., Sederberg, P. B., Brown, S. D., & Steyvers, M. (2013). A method for efficiently sampling from distributions with correlated dimensions. *Psychological Methods*, 18(3), 368–384.
- Tversky, A., & Kahneman, D. (1981). The framing of decisions and the psychology of choice. *Science*, 211(4481), 453–458.
- Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review*, 108(3), 550–592.
- Vandekerckhove, J., & Tuerlinckx, F. (2007). Fitting the Ratcliff diffusion model to experimental data. *Psychonomic Bulletin & Review*, 14(6), 1011–1026.
- Vandekerckhove, J., & Tuerlinckx, F. (2008). Diffusion model analysis with MATLAB: A DMAT primer. *Behavior Research Methods*, 40(1), 61–72.
- Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response times. *Psychological Methods*, 16(1), 44–62.
- Van Schouwenburg, M., Aarts, E., & Cools, R. (2010). Dopaminergic modulation of cognitive control: Distinct roles for the prefrontal cortex and the basal ganglia. *Current Pharmaceutical Design*, 16(18), 2026–2032.
- Van Zandt, T., Colonius, H., & Proctor, R. W. (2000). A comparison of two response time models applied to perceptual matching. *Psychonomic Bulletin & Review*, 7(2), 208–256.
- Van Zandt, T., & Townsend, J. T. (2013). Designs for and analyses of response time experiments. In T. Little (Ed.), *The Oxford handbook of quantitative methods* (Vol. 1, pp. 260–285). New York, NY: Oxford University Press.
- Verdonck, S., & Tuerlinckx, F. (2014). The Ising Decision Maker: a binary stochastic network for choice response time. *Psychological Review*, 121(3), 422–462.
- Vickers, D. (1979). *Decision processes in visual perception*. New York, NY: Academic Press.
- Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, 32(7), 1206–1220.
- Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model

- analysis. *Behavior Research Methods*, 39(4), 767–775.
- Wabersich, D. (2014). Wiener process distribution functions [Computer software manual]. Retrieved from <https://sourceforge.net/projects/rwiener/> (R package version 1.2-0)
- Wagenmakers, E.-J. (2009). Methodological and empirical developments for the Ratcliff diffusion model of response times and accuracy. *European Journal of Cognitive Psychology*, 21(5), 641–671.
- Wagenmakers, E.-J., Farrell, S., & Ratcliff, R. (2004). Estimation and interpretation of $1/f^\alpha$ noise in human cognition. *Psychonomic Bulletin & Review*, 11(4), 579–615.
- Wald, A., & Wolfowitz, J. (1948). Optimum character of the sequential probability ratio test. *The Annals of Mathematical Statistics*, 19(3), 326–339.
- Wang, X.-J. (2008). Decision making in recurrent neuronal circuits. *Neuron*, 60(2), 215–234.
- Yang, Z., & Rodríguez, C. E. (2013). Searching for efficient Markov chain Monte Carlo proposal kernels. *Proceedings of the National Academy of Sciences*, 110(48), 19307–19312.
- Zandbelt, B., Purcell, B. A., Palmeri, T. J., Logan, G. D., & Schall, J. D. (2014). Response times from ensembles of accumulators. *Proceedings of the National Academy of Sciences*, 111(7), 2848–2853.
- Zhang, J., & Rowe, J. B. (2014). Dissociable mechanisms of speed-accuracy tradeoff during visual perceptual learning are revealed by a hierarchical drift-diffusion model. *Frontiers in Neuroscience*, 8. doi: 10.3389/fnins.2014.00069

Appendix A: Study A1 - Mean Sample Paths

The code in Appendix A was used to obtain results for Study A1. All R packages and support functions are listed in Code A.2.

All the computations for simulating sample paths and then creating graphs out of them are organized with two R scripts. Each script allows for various modifications of computation carried out by support functions and then saves all the results in separate files. The loading and saving of files uses relative paths and assumes the following directory structure for the code in this and other appendices:



```

1 # load all the support functions
2 source("src/sample_path/load_dependencies.R")
3
4 # catch errors
5 errors <- file("results/sample_path/errors.Rout", open = "wt")
6 sink(file = errors, append = TRUE, type = "message")
7
8 # Specifies simulation settings
9 settings <- list(models = c("independent", "normal", "t"),
10                   smpl_size = 1e5,
11                   sim_size = 1e3,
12                   seeds = c(1316048320, -1572737661, 195896225),
13                   sigma = .1,
14                   time_unit = 1e-3,
15                   cores = 50,
16                   group_n = 72)
17
18 # Simulates paths for the three models
19 timer <- proc.time()
20 paths <- with(data = settings,
21               expr = mapply(FUN = simul_paths,
22                             model = models,
23                             seed = seeds,
24                             MoreArgs = list(smpl_size, sim_size, sigma,
25                                             time_unit, cores, group_n),
26                             SIMPLIFY = FALSE))
27 timer <- proc.time() - timer
28
29 # Saves computational time
30 capture.output(timer,
31                 file = paste0("results/sample_path/
32                               paths-computational-time-",
33                               Sys.Date(), ".txt"))
34 # Saves a compressed nested list of paths in binary format
35 save(paths, file = paste0("results/sample_path/paths-",
36                           Sys.Date(), ".RData"))
37 # Saves machine/software/settings configuration for reproducibility
38 capture.output(sessionInfo(),
39                 file = paste0("results/sample_path/
40                               machine-software-config-",
41                               Sys.Date(), ".txt"))
42 capture.output(settings,
43                 file = paste0("results/sample_path/simulation-config-",
44                               Sys.Date(), ".txt"))

```

Code A.1: Script for simulating sample paths

```

1 # packages
2 library("compiler")
3 library("dplyr")
4 library("magrittr")
5 library("doParallel")
6 library("doRNG")
7
8 # support functions
9 source("src/sample_path/calculate_weibull.R")
10 source("src/sample_path/calculate_path_stats.R")
11 source("src/sample_path/wiener_parameters.R")
12 source("src/sample_path/combine_parameters.R")
13 source("src/sample_path/simulate_parameters.R")
14 source("src/sample_path/simulate_rndwalk.R")
15 source("src/sample_path/simulate_sample_paths.R")

```

Code A.2: Support function that loads all packages and other support functions necessary for simulation

```

1 # weibull psychometric model
2 weibull_param <- data.frame(lower = c(-.352, -.565),
3                             upper = c(.329, .511),
4                             scale = c(.526, .521),
5                             shape = c(4.413, 5.227),
6                             row.names = c("acc", "spd"))
7
8 # brightness covariates
9 bright_spd <- data.frame(prop = c(.512, .527, .547, .570, .605, .740))
10
11 bright_acc <- data.frame(prop = c(.503, .515, .530, .545, .565, .590))
12
13 # drift rate (reflects effect of s/a and brightness)
14 nu <- as.data.frame(t(data.frame(
15   acc = as.vector(weibull(bright_acc,
16                     weibull_param[ "acc", "lower" ],
17                     weibull_param[ "acc", "upper" ],
18                     weibull_param[ "acc", "scale" ],
19                     weibull_param[ "acc", "shape" ])),
20   spd = as.vector(weibull(bright_spd,
21                     weibull_param[ "spd", "lower" ],
22                     weibull_param[ "spd", "upper" ],
23                     weibull_param[ "spd", "scale" ],
24                     weibull_param[ "spd", "shape" ]))))))
25 colnames(nu) <- paste("prop", as.character(1:6), sep = "")
26
27 # wiener process (reflects effect of s/a)
28 wiener <- data.frame(alpha = c(.221, .05), eta = c(.127, .127),
29                       lambda = c(.464, .464), gamma = c(.065, .065),
30                       chi = c(.279, .279), phi = c(.041, .041),
31                       row.names = c("acc", "spd"))
32
33 # copula degrees of freedom and correlations
34 omega <- 3
35 rho <- data.frame(rho_db = c(.15, .5, .85, -.15, -.5,
36                           -.85, .15, .5, .85),
37                     rho_dt = c(-.15, -.5, -.85, -.15, -.5,
38                           -.85, .15, .5, .85),
39                     rho_bt = c(-.15, -.5, -.85, .15, .5,
40                           .85, .15, .5, .85)))

```

Code A.3: Support function that loads parameter values used in simulations

```

1 weibull <- function(bright, lower, upper, scale, shape) {
2   # Inputs: numeric vector bright, numeric scalars lower,
3   #           upper, scale, shape
4   # Output: numeric vector drifts
5
6   drifts <- lower +
7     (upper - lower) *
8     (1 - exp((-bright ^ shape) / (scale ^ shape)))
9   drifts
10 }

```

Code A.4: Support function that implements Weibull function that maps brightness level to drift rate

```

1 extend_paths <- function(path_sample) {
2   # Inputs: list of double vectors path_sample
3   # Output: list of double vectors path_sample
4
5   bounds <- sapply(path_sample, length) %>% quantile(c(.35, .7))
6   path_sample %>% sapply(length) %>% max %>%
7     as.data.frame %>%
8   do({ sapply(X = path_sample,
9     FUN = function(path) {
10       vec <- sort(c(0, bounds[1],
11                     bounds[2]))
12       speed <- findInterval(x = length(path),
13                               vec = vec)
14       alpha <- sort(alpha)
15       last_state <- rev(path)[1]
16       threshold <- findInterval(x = last_state,
17                                   vec = c(-0.3, .001,
18                                         alpha[2] + .1,
19                                         alpha[3] + .1))
20       c(path,
21         rep(x = alpha[threshold],
22               times = as.numeric(.) - length(path)),
23         speed)
24     },
25     simplify = TRUE) %>% as.data.frame})
26   path_sample
27 }

```

Code A.5: Support function that extends all paths with boundary at which they absorbed

```

1 calc_response_path_stats <- function(path_sample) {
2   # Inputs: list of numeric vectors path_sample
3   # Output: list of numeric vectors response_stats
4
5   if (length(path_sample) %in% c(0, 1)) {
6     stop("Need more than one path in a sample")
7   }
8
9   alpha <- c(0, .050, .221)
10  path_sample <- extend_paths(path_sample)
11  response_stats <- list(
12    fast_mean = path_sample[-idx_row, path_sample[idx_row, ] == 1] %>%
13      as.matrix %>%
14      rowMeans(na.rm = TRUE),
15    middle_mean = path_sample[-idx_row, path_sample[idx_row, ] == 2] %>%
16      as.matrix %>%
17      rowMeans(na.rm = TRUE),
18    slow_mean = path_sample[-idx_row, path_sample[idx_row, ] == 3] %>%
19      as.matrix %>%
20      rowMeans(na.rm = TRUE),
21    fast_sd = path_sample[-idx_row, path_sample[idx_row, ] == 1] %>%
22      as.matrix %>%
23      apply(MARGIN = 1, FUN = sd, na.rm = TRUE),
24    middle_sd = path_sample[-idx_row, path_sample[idx_row, ] == 2] %>%
25      as.matrix %>%
26      apply(MARGIN = 1, FUN = sd, na.rm = TRUE),
27    slow_sd = path_sample[-idx_row, path_sample[idx_row, ] == 3] %>%
28      as.matrix %>%
29      apply(MARGIN = 1, FUN = sd, na.rm = TRUE))
30  response_stats <- lapply(response_stats, function(x) x[!is.na(x)])
31  response_stats
32 }

```

Code A.6: Support function that calculates sample path means and standard deviations for each speed and both decisions

```

1 calc_model_paths_stats <- function(path_sample, smpl_size) {
2   # Inputs: list of n double vectors path_sample,
3   #          integer scalar smpl_size
4   # Output: list of lists of 2 double vectors paths_stats
5
6   if (length(path_sample) != smpl_size)
7     return(path_sample)
8   timer <- proc.time()
9   resp_index <- sapply(X = path_sample,
10                         FUN = function(x) {
11                           min(x, na.rm = TRUE) %>%
12                           findInterval(vec = c(-.1, 0),
13                                         rightmost.closed = T) == 1
14                         })
15   timer <- proc.time() - timer
16   cat(paste("calculated", timer["elapsed"], "\n"),
17        file = "results/sample_path/progress.txt",
18        append = TRUE)
19   paths_stats <- list(
20     upper = calc_response_path_stats(path_sample[!resp_index]),
21     lower = calc_response_path_stats(path_sample[resp_index]))
22   paths_stats
23 }
```

Code A.7: Support function that splits paths into lower and upper bound hitting paths and calculates their means and standard deviations

```

1 combine_stats <- function(...) {
2   # Inputs: lists of double vectors
3   # Output: list of lists of numeric vectors stats
4
5   paths <- list(...)
6   cat(paste("combining", "\n"),
7       file = "results/sample_path/progress.txt",
8       append = TRUE)
9   smpl_size <- length(paths[[2]])
10  timer <- proc.time()
11  stats <- mclapply(paths, calc_model_paths_stats,
12                     smpl_size = smpl_size,
13                     mc.cores = 1)
14  timer <- proc.time() - timer
15  cat(paste("combined", timer["elapsed"], "\n"),
16      file = "results/sample_path/progress.txt",
17      append = TRUE)
18  if (length(stats[[1]][[1]]) == 6)
19    return(stats)
20  else
21    stats <- append(stats[[1]], stats[-1])
22  stats
23}

```

Code A.8: Support function that combines path means and standard deviations across conditions

```

1 combine_nu_wiener <- function(nu, wiener) {
2   # Inputs: double vector drifts, double matrix wiener
3   # Output: double matrix nu_wiener
4
5   n_combos <- dim(nu)[2] * 2
6   wiener_index <- rep(c("acc", "spd"), each = n_combos / 2)
7   nu_wiener <- matrix(0, nrow = n_combos, ncol = 7)
8   nu <- c(as.matrix(nu["acc", ]), as.matrix(nu["spd", ]))
9   for (item in seq_len(n_combos)) {
10     nu_wiener[item, ] <- c(nu[item],
11                           as.matrix(wiener[wiener_index[item], ]))
12   }
13   nu_wiener
14 }

```

Code A.9: Support function that combines bounded wiener process parameters into a single matrix

```

1 combine_wiener_copula <- function(nu_wiener, rho, omega) {
2   # Inputs: double matrix nu_wiener, double vector rho,
3   #          double scalar omega
4   # Output: double matrix all_params
5
6   n_corrs <- dim(rho)[1]
7   n_wiener <- dim(nu_wiener)[1]
8   n_combos <- n_wiener * n_corrs
9   all_params <- cbind(matrix(rep(t(nu_wiener), times = n_corrs),
10                         ncol = 7, nrow = n_combos, byrow = TRUE),
11                         as.matrix(rho[rep(seq_len(n_corrs),
12                           each = n_wiener), ])),
13                         rep(omega, times = n_combos))
14   all_params
15 }
```

Code A.10: Support function that combines all wiener process parameters with copula parameters

```

1 combine_param <- function(nu, wiener, rho, omega) {
2   # Purpose: wraps both combine functions and cleans up the output
3   # Inputs: double vector nu, double matrix wiener,
4   #          double vector rho, double scalar omega
5   # Output: double matrix all_params
6
7   nu_wiener <- combine_nu_wiener(nu = nu, wiener = wiener)
8   all_params <- as.data.frame(
9     combine_wiener_copula(nu_wiener = nu_wiener,
10                           rho = rho,
11                           omega = omega))
12   colnames(all_params) <- c("nu", colnames(wiener),
13                           colnames(rho), "omega")
14   rownames(all_params) <- seq_len(dim(all_params)[1])
15   all_params <- all_params[, c(2, 1, seq(3, 11))]
16   all_params
17 }
```

Code A.11: Support function that combines all wiener process parameters with copula parameters

```

1 simul_copula <- function(smpl_size, params, model) {
2   # Inputs: integer scalar smpl_size, double matrix params, string model
3   # Output: double matrix cop_smpl
4   # Note: params order is alpha, nu, eta, lambda, gamma,
5   #        chi, phi, 3 rhos, omega
6
7   if (model == 'independent') {
8     cop_smpl <- matrix(runif(2 * smpl_size), nrow = smpl_size, ncol = 2)
9     return(cop_smpl)
10  }
11  cop_pdf <- ellipCopula(model, param = params[, "rho_db"],
12    dim = 2, dispstr = 'un',
13    df = params[, "omega"])
14  cop_smpl <- rCopula(n = smpl_size, copula = cop_pdf)
15  cop_smpl
16}

```

Code A.12: Support function that draws a sample from a specified copula

```

1 calc_param <- function(cop_smpl, params) {
2   # Inputs: double matrices cop_smpl, params
3   # Output: double matrix param_smpl
4   # Notes: params order - alpha, nu, eta, lambda, gamma,
5   #        chi, phi, 3 rhos, omega
6
7   param_smpl <- matrix(0, nrow = dim(cop_smpl)[1], ncol = 3,
8     dimnames = list(row = seq_len(dim(cop_smpl)[1]),
9       col = c("alpha", "delta", "beta")))
10  param_smpl[, 1] <- as.numeric(params["alpha"])
11  param_smpl[, 2] <- qnorm(cop_smpl[, 1], as.numeric(params["nu"]),
12    as.numeric(params["eta"]))
13  shape1 <- as.numeric(((1 - params["lambda"]) / params["gamma"] ^ 2 -
14    1 / params["lambda"]) * params["lambda"] ^ 2)
15  shape2 <- as.numeric(shape1 * (1 / params["lambda"] - 1))
16  param_smpl[, 3] <- qbeta(cop_smpl[, 2], shape1, shape2)
17  param_smpl
18}

```

Code A.13: Support function that transforms copula draws to parameter draws using inverse cdfs

```

1 smpl_param <- function(params, smpl_size, model) {
2   # Inputs: double matrix params, integer scalar smpl_size, string model
3   # Output: double data.frame param_smpl
4
5   cop_smpl <- simul_copula(smpl_size = smpl_size,
6                             params = params,
7                             model = model)
8   param_smpl <- calc_param(cop_smpl = cop_smpl, params = params)
9   as.data.frame(param_smpl)
10 }

```

Code A.14: Support function that wraps simulation and transformation of copula draws

```

1 rndwalk <- function(smpl_size = 1, delta = .01, sigma = 1, beta = .75,
2                           alpha = 1.3, low_bound = 0, time_unit = 1e-3) {
3   # Inputs: integer scalar smple_size , double scalars delta , sigma ,
4   #           beta , alpha , low_bound , time_unit
5   # Output: double vector path_smpl
6
7   path_smpl <- vector(mode = "list" , length = smpl_size)
8   state_unit <- sigma * sqrt(time_unit)
9   prob_up <- .5 * (1 + delta * sqrt(time_unit) / sigma)
10  if (prob_up > 1) prob_up <- 1
11  if (prob_up < 0) prob_up <- 0
12  for (draw in seq_len(smpl_size)) {
13    path <- vector(mode = "double" , length = 1e4)
14    path[1] <- position <- beta * alpha
15    timer <- 0
16    counter <- 2
17    while (all(position < alpha , position > low_bound)) {
18      if (rbinom(1, 1, prob_up) == 1) {
19        position <- position + state_unit
20        path[counter] <- position
21      }
22      else {
23        position <- position - state_unit
24        path[counter] <- position
25      }
26      timer <- timer + time_unit
27      counter <- counter + 1
28    }
29    path_smpl[[draw]] <- path[path != 0]
30  }
31  path_smpl
32}
33
34 # Compiles just-in-time
35 rndwalk_cmp <- cmpfun(f = rndwalk)
36
37 # Vectorizes with respect to variable parameters
38 rndwalk_vec <- Vectorize(FUN = rndwalk_cmp,
39                           vectorize.args = list("delta" , "beta" , "alpha"))

```

Code A.15: Support function that runs a random walk simulation of a bounded Wiener process

```

1 par_sim <- function(group_n, ind_param, tasks, task_n,
2                      sim_size, smpl_size, model, time_unit) {
3   # Inputs: integer scalar group_n, double matrix ind_param,
4   #          integer vector tasks, scalar integer task_n,
5   #          integer scalars sim_size, smpl_size,
6   #          string model, double scalar time_unit
7   # Output: list of lists of double vectors
8
9   for (i in seq_len(group_n)) {
10     timer <- proc.time()
11     params <- nextElem(ind_param)$value
12     results <- foreach(tasks,
13                           seeds = rng[(params[[1]] - 1) * task_n + tasks],
14                           .combine = "c", .multicombine = TRUE,
15                           .inorder = FALSE,
16                           .maxcombine = task_n) %dopar% {
17       rngtools::setRNG(seeds)
18       trial_param <- smpl_param(params = params,
19                                   smpl_size = sim_size,
20                                   model = model)
21       path_sample <- rndwalk_vec(smpl_size = 1,
22                                    delta = trial_param$delta,
23                                    sigma = sigma, trial_param$beta,
24                                    trial_param$alpha,
25                                    low_bound = 0, time_unit = time_unit)
26       remove(trial_param)
27       return(path_sample)
28   } %>%
29   calc_model_paths_stats(smpl_size) %>%
30   list %>%
31   append(results, .)
32   timer <- proc.time() - timer
33   cat(paste("simulated", timer["elapsed"], "\n"),
34        file = "results/sample-path/progress.txt",
35        append = TRUE)
36 }
37 results
38 }
```

Code A.16: Support function that runs a parallel simulation to obtain sample paths

```

1 simul_paths <- function(model, smpl_size, sim_size, seed = 2132326000,
2                           sigma = .1, time_unit = 1e-3, cores = 1,
3                           group_n = 1) {
4   # Inputs: string model, integer scalars smpl_size, seed,
5   #           double scalars, sigma, time_unit,
6   #           integer scalars cores, group_n
7   # Output: list of lists of numeric vectors results
8
9   name <- paste0("results/sample_path/progress.txt")
10  writeLines(text = "", con = name)
11
12  cond <- seq_len(72)
13  ind_param <- combine_param(nu = nu, wiener = wiener,
14                             rho = rho, omega = omega) %>% slice(cond)
15  alpha <- c(0, rev(unique(ind_param$alpha)))
16
17  task_n <- smpl_size / sim_size
18  tasks <- seq_len(task_n)
19  outer <- nrow(ind_param)
20  inner <- length(tasks)
21  cond_n <- outer / group_n
22  rng <- RNGseq(outer * inner, seed)
23  results <- list()
24  group_tags <- seq_len(group_n) %>% rep(each = cond_n)
25  ind_param <- cbind(ind_param, cond) %>% isplit(group_tags)
26
27  cat(paste("initiating", "\n"),
28       file = "results/sample_path/progress.txt",
29       append = TRUE)
30  registerDoParallel(cores = cores)
31  results <- par_sim(group_n, ind_param, tasks, task_n, params,
32                      sim_size, smpl_size, model, time_unit)
33  results
34}

```

Code A.17: Support function that simulates sample paths in parallel for different copula models

```

1 # Sets the root directory and then uses relative paths
2 source("src/predictions/sample_path/plot_mean_paths.R")
3 load(file = "results/sample_path/paths-2015-06-20.RData")
4
5 # Specifies plotting settings
6 settings <- list(cores = 3,
7                   rows = 6,
8                   cols = 2)
9
10 # Generates nplots .pdf images in the results directory for sample paths
11 # Note: ignore Error: Results... it is an irrelevant feature of ggplot2
12 timer <- proc.time()
13 with(data = settings, expr = plot_mean_paths(paths = paths, cores =
14       cores,
15               rows = rows, cols = cols))
16 timer <- proc.time() - timer
17 # Captures time, settings and data used in a metadata file
18 capture.output(timer["elapsed"],
19                 settings,
20                 print("data = paths-2015-06-20.RData"),
21                 file = paste0("results/sample_path/plots-metadata-",
22                               Sys.Date(), ".txt"))

```

Code A.18: Script for plotting simulated paths

```

1 process_paths <- function(paths) {
2   # Inputs: list of lists of double vectors paths
3   # Output: data_frame model_paths
4
5   library("reshape2")
6   library("ggplot2")
7   library("ggthemes")
8   source("src/sample_path/load_dependencies.R")
9
10  ind_param <- combine_param(nu = nu, wiener = wiener,
11                               rho = rho, omega = omega) %>%
12    slice(1:72) %>%
13    mutate(condition = 1:72)
14  acc_index <- filter(ind_param, alpha == 0.221) %>%
15    dplyr::select(condition)
16
17  alpha <- c(0, rev(unique(ind_param$alpha)))
18  thresholds <- data_frame(upper = rep(unique(ind_param$alpha),
19                           each = 6),
20                           lower = 0,
21                           plots = seq_len(72),
22                           graphs = rep(1:6, each = 12))
23
24  model_paths <- paths %>%
25    melt %>%
26    rename(value = value, sum_stat = L4, response = L3,
27           condition = L2, model = L1) %>%
28    mutate(graph = cut(x = condition, labels = FALSE,
29                      breaks = 6)) %>%
30    group_by(sum_stat, response, condition, model) %>%
31    mutate(time = seq(from = 1, by = 1, length.out = n())) %>%
32    group_by(condition) %>%
33    mutate(instruction = rep(unique(condition) %in%
34                             acc_index$condition + 1,
35                             length(condition))) %>%
36    ungroup
37    write.table(model_paths, "results/sample_path/plot_values.txt",
38                row.names = FALSE)
39}

```

Code A.19: Support function that reshapes simulation output and saves and then returns results

```

1 plot_mean_paths <- function(paths, cores = 1, rows = 6, cols = 2) {
2   # Inputs: list of lists of double vectors paths,
3   #          integer scalars cores, rows, cols
4   # Output: pdf images
5
6   process_paths(paths) %>% group_by(instruction) %>%
7     do({cutoff <- dplyr::select(., instruction) %>% distinct %>%
8         unlist %>% '[ '(c(250, 3000), .)
9         filter(., time < cutoff)
10    }) %>%
11   group_by(graph) %>%
12   do({graph_index <- dplyr::select(., graph) %>%
13       distinct %>%
14       as.numeric
15       plot_order <- group_by(., instruction) %>%
16         dplyr::select(condition) %>%
17         unique
18       plot_order <- ungroup(plot_order) %>%
19         dplyr::select(condition) %>%
20         unlist %>%
21         matrix(nrow = 2, ncol = 6, byrow = TRUE) %>% c
22       reordered <- mutate(., plots = factor(x = .$condition,
23                                         levels = as.character(plot_order)))
24
25   ts.graph <- ggplot() +
26     facet_wrap(facets = ~ plots, nrow = 6, ncol = 2,
27                scales = c("free"), as.table = TRUE) +
28     geom_line(aes(x = time, y = value,
29                  group = interaction(response, plots,
30                  model, sum_stat),
31                  color = model),
32     data = filter(reordered,
33                   sum_stat %in% c("fast_mean",
34                               "middle_mean",
35                               "slow_mean")),
36                   alpha = 1, size = .2) +
37     theme_solarized_2() +
38     ylab("Relative Evidence") +
39     xlab("Time (ms)")
40     ggsave(ts.graph,
41             filename = paste0(
42               "results/sample_path/path-plot-corr",
43               unique(.graph), "-", Sys.Date(), ".pdf"))
44 }
45 }
```

Code A.20: Support function that plots a grid of mean paths of 3 models for nrow * ncol conditions

Appendix B: Study A2 - Response Times and Responses

Appendix B contains R code for sampling and plotting summaries of the three joint distributions of behavioral data that make up Study A2. All R packages and support functions are listed in Code B.2.

The code structure is similar to Appendix A, there is a script for simulations and a script for plotting, each with its own support functions. Code A.3, A.4, A.9, A.10, A.11 was reused in Study B and is not re-listed below.

```

1 # loads required functions
2 source("src/behavior/load_dependencies.R")
3
4 # catches errors
5 errors <- file(paste0("results/behavior/errors-report-", 
6                     Sys.Date(),
7                     ".Rout"),
8                     open = "wt")
9 sink(file = errors, append = TRUE, type = "message")
10
11 # specifies simulation settings
12 settings <- list(models = c("independent", "normal", "t"),
13                     smpl_size = 1e7, sim_size = 1e3,
14                     seeds = c(-1605742457, -1480525591, 1868731723),
15                     sigma = 0.1, cores = 35, group_n = 54)
16
17 # simulates behavior for the three models
18 timer <- proc.time()
19 behavior_sum <- with(data = settings,
20                     expr = mapply(FUN = simul_behavior,
21                                   model = models,
22                                   seed = seeds,
23                                   MoreArgs = list(smpl_size, sim_size,
24                                                   sigma, cores,
25                                                   group_n),
26                                   SIMPLIFY = FALSE)) %>% bind_rows
27 timer <- proc.time() - timer
28
29 # saves a table of behavior summaries in plain text format
30 write.table(x = behavior_sum,
31             file = paste0(
32                 "results/behavior/behavior-summary-", Sys.Date(), ".txt"))
33
34 # saves computational time
35 capture.output(timer,
36                 file = paste0(
37                     "results/behavior/computational-time-",
38                     Sys.Date(), ".txt"))
39
40 # saves machine/software/settings configuration for reproducibility
41 capture.output(sessionInfo(),
42                 file = paste0(
43                     "results/behavior/machine-software-config-",
44                     Sys.Date(), ".txt"))
45 capture.output(settings,
46                 file = paste0(
47                     "results/behavior/simulation-config-",
48                     Sys.Date(), ".txt")))

```

Code B.1: Script for simulating behavioral predictions and summarizing them as input for a plotting script

```

1 # packages
2 library("dplyr")
3 library("magrittr")
4 library("doParallel")
5 library("doRNG")
6 library("copula")
7 library("RWiener")
8 library("ggplot2")
9 library("ggthemes")
10
11 # custom functions
12 source("src/behavior/calculate_weibull.R")
13 source("src/behavior/wiener_parameters.R")
14 source("src/behavior/combine_parameters.R")
15 source("src/behavior/simulate_wiener_parameters.R")
16 source("src/behavior/simulate_rndwalk_rts.R")
17 source("src/behavior/calculate_summary.R")
18 source("src/behavior/simulate_behavior.R")
19 source("src/behavior/plot_behavior.R")

```

Code B.2: Support function that loads packages and other support functions

```

1 simul_copula <- function(smpl_size, params, model) {
2   # Purpose: draws a sample from a model-specified copula
3   # Inputs: integer scalar smpl_size, double vector params,
4   #          string model
5   # Output: numeric matrix cop_smpl
6   # Notes: params order- alpha, nu, eta, lambda, gamma,
7   #          chi, phi, 3 rhos, omega
8
9   if (model == 'independent') {
10     cop_smpl <- matrix(runif(n = 3 * smpl_size), nrow = smpl_size)
11     return(cop_smpl)
12   }
13   cop_pdf <- ellipCopula(family = model,
14                           param =
15                             as.numeric(params[c("rho_db",
16                               "rho_dt",
17                               "rho_bt")]),
18                           dim = 3,
19                           dispstr = 'un',
20                           df = as.numeric(params["omega"]))
21   cop_smpl <- rCopula(n = smpl_size, copula = cop_pdf)
22   cop_smpl
23 }

```

Code B.3: Support function that samples parameters from copulas

```

1 calc_param <- function(cop_smpl, params) {
2   # Inputs: double matrix cop_smpl, double vector params,
3   # Output: double matrix param_smpl
4   # Notes: params order= alpha, nu, eta, lambda, gamma,
5   #          chi, phi, 3 rhos, omega
6
7   param_smpl <- matrix(0, nrow = nrow(cop_smpl), ncol = 4,
8                         dimnames = list(row = seq_len(nrow(cop_smpl)),
9                                           col = c("alpha", "delta",
10                                              "beta", "t_nd")))
11
12  shape1 <- as.numeric(((1 - params["lambda"]) /
13    params["gamma"] ^ 2 -
14    1 / params["lambda"])) *
15    params["lambda"] ^ 2)
16  shape2 <- as.numeric(shape1 * (1 / params["lambda"] - 1))
17  shape <- (as.numeric(params[1, "chi"]) /
18    as.numeric(params[1, "phi"])) ^ 2
19  scale <- as.numeric(params[1, "phi"]) ^ 2 /
20    as.numeric(params[1, "chi"]))
21
22  param_smpl[, "alpha"] <- as.numeric(params["alpha"])
23  param_smpl[, "delta"] <- qnorm(p = cop_smpl[, 1],
24                                    mean = as.numeric(params["nu"]),
25                                    sd = as.numeric(params["eta"]))
26  param_smpl[, "beta"] <- qbeta(p = cop_smpl[, 2],
27                                 shape1 = shape1, shape2 = shape2)
28  param_smpl[, "t_nd"] <- qgamma(p = cop_smpl[, 3],
29                                 shape = shape,
30                                 scale = scale)
31
32  param_smpl
}

```

Code B.4: Support function that transforms copula draws to parameter draws using inverse cdfs and adds the threshold parameter to completely define a random walk

```

1 smpl_param <- function(params, smpl_size, model) {
2   # Inputs: double vector params, integer scalar smpl_size,
3   #          string scalar model
4   # Output: double data_frame param_smpl
5
6   cop_smpl <- simul_copula(smpl_size = smpl_size,
7                             params = params,
8                             model = model)
9   param_smpl <- as.data.frame(calc_param(cop_smpl = cop_smpl,
10                                params = params))
11  param_smpl
12}

```

Code B.5: Support function that wraps simulation and transformation of copula draws to give a parameter sample

```

1 rwiener_num <- function(n, alpha, tau, beta, delta) {
2   # Inputs: integer scalar n, double scalars alpha,
3   #          tau, beta, delta
4   # Output: double matrix rt_r
5
6   rt_r <- rwiener(n, alpha, tau, beta, delta) %>%
7     unlist %>%
8     matrix(nrow = n)
9   rt_r
10 }
11
12 # vectorizes rejection sampling from a Wiener density
13 rwiener_vec <- Vectorize(FUN = rwiener_num,
14                           vectorize.args = c("alpha", "tau",
15                                         "beta", "delta"))

```

Code B.6: Support function that samples and converts dataframe output of rwiener into a matrix to enable vectorization

```

1 smpl_rts <- function(n, alpha, tau, beta, delta, sigma) {
2   # Inputs: integer scalar n, double scalars alpha, tau,
3   #          beta, delta, sigma
4   # Output: numeric data_frame rt_r with row and column names
5
6   rts <- rwiener_vec(n, alpha / sigma, tau, beta, delta / sigma) %>%
7     matrix(nrow = length(alpha), byrow = TRUE) %>%
8     as.data.frame
9   colnames(rts) <- c("rt", "choice")
10  rownames(rts) <- length(alpha) %>% seq_len
11  rts
12}

```

Code B.7: Support function that rescales parameters and cleans up the simulated sample

```

1 sample_par <- function(ind_param, tasks, group_n, cond_n, sim_size,
2                         task_n, outer, sigma, model) {
3   # Inputs: double matrix ind_param, integer vector tasks,
4   #           integer scalars group_n, cond_n, sim_size,
5   #           task_n, outer, double scalar sigma, string model
6   # Output: double matrix results
7
8   results <- matrix(0, ncol = 3, nrow = outer * 10)
9   for (i in seq_len(group_n)) {
10     cat(paste("group simulating", "\n"),
11         file = "results/behavior/progress.txt",
12         append = TRUE)
13     timer <- proc.time()
14     results[((i - 1) * cond_n * 10 + 1):(i * cond_n * 10), ] <- foreach(
15       params = iter(obj = nextElem(ind_param)$value, by = "row"),
16       .multicombine = TRUE, .combine = "combine_quant",
17       .maxcombine = cond_n) %:%
18     foreach(tasks, seeds = rng[(params[[11]] - 1) * task_n + tasks],
19             .combine = "rbind", .multicombine = TRUE,
20             .maxcombine = sim_size, .inorder = FALSE) %dopar% {
21       rngtools::setRNG(seeds)
22       cat(paste("simulating", "\n"),
23           file = "results/behavior/progress.txt",
24           append = TRUE)
25       trial_param <- smpl_param(params = params,
26                                   smpl_size = sim_size,
27                                   model = model)
28       behav_smpl <- smpl_rts(n = 1,
29                               alpha = trial_param$alpha,
30                               tau = trial_param$t_nd,
31                               beta = trial_param$beta,
32                               delta = trial_param$delta,
33                               sigma = sigma)
34       remove(trial_param)
35       behav_smpl
36     }
37   results
38 }
```

Code B.8: Support function that samples behavioral data in parallel

```

1 simul_behavior <- function(model, smpl_size, sim_size,
2                             seed = 1771363045,
3                             sigma = .1, cores = 1, group_n = 1) {
4   # Inputs: string model, integer scalars smpl_size, seed,
5   #           double scalar sigma, integer scalars cores, group_n
6   # Output: numeric data_frame
7
8   name <- paste0("results/behavior/progress.txt")
9   writeLines(text = "", con = name)
10
11  cond <- seq_len(108)
12  task_n <- smpl_size / sim_size
13  tasks <- seq_len(task_n)
14  outer <- length(cond)
15  inner <- task_n
16  cond_n <- outer / group_n
17  rng <- RNGseq(outer * inner, seed)
18  group_tags <- seq_len(group_n) %>% rep(each = cond_n)
19  ind_param <- cbind(combine_param(nu = nu, wiener = wiener,
20                                    rho = rho, omega = omega),
21                      cond) %>% isplit(group_tags)
22
23  cat(paste("initiating", "\n"),
24       file = "results/behavior/progress.txt",
25       append = TRUE)
26  set.seed(seed = seed)
27  registerDoParallel(cores = cores)
28  results <- sample_par(ind_param, tasks, group_n, cond_n, sim_size,
29                         task_n, outer, sigma, model)
30  timer <- proc.time() - timer
31  cat(paste("simulated", timer["elapsed"], "\n"),
32       file = "results/behavior/progress.txt",
33       append = TRUE)
34 }
35 results <- as.data.frame(results)
36 colnames(results) <- c("quant", "prob", "quant_rank")
37 results %>%
38   mutate(instruction = c(
39     "accuracy", "speed") [rep(c(1, 2), each = 60)] %>%
40     rep(times = 9),
41     condition = rep(cond, each = 10),
42     model = model)
43 results
44 }
```

Code B.9: Support function that simulates reaction times and choices in parallel for different copula models

```

1 calc_quant_prob <- function behav_smpl) {
2   # Inputs: numerical data_frame behav_smpl
3   # Output: numerical data_frame behav_sum
4
5   if (ncol(behav_smpl) == 3)
6     return(behav_smpl)
7   probs <- filter(behav_smpl, choice == 1) %>%
8     summarise(., prob = n() / nrow(behav_smpl)) %>%
9     unlist
10  probs <- c(probs, 1 - probs)
11
12  behav_sum <- group_by(behav_smpl, choice) %>%
13    do({behavior <- quantile(x = .$rt,
14      probs = seq(.1, .9, .2),
15      type = 8) %>%
16      as.data.frame})
17
18  choice_n <- attr(behav_sum, "group_sizes")
19  quant <- behav_sum[, 2] %>% unlist
20  quant_rank <- rep(x = seq_len(5), times = 2)
21  prob <- rep(probs, times = choice_n)
22  behav_sum <- data_frame(quant, prob, quant_rank)
23  behav_sum
24 }

```

Code B.10: Support function that calculates point estimates of reaction time quantiles and choice probabilities from a sample of draws from the joint density

```

1 combine_quant <- function (...) {
2   # Inputs: double matrices ...
3   # Output: double matrix behav_sum
4
5   cat(paste("combining", "\n"),
6       file = "results/behavior/progress.txt",
7       append = TRUE)
8   old_res <- list(...)
9   timer <- proc.time()
10  results <- mclapply(old_res, calc_quant_prob,
11    mc.cores = length(old_res))
12  behav_sum <- bind_rows(results) %>% unlist
13  timer <- proc.time() - timer
14  cat(paste("combined", timer["elapsed"], "\n"),
15      file = "results/behavior/progress.txt",
16      append = TRUE)
17  behav_sum
18 }

```

Code B.11: Support function that summarizes behavior into quantiles and probabilities

```

1 # loads necessary functions
2 source("src/predictions/behavior/load_dependencies.R")
3 behavior_sum <-
4   read.table(file = "results/behavior/behavior-summary-2015-06-15.txt",
5             header = TRUE)
6
7
8 # specifies plotting settings
9 settings <- list(rows = 3, cols = 2)
10
11 # generates nplots .pdf images in the results directory for sample paths
12 # ignore Error: Results...
13 with(data = settings,
14       expr = plot_behavior_sum(behavior_sum = behavior_sum,
15                                 rows = rows, cols = cols))

```

Code B.12: Script that plots quantiles-probability graphs

```

1 plot_behavior_sum <- function(behavior_sum, rows = 3, cols = 2) {
2   # Inputs: data_frame behavior_sum, integer scalars rows, cols
3   # Output: pdf images
4
5   behavior_sum %>% mutate(graph = cut(condition,
6                             breaks = 3,
7                             labels = FALSE)) %>%
8     group_by(graph) %>%
9     mutate(subgraph = cut(condition, breaks = 6, labels = FALSE)) %>%
10    do({qp_graph <- ggplot(data = .,
11                           mapping = aes(x = prob, y = quant,
12                                         group = interaction(quant_rank,
13                                                               subgraph,
14                                                               model),
15                                         color = model)) +
16      facet_wrap(facets = ~subgraph, nrow = 3, ncol = 2,
17                  scales = c("free"), as.table = TRUE) +
18      geom_line(alpha = 1, size = .2) +
19      geom_point(size = 1) +
20      theme_solarized_2() +
21      ylab("Reaction Time (sec)") +
22      xlab("Choice Probability")
23
24      ggsave(qp_graph,
25             filename = paste0("results/behavior/qp-plot-corr",
26                               unique(.\$graph), "-",
27                               Sys.Date(), ".pdf"))
28    })
29 }

```

Code B.13: Support function that plots a grid of mean paths of 3 models for nrow * ncol conditions

Appendix C: Study B - Benchmark Dataset Analysis

The code in Appendix C was used to obtain results for Study B. All R packages and support functions are listed in Code C.2.

Chains obtained in Study B were generated using a script and its support functions. Support functions in Code A.4, B.3, B.5, and B.7 were reused in Study B, and are not listed below.

Triple integrals required for elliptical copula models were solved with C code from cubature package, specifically hcubature algorithm (S. G. Johnson & Narasimhan, 2013). C++ code for Wiener density was based on the underlying C code in RWiener package (Verdonck & Tuerlinckx, 2014). The integrator was combined with other support functions written in C++ into several separate files with headers connecting the files. All compiled functions were organized as a R package and imported into R as other packages. To incorporate C++ as a package one can use base functions for creating a package skeleton and R CMD facilities like build and INSTALL to make the code loadable after compilation.

```

1 # loads functions and data
2 source("src/fitting/load_dependencies.R")
3 train_data <- read.table("data/train_data.txt") %>%
4   filter(subj == "jf") %>%
5   dplyr::select(rt, resp, prop, instr) %>%
6   mutate(rt = rt / 1000, prop = prop / 32)
7
8 # catches errors
9 errors <- file("results/fitting/jf-errors.Rout", "wt")
10 sink(errors, TRUE, "message")
11
12 # note: theta_n = 20 (23) for model = independent (normal)
13 model <- "normal"
14 theta_n <- 23
15 initial <- TRUE
16
17 # sampler settings
18 settings <- list(train_data = train_data, theta_n = theta_n,
19                     draw_n = 1e5, model = model, alpha = .15,
20                     thread_n = 60, chunk_n = 1, tol = 1e-3,
21                     maxEvals = 5e6, seed = 1800968452)
22
23 # saves machine/software/settings configuration for reproducibility
24 capture.output(sessionInfo(), file = paste0(
25   "results/fitting/jf-machine-software-config-norm-fit-",
26   Sys.Date(), ".txt"))
27 capture.output(settings[-1], file = paste0(
28   "results/fitting/jf-simulation-config-norm-fit-",
29   Sys.Date(), ".txt"))
30
31 # obtains posterior sample for a model
32 attach(settings, 2, warn.conflicts = FALSE)
33 timer <- proc.time()
34 res <- sample_posterior()
35 timer <- proc.time() - timer
36
37 # saves a array of posterior draws in plain text format
38 write.table(x = res,
39             file = "results/fitting/jf-posterior-chains-norm-fit.txt")
40
41 # saves computational time
42 capture.output(timer, file = paste0(
43   "results/fitting/jf-computational-time-norm-fit-",
44   Sys.Date(), ".txt"))

```

Code C.1: Script that plots quantiles-probability graphs

```

1 # load packages
2 library("copula")
3 library("mvtnorm")
4 library("magrittr")
5 library("dplyr")
6 library("RWiener")
7 library("Rcpp")
8 library("RcppArmadillo")
9 library("integral")
10
11 # source all the custom function relative to the working directory
12 source("src/fitting/calculate_weibull.R")
13 source("src/fitting/reflect.R")
14 source("src/fitting/check_support.R")
15 source("src/fitting/simulate_wiener_parameters.R")
16 source("src/fitting/simulate_rndwalk_rts.R")
17 source("src/fitting/sample_experiment.R")
18 source("src/fitting/initiate_sampler.R")
19 source("src/fitting/update_chains.R")
20 source("src/fitting/sample_posterior.R")
21 source("src/fitting/sample_am.R")

```

Code C.2: Script that plots quantiles-probability graphs

```

1 sample_posterior <- function() {
2   # Inputs: global variables defined in a script
3   # Output: returns double array posterior_all,
4   #          saves double matrix partial_res to disc
5   # Notes: theta_n + 1 contains likelihood calculations ,
6   #          theta_n + 2 contains acceptance indicator
7
8   progress_record <- paste0(
9     "results/fitting/kr-progress-log-norm-fit", ".txt")
10  writeLines("initialize chains", progress_record)
11
12  theta_idx <- -(theta_n + 1:2)
13  posterior_all <- matrix(0, draw_n, theta_n + 2)
14  partial_res <- matrix(0, draw_n, theta_n + 2)
15  tune_constant_1 <- .001 ^ 2 / theta_n
16  tune_constant_2 <- 2.38 ^ 2 / theta_n
17  posterior_all[1, theta_n + 2] <- tune_constant_2
18
19  set.seed(seed)
20  posterior_all[1, theta_idx] <- sample_prior(model)
21  data_mat <- combine_data(posterior_all[1, theta_idx])
22  timer <- proc.time()
23  posterior_all[1, theta_n + 1] <-
24    integral$joint_logdensity_cpp(data_mat, posterior_all[1, theta_idx],
25                                   model, thread_n, chunk_n,
26                                   tol, maxEvals)
27  timer <- proc.time() - timer
28  cat(paste("draw 1 time ", timer["elapsed"]),
29       file = progress_record, sep = "\n", append = TRUE)
30
31  save_points <- seq(5, draw_n + 5, 5)
32  for (draw in seq(2, draw_n)) {
33    chain_idx <- draw - 1
34    iter_idx <- seq_len(chain_idx)
35    timer <- proc.time()
36    posterior_all[draw, ] <- update_chains()
37    timer <- proc.time() - timer
38    cat(paste("draw", draw, "time ", timer["elapsed"]),
39         file = progress_record, sep = "\n", append = TRUE)
40    if (draw %in% save_points) {
41      partial_res[seq_len(draw), ] <- posterior_all[seq_len(draw), ]
42      save(partial_res,
43            file = "results/fitting/kr-posterior-chains-norm-fit.RData")
44    }
45  }
46  posterior_all
47}

```

Code C.3: Support function that samples from the posterior using parallelized adaptive mcmc

```

1 check_beta <- function(lambda, gamma) {
2   # Inputs: numerical vectors lambda, gamma
3   # Output: logical vector
4
5   check <- gamma ^ 2 < lambda * (1 - lambda)
6   check
7 }
```

Code C.4: Support function that checks that mean and variance of a beta variable are proper

```

1 check_chi_phi <- function(chi, phi) {
2   # Inputs: double vectors chi, phi
3   # Output: logical vector
4
5   check <- chi / phi >= 1
6   check
7 }
```

Code C.5: Support function that control for unreasonable combinations of chi and phi that drastically slow down integration

```

1 trans_theta_sampler <- function(theta) {
2   # Inputs: double vector theta
3   # Output: double vector theta
4
5   beta_sh1_1 <- ((1 - theta[7]) / theta[8] ^ 2 -
6     1 / theta[7]) * theta[7] ^ 2
7   beta_sh2_1 <- beta_sh1_1 * (1 / theta[7] - 1)
8   beta_sh1_2 <- ((1 - theta[17]) / theta[18] ^ 2 -
9     1 / theta[17]) * theta[17] ^ 2
10  beta_sh2_2 <- beta_sh1_2 * (1 / theta[17] - 1)
11  tau_sh_1 <- (theta[9] / theta[10]) ^ 2
12  tau_sc_1 <- theta[10] ^ 2 / theta[9]
13  tau_sh_2 <- (theta[19] / theta[20]) ^ 2
14  tau_sc_2 <- theta[20] ^ 2 / theta[19]
15  theta[c(7, 8, 17, 18, 9, 10, 19, 20)] <- c(beta_sh1_1, beta_sh2_1,
16                                beta_sh1_2, beta_sh2_2,
17                                tau_sh_1, tau_sc_1,
18                                tau_sh_2, tau_sc_2)
19  theta[c(1, 4:11, 14:20)] <- log(theta[c(1, 4:11, 14:20)])
20  theta
21 }
```

Code C.6: Support function that transform parameters into the sampling space

```

1 sample_prior <- function(model) {
2   # Inputs: character scalar model
3   # Output: double vector params
4   # Notes: order of parameters follows exposition of the model
5
6   repeat {
7     prior_theta <- c(runif(n = 1, min = 0.001, max = 0.786), # alpha
8                  runif(n = 1, min = -.5, max = 0), # nu_lo
9                  runif(n = 1, min = 0, max = .5), # nu_hi
10                 runif(n = 1, min = 0, max = 25), # nu_sh
11                 runif(n = 1, min = 0, max = 5), # nu_sc
12                 runif(n = 1, min = 0, max = 0.658), # eta
13                 runif(n = 1, min = .35, max = .65), # beta_mu
14                 runif(n = 1, min = 0, max = 0.5), # beta_sd
15                 runif(n = 1, min = 0, max = 1), # tnd_mu
16                 runif(n = 1, min = 0, max = .5), # tnd_sd
17                 runif(n = 1, min = 0.001, max = 0.786), # alpha
18                 runif(n = 1, min = -.5, max = 0), # nu_lo
19                 runif(n = 1, min = 0, max = .5), # nu_hi
20                 runif(n = 1, min = 0, max = 25), # nu_sh
21                 runif(n = 1, min = 0, max = 5), # nu_sc
22                 runif(n = 1, min = 0, max = 0.658), # eta
23                 runif(n = 1, min = .35, max = .65), # beta_mu
24                 runif(n = 1, min = 0, max = 0.5), # beta_sd
25                 runif(n = 1, min = 0, max = 1), # tnd_mu
26                 runif(n = 1, min = 0, max = .5)) # tnd_sd
27
28   if (initial)
29     prior_theta <- c(.221, -.352, .329, 4.413, .526, .127, .464,
30                      .065, .279, .041, .05, -.565, .511, 5.227,
31                      .521, .127, .464, .065, .279, .041)
32   if (all(check_beta(prior_theta[7], prior_theta[8]),
33           check_beta(prior_theta[17], prior_theta[18]),
34           check_chi_phi(prior_theta[9], prior_theta[10]),
35           check_chi_phi(prior_theta[19], prior_theta[20])))
36     break
37   }
38   prior_theta <- trans_theta_sampler(prior_theta)
39   if (model == "normal") {
40     prior_theta <- c(prior_theta,
41                       as.numeric(clusterGeneration::rcorrmatrix(d = 3,
42                                                       alpha = 1)[upper.tri(diag(3))]))
43     if (initial)
44       prior_theta <- c(prior_theta[1:20], 0, 0, 0)
45   }
46   prior_theta
47 }
```

Code C.7: Support function that samples initial vector of parameters for a given model

```

1 check_support <- function(theta_new) {
2   # Inputs: numeric vector theta_new
3   # Output: numeric vector
4
5   drift <- theta_new[c(2, 12)] < theta_new[c(3, 13)]
6
7   if (model == "normal") {
8     rho <- theta_new[21:23]
9     det_rho <- 1 - sum(rho ^ 2) + 2 * prod(rho)
10    test <- if (det_rho > 0) TRUE else FALSE
11    drift <- c(drift, test)
12  }
13  all(drift)
14}

```

Code C.8: Support function that checks proposed parameters are within proper sets

```

1 sample_behavior <- function(params, smpl_size, model, sigma = .1) {
2   # Inputs: double vector params, integer scalar smpl_size,
3   #          character scalar model, double scalar sigma
4   # Output: double matrix behav_smpl
5
6   trial_param <- smpl_param(params = params,
7                               smpl_size = smpl_size,
8                               model = model)
9
10  behav_smpl <- smpl_rts(alpha = trial_param$alpha,
11                           tau = trial_param$t_nd,
12                           beta = trial_param$beta,
13                           delta = trial_param$delta,
14                           sigma = sigma)
15  behav_smpl
16}

```

Code C.9: Support function that samples response times and responses

```

1 sample_experiment <- function(theta, model, prop, smpl_size) {
2   # Inputs: double vector theta, character scalar model,
3   #          double vector prop, integer scalar smpl_size
4   # Output: data_matrix data_mat
5
6   prop_n <- length(prop)
7   instr <- c(1, 0)
8   theta[1:20] <- exp(theta[1:20])
9
10  exper_theta <-
11    bind_rows(data_frame(alpha = theta[1],
12                          nu = weibull(bright = prop, lower = theta[2],
13                                         upper = theta[3], shape = theta
14                                         [4],
15                                         scale = theta[5]),
16                          eta = theta[6], shape1 = theta[7],
17                          shape2 = theta[8], shape = theta[9],
18                                         scale = theta[10]),
19    data_frame(alpha = theta[11],
20              nu = weibull(bright = prop, lower = theta[12],
21                            upper = theta[13], shape = theta
22                            [14],
23                            scale = theta[15]),
24              eta = theta[16], shape1 = theta[17],
25              shape2 = theta[18], shape = theta[19],
26                                         scale = theta[20]))
27  if (model == "normal") {
28    exper_theta <-
29    bind_cols(exper_theta,
30              data_frame(
31                rho_db = rep(x = theta[21], times = 2 * prop_n),
32                rho_dt = rep(x = theta[22], times = 2 * prop_n),
33                rho_bt = rep(x = theta[23], times = 2 * prop_n)))
34  }
35
36 behav_data <- apply(X = exper_theta, MARGIN = 1, FUN = sample_behavior
37   ,
38   smpl_size = smpl_size, model = model) %>%
39   bind_rows
40 colnames(behav_data) <- c("rt", "resp")
41 prop_data <- data_frame(prop = rep(prop, each = smpl_size) %>%
42                           rep(times = 2))
43 instr_data <- data_frame(instr = rep(instr, each = prop_n * smpl_size)
44   )
45 data_mat <- bind_cols(behav_data, prop_data, instr_data)
46 data_mat
47 }
```

Code C.10: Support function that samples data from a brightness discrimination task

```

1 combine_data <- function(theta) {
2   # Inputs: data_frame train_data, double vector theta
3   # Output: data_frame dat_mat
4
5   theta [c(1, 4:11, 14:20)] <- exp(theta [c(1, 4:11, 14:20)])
6
7   acc <- data_frame(rt = filter(train_data, instr == 1) %>%
8     dplyr::select(rt) %>%
9     unlist(use.names = FALSE),
10    choice = filter(train_data, instr == 1) %>%
11      dplyr::select(resp) %>%
12        unlist(use.names = FALSE),
13        alpha = theta [1],
14        nu = weibull(filter(train_data, instr == 1) %>%
15          dplyr::select(prop) %>%
16            unlist(use.names = FALSE),
17              theta [2], theta [3], theta [4], theta [5]),
18              eta = theta [6], shape1 = theta [7], shape2 = theta [8],
19              shape = theta [9], scale = theta [10], rho_db = theta
20                [21],
21                rho_dt = theta [22], rho_bt = theta [23])
22   spd <- data_frame(rt = filter(train_data, instr == 0) %>%
23     dplyr::select(rt) %>%
24     unlist(use.names = FALSE),
25    choice = filter(train_data, instr == 0) %>%
26      dplyr::select(resp) %>%
27        unlist(use.names = FALSE),
28        alpha = theta [11],
29        nu = weibull(filter(train_data, instr == 0) %>%
30          dplyr::select(prop) %>%
31            unlist(use.names = FALSE),
32              theta [12], theta [13], theta [14], theta
33                [15]),
34              eta = theta [16], shape1 = theta [17], shape2 = theta
35                [18],
36                shape = theta [19], scale = theta [20], rho_db = theta
37                  [21],
38                  rho_dt = theta [22], rho_bt = theta [23])
39   as.matrix(rbind.data.frame(acc, spd))
40 }

```

Code C.11: Support function that forms a data matrix with observations and parameters

```

1 update_chains <- function() {
2   # Inputs: double matrix posterior_cur,
3   #          integer scalars chain_n, theta_n
4   # Output: double matrix posterior_updated
5   e1 <- parent.frame(1)
6
7   theta_proposal <- sample_am()
8   joint_old <- e1$posterior_all[e1$chain_idx, theta_n + 1]
9
10  if (theta_proposal[theta_n + 1] == 0)
11    return(e1$posterior_all[e1$chain_idx, ])
12  else
13    data_mat <- combine_data(theta_proposal[e1$theta_idx])
14
15  joint_new <- integral$joint_logdensity_cpp(
16    data_mat, theta_proposal[e1$theta_idx],
17    model, thread_n, chunk_n, tol, maxEvals)
18
19  alpha_log <- min(joint_new - joint_old, 0)
20  tune_updated <- e1$posterior_all[e1$chain_idx, theta_n + 2]
21  if (e1$chain_idx > 200)
22    tune_updated <- exp(log(tune_updated) +
23      1 / e1$chain_idx^(1 / 2) *
24      (exp(alpha_log) - alpha))
25
26  if (log(runif(1)) < alpha_log)
27    c(theta_proposal[e1$theta_idx], joint_new, tune_updated)
28  else
29    c(e1$posterior_all[e1$chain_idx, e1$theta_idx],
30      joint_old, tune_updated)
31}

```

Code C.12: Support function that explores the posterior with MCMC

```

1 reflect <- function(theta_new) {
2   # Inputs: double vector theta_new, character scalar model (global)
3   # Output: double vector theta_new
4
5   if (model == "independent")
6     return(theta_new)
7
8   results <- logical(3)
9
10  while (any(!results)) {
11    test <- findInterval(theta_new[21], c(-1, 1))
12    if (test == 1)
13      results[1] <- TRUE
14    else if (test == 0)
15      theta_new[21] <- -2 - theta_new[21]
16    else if (test == 2)
17      theta_new[21] <- 2 - theta_new[21]
18
19    test <- findInterval(theta_new[22], c(-1, 1))
20    if (test == 1)
21      results[2] <- TRUE
22    else if (test == 0)
23      theta_new[22] <- -2 - theta_new[22]
24    else if (test == 2)
25      theta_new[22] <- 2 - theta_new[22]
26
27    test <- findInterval(theta_new[23], c(-1, 1))
28    if (test == 1)
29      results[3] <- TRUE
30    else if (test == 0)
31      theta_new[23] <- -2 - theta_new[23]
32    else if (test == 2)
33      theta_new[23] <- 2 - theta_new[23]
34  }
35  theta_new
36}

```

Code C.13: Support function that checks that a MCMC proposal falls into the posterior support and reflects it back if not

```

1 sample_am <- function() {
2   # Inputs: integer vectors chain_idx, theta_idx ,
3   #          double matrix posterior_all ,
4   #          double scalar tune_constant (all global)
5   # Output: double vector c(theta_reflected , 1) or
6   #          double vector c(theta_old , 1)
7
8   e1 <- parent.frame(1)
9   e2 <- parent.frame(2)
10
11  tune_constant_2 <- e2$posterior_all [e2$chain_idx , theta_n + 2]
12  if (e2$chain_idx < 200)
13    cov_estimate <- e2$tune_constant_1 * diag(theta_n)
14  else
15    cov_estimate <- tune_constant_2 *
16                  cov(e2$posterior_all [e2$iter_idx , e2$theta_idx])
17  theta_old <- e2$posterior_all [e2$chain_idx , e2$theta_idx]
18  theta_new <- rmvnorm(1, theta_old , cov_estimate)
19  theta_reflected <- reflect(theta_new)
20  check <- check_support(theta_reflected)
21  if (check)
22    c(theta_reflected , 1)
23  else
24    c(theta_old , 0)
25}

```

Code C.14: Support function that proposes steps according to adaptive metropolis

```

1 PKG_CXXFLAGS="-fopenmp"
2 PKG_LIBS=$(shell $(R_HOME)/bin/Rscript -e "Rcpp:::LdFlags()") $(LAPACK_
LIBS) $(BLAS_LIBS) $(FLIBS) -fopenmp

```

Code C.15: Instructions to the GNU g++ compiler to properly compile parallelized C++ code that can be included in a Makevars file

```

1 #include <RcppArmadillo.h>
2 using namespace Rcpp;
3 using namespace arma;
4 using namespace std;
5
6 vec forwsub_c(mat A, vec b) {
7   vec x = zeros(3);
8   for (int i = 0; i < 3; ++i) {
9     x[i] = (b[i] - as_scalar(A.row(i) * x)) / A(i, i);
10   }
11   return x;
12 }

```

Code C.16: Support function that solves linear system with forward substitution algorithm and used for evaluation of multivariate normal density

```

1 double normal_logcopula_c(vec cop_value, mat sqrt_rho) {
2     vec x(3), marginals(3);
3     unsigned i;
4
5     for (i = 0; i < 3; ++i) {
6         x[i] = R::qnorm(cop_value[i], 0, 1, true, false);
7         marginals[i] = R::dnorm(x[i], 0, 1, false);
8     }
9     uvec test = marginals == 0.0;
10    if (any(test)) {
11        marginals.elem(find(test)).fill(DBL_EPSILON);
12    }
13    vec sqrt_rho_x = forwsub_c(trans(sqrt_rho), x);
14    vec rho_x = sum(pow(sqrt_rho_x, 2));
15    vec rho_diag = sqrt_rho.diag();
16    test = rho_diag == 0.0;
17    if (any(test)) {
18        rho_diag.elem(find(test)).fill(DBL_EPSILON);
19    }
20    double logdens = as_scalar((-1.5 * log(2 * M_PI) -
21                                sum(log(rho_diag)) -
22                                0.5 * rho_x) -
23                                sum(log(marginals)));
24    return logdens;
25}

```

Code C.17: Support function that evaluates normal copula log-density function

```

1 double calc_logdensity_c(double delta, double beta, double t_nd,
2                           double nu, double eta, double shape1,
3                           double shape2, double shape, double scale,
4                           mat sqrt_rho, string model) {
5   vec marginal_dens(3);
6   marginal_dens[0] = R::dnorm(delta, nu, eta, false);
7   marginal_dens[1] = R::dbeta(beta, shape1, shape2, false);
8   marginal_dens[2] = R::dgamma(t_nd, shape, scale, false);
9
10  uvec test_bounds = marginal_dens == 0.0;
11  if (any(test_bounds)) {
12    marginal_dens.elem(find(test_bounds)).fill(DBL_EPSILON);
13  }
14  bool test_inf = marginal_dens.is_finite();
15  if (test_inf) {
16    marginal_dens.elem(find_nonfinite(marginal_dens)).fill(
17      numeric_limits<double>::max());
18  }
19  double marginal_logdens = sum(log(marginal_dens));
20
21  if (model == "independent")
22    return marginal_logdens;
23
24  vec cop_value(3);
25  cop_value[0] = R::pnorm(delta, nu, eta, true, false);
26  cop_value[1] = R::pbeta(beta, shape1, shape2, true, false);
27  cop_value[2] = R::pgamma(t_nd, shape, scale, true, false);
28
29  test_bounds = cop_value == 0.0;
30  if (any(test_bounds)) {
31    cop_value.elem(find(test_bounds)).fill(DBL_EPSILON);
32  }
33  test_bounds = cop_value == 1.0;
34  if (any(test_bounds)) {
35    cop_value.elem(find(test_bounds)).fill(1 - DBL_EPSILON);
36  }
37  double joint_logdens = normal_logcopula_c(cop_value, sqrt_rho);
38
39  return marginal_logdens + joint_logdens;
40}
41

```

Code C.18: Support function that evaluates log-density of Wiener parameters

```

1 double dwiener_cpp(double q, double alpha, double tau,
2                     double beta, double delta, double choice) {
3     double kl, ks, ans;
4     int k,K;
5     double err = 1e-6;
6     // choice is 1 for lower, 2 for upper boundary
7     if (choice == 2.0) {
8         beta = 1 - beta;
9         delta = -delta;
10    }
11    q = (q - tau) / pow(alpha, 2);
12    if (M_PI * q * err < 1) {
13        kl = sqrt(-2 * log(M_PI * q * err)) / (pow(M_PI, 2) * q));
14        kl = kl > 1 / (M_PI * sqrt(q)) ? kl : 1 / (M_PI * sqrt(q));
15    } else {
16        kl = 1 / (M_PI * sqrt(q));
17    }
18    if ((2 * sqrt(2 * M_PI * q) * err) < 1) {
19        ks = 2 + sqrt(-2 * q * log(2 * sqrt(2 * M_PI * q) * err));
20        ks = ks > sqrt(q) + 1 ? ks : sqrt(q) + 1;
21    } else {
22        ks = 2;
23    }
24    ans = 0;
25    if (ks < kl) {
26        K = ceil(ks);
27        for (k = -floor((K - 1) / 2); k <= ceil((K - 1) / 2); k++) {
28            ans = ans + (beta + 2 * k) * exp(-(pow((beta + 2 * k), 2))
29                / 2 / q);
30        }
31        ans = ans / sqrt(2 * M_PI * pow(q, 3));
32    } else {
33        K = ceil(kl);
34        for (k = 1; k <= K; k++) {
35            ans = ans + k * exp(-(pow(k, 2)) * (pow(M_PI, 2)) * q / 2) *
36                sin(k * M_PI * beta);
37        }
38        ans = ans * M_PI;
39    }
40    ans = ans * exp(-delta * alpha * beta - (pow(delta, 2)) *
41        (q * pow(alpha, 2)) / 2) / (pow(alpha, 2));
42    if (ans < numeric_limits<double>::min() ||
43        ans == datum::inf ||
44        isnan(ans)) {
45        ans = DBL_EPSILON;
46    }
47    return ans;
48}

```

Code C.19: Support function that evaluates joint density of response time and response

```

1 double calc_density_integrand_c(const double * x, double rt,
2                                 double choice, double sigma,
3                                 double alpha, double nu,
4                                 double eta, double shape1,
5                                 double shape2, double shape,
6                                 double scale, mat sqrt_rho,
7                                 string model) {
8     double delta_trans = *x;
9     double beta = *(x + 1);
10    double t_nd = *(x + 2);
11    double delta = delta_trans / (1 - pow(delta_trans, 2));
12
13    double behav_dens = dwiener_cpp(rt, alpha / sigma, t_nd,
14                                     beta, delta / sigma, choice);
15    double param_logdens = calc_logdensity_c(delta, beta, t_nd,
16                                              nu, eta, shape1, shape2,
17                                              shape, scale,
18                                              sqrt_rho, model);
19
20    double jacobian = (1 + pow(delta_trans, 2)) /
21                      pow(1 - pow(delta_trans, 2), 2);
22
23    double integrand = exp(log(behav_dens) +
24                           param_logdens +
25                           log(jacobian));
26
27    if (isnan(integrand) || integrand == datum::inf)
28        integrand = DBL_EPSILON;
29
30    return integrand;
31}

```

Code C.20: Support function that evaluates integrand which is a product of data and parameter densities

```

1 #include <limits>
2 #include <RcppArmadillo.h>
3 #include <omp.h>
4 #include "integrand.h"
5 #include "cubature.h"
6
7 using namespace Rcpp;
8 using namespace std;
9 using namespace arma;
10
11 struct params {
12     double rt;
13     double choice;
14     double sigma;
15     double alpha;
16     double nu;
17     double eta;
18     double shape1;
19     double shape2;
20     double shape;
21     double scale;
22     mat sqrt_rho;
23     string model;
24     unsigned * counts;
25 };
26
27 double calc_prior_cpp(vec theta, string model, double omega = 1) {
28     double prior_density = 0;
29     unsigned i;
30     for (i = 0; i < 20; ++i)
31         prior_density += R::dnorm(theta[i], 0, 100, true);
32     if (model == "normal")
33         prior_density += log(pow((1 - pow(theta[20], 2) -
34                             pow(theta[21], 2) -
35                             pow(theta[22], 2) +
36                             2 * theta[20] *
37                             theta[21] * theta[22]), omega - 1.0));
38
39     return prior_density;
40 }

```

Code C.21: Support function that evaluates prior density of unknown parameters

```

1 int fwrap_v(unsigned ndim, size_t npts, const double * x,
2             void * fdata, unsigned fdim, double * fval) {
3     params mydata = *((params *) fdata);
4     vec res(fval, npts, false);
5     unsigned i;
6     for (i = 0; i < npts; ++i) {
7         double y[3];
8         y[0] = x[i * 3];
9         y[1] = x[i * 3 + 1];
10        y[2] = x[i * 3 + 2];
11
12        res[i] = calc_density_integrand_c(y, mydata.rt,
13                                           mydata.choice,
14                                           mydata.sigma,
15                                           mydata.alpha,
16                                           mydata.nu,
17                                           mydata.eta,
18                                           mydata.shape1,
19                                           mydata.shape2,
20                                           mydata.shape,
21                                           mydata.scale,
22                                           mydata.sqrt_rho,
23                                           mydata.model);
24    }
25    *mydata.counts += npts;
26    return 0;
27}

```

Code C.22: Support function that transforms copula draws to parameter draws using inverse cdfs and adds the threshold parameter to completely define a random walk

```

1 double integrate_density(double rt, double choice, double sigma,
2                           double alpha, double nu, double eta,
3                           double shape1, double shape2, double shape,
4                           double scale, mat sqrt_rho, string model,
5                           double tol, size_t maxEval) {
6     double integral;
7     double error;
8     unsigned counts;
9     params mydata;
10
11    double * integral_pt = &integral;
12    double * error_pt = &error;
13
14    mydata.rt = rt;
15    mydata.choice = choice;
16    mydata.sigma = sigma;
17    mydata.alpha = alpha;
18    mydata.nu = nu;
19    mydata.eta = eta;
20    mydata.shape1 = shape1;
21    mydata.shape2 = shape2;
22    mydata.shape = shape;
23    mydata.scale = scale;
24    mydata.sqrt_rho = sqrt_rho;
25    mydata.model = model;
26    mydata.counts = &counts;
27
28    const double xmin[3] = {-1 + DBL_EPSILON,
29                           numeric_limits<double>::min(),
30                           numeric_limits<double>::min()};
31    xmax[3] = {1 - DBL_EPSILON,
32               1 - DBL_EPSILON,
33               rt};
34
35    do {
36        counts = 0;
37        hcubature_v(1, fwrap_v, &mydata, 3, xmin, xmax,
38                     maxEval, 0, tol, ERROR_L1, integral_pt, error_pt);
39        tol *= 0.1;
40    } while (counts < 2e3);
41
42    return integral;
43}

```

Code C.23: Support function that calculates triple integral of behavioral density with respect to density of parameters

```

1 vec calc_likelihood_cpp(mat data_mat, string model, size_t thread_n,
2                         size_t chunk_n, double tol, size_t maxEvals) {
3     mat sqrt_rho;
4     if (model == "normal") {
5         double rho_db = data_mat(0, 9);
6         double rho_dt = data_mat(0, 10);
7         double rho_bt = data_mat(0, 11);
8         sqrt_rho << 1 << rho_db << rho_dt << endl
9             << rho_db << 1 << rho_bt << endl
10            << rho_dt << rho_bt << 1 << endl;
11         sqrt_rho = chol(sqrt_rho);
12     }
13
14     unsigned i, n;
15     n = data_mat.n_rows;
16     vec behav_density(n);
17
18     vec rt(data_mat.colptr(0), n, false);
19     vec choice(data_mat.colptr(1), n, false);
20     vec alpha(data_mat.colptr(2), n, false);
21     vec nu(data_mat.colptr(3), n, false);
22     vec eta(data_mat.colptr(4), n, false);
23     vec shape1(data_mat.colptr(5), n, false);
24     vec shape2(data_mat.colptr(6), n, false);
25     vec shape(data_mat.colptr(7), n, false);
26     vec scale(data_mat.colptr(8), n, false);
27
28 #pragma omp parallel for \
29 schedule(dynamic, chunk_n) \
30 num_threads(thread_n)
31
32     for (i = 0; i < n; ++i) {
33         behav_density[i] = integrate_density(rt[i], choice[i], 0.1,
34                                             alpha[i], nu[i], eta[i],
35                                             shape1[i], shape2[i],
36                                             shape[i], scale[i], sqrt_rho,
37                                             model, tol, maxEvals);
38     }
39
40     uvec zero_test = behav_density <= 0;
41     if (any(zero_test))
42         behav_density.elem(find(zero_test)).fill(DBL_EPSILON);
43     return behav_density;
44 }
```

Code C.24: Support function that evaluates likelihood function of response times and responses

```

1 double joint_logdensity_cpp(mat data_mat, vec theta_prop, string model,
2                             unsigned thread_n, unsigned chunk_n,
3                             double tol, unsigned maxEvals) {
4     double logprior = calc_prior_cpp(theta_prop, model);
5     vec likelihood = calc_likelihood_cpp(data_mat, model, thread_n,
6                                         chunk_n, tol, maxEvals);
7     unsigned i, n;
8     n = likelihood.n_elem;
9     double loglikelihood = 0;
10
11    for (i = 0; i < n; ++i) {
12        loglikelihood += log(likelihood[i]);
13    }
14    return logprior + loglikelihood;
15}

```

Code C.25: Support function that evaluates joint density of behavior and parameters

```

1 RCPP_MODULE(integral) {
2     Rcpp::function("calc_likelihood_cpp",
3                     &calc_likelihood_cpp,
4                     "calculates likelihood");
5 }

```

Code C.26: A method for exporting a C++ to R