Advancing Markov Decision Processes and Multivariate Gaussian Processes as Tools for Computational Psychiatry

Diss. ETH No. 24661

Advancing Markov Decision Processes and Multivariate Gaussian Processes as Tools for Computational Psychiatry

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Daniel Renz

Dipl-Inf. in Computer Science, Humboldt University Berlin
born 17.10.1980
citizen of Germany

accepted on the recommendation of

Prof. Dr. Dr.med. Klaas E. Stephan, examiner
Prof. Dr. Karsten M. Borgwardt, co-examiner
Dr. Dr.med. Quentin J.M. Huys, co-examiner

2018

Contents

Abstract (english)

In the relatively recent field of computational psychiatry, computational methods can advance the current state of the art in various ways. Progress is possible by advancing current methods of statistical analyses from a model first perspective, for example to increase sensitivity or to regularize complex models. We propose a simple extension to multiple sequential linear regression - the *Gaussian Process Trajectory Prediction*. The method predicts individual non-linear time courses of a longitudinal marker based on cross-sectional baseline data. It implicitly assumes that each baseline feature contributes a temporal kernel (i.e., feature weight trajectory) to the evolution of the marker. The Gaussian Process acts as a prior on these feature weight trajectories, making an otherwise overly complex regression problem solvable in many practical settings. The time course of the clinical marker then arises as a weighted linear combination of these kernels. The method is tested on a large trial of depression treatments, where it provides prediction of treatment success sufficiently better than previous approaches as well as the first prediction of chance of relapse.

While methods that do not rest on a model of the cognitive or neurobiological processes may provide improved predictions, they do not help to gain better insights into how psychiatric diseases work mechanistically. Thus, we investigate in the second part models of complex, sequential decision-making. This is relevant because in order to solve the sequential decision-making problem, the problem of how and in which order to evaluate the possible options before committing to an actual decision (the meta-decision problem) has to be considered. Emotions are thought to result in fast approximations to this meta problem, and since depression is an emotional disorder, meta-decision making might be dysfunctional in depression. We develop several detailed models describing both the decision and meta-decision processes, and fit them to data that was collected from an experiment specifically designed for this purpose. We focus on the development of inference methods, as the problem of how exactly to fit the models turns out to be hard. Thus, we develop an approximative Expectation-Maximization sampling scheme across trials, as well as a trial-by-trial inference based on advanced Markov Chain Monte Carlo methods.

Abstract (deutsch)

In dem relativ neuen Gebiet der "Computational Psychiatry" können rechnerische Methoden den aktuellen Stand der Wissenschaft auf verschiedene Art voranbringen. Aktuelle Methoden der statistischen Analyse können verbessert werden, zum Beispiel um ihre Sensitivität zu erhöhen oder um komplexe Modelle zu regularisieren. Wir schlagen eine einfache Erweiterung der multiplen sequentiellen linearen Regression vor, die *Gaussian Process Trajectory Prediction*. Diese Methode sagt individuelle nicht-lineare Zeitreihen eines longitudinalen Markers voraus, basierend auf Querschnitts-Daten von einer Baseline-Messung. Sie nimmt implizit an, dass jedes gemessene Merkmal der Baseline einen zeitlichen Kernel (d.h. eine Merkmals-Gewichts-Trajektorie), zu der Evolution des klinischen Markers beiträgt. Der Gaußsche Prozess agiert as A-Priori Wahrscheinlichkeit auf diesen Merkmals-Gewichts-Trajektorien, so dass ein andernfalls zu komplexes Regressionsproblem in vielen praktischen Situationen lösbar wird. Die Zeitreihe des klinischen Markers ist dann eine gewichtete lineare Kombination der Merkmals-Kernel. Die Methode wird auf Daten einer grossen klinischen Studie zur Behandlung von Depression getestet und verbessert die Vorhersage von Behandlungserfolg im Vergleich zu vorherigen Ansätzen, sowie liefert die erste Vorhersage von Rückfall in eine depressive Episode auf diesen Daten.

Während Methoden, die nicht auf einem Modell der kognitiven oder neurobiologischen Prozesse basieren, bessere Vorhersagen liefern können, helfen sie nicht dabei zu verstehen, wie psychiatrische Krankheiten mechanistisch funktionieren. Deshalb untersuchen wir im zweiten Teil Modelle der sequentiellen Entscheidungsfindung. Dies ist relevant, weil um sequentielle Entscheidungen machen zu können zunächst das Problem gelöst werden muss, wie and in welcher Reihenfolge die möglichen Optionen evaluiert werden sollen bevor eine Entscheidung getroffen wird (Das Meta-Entscheidungsproblem). Emotionen resultieren in schnellen Heuristiken dieses Meta-Problems, und da Depression eine emotionale Krankheit ist, könnte die Meta-Entscheidungsfindung in der Depression dysfunktional sein. Wir entwickeln mehrere detaillierte Modelle, die sowohl den Entscheidungsprozess als auch den Meta-Prozess beschreiben, und benutzen sie, um Daten zu beschreiben, die aus einem eigens für diesen Zweck durchgeführten Experiment stammen. Wir fokusieren uns auf die Entwicklung der Inferenz-Methoden, da es sich als schwierig herausstellt, wie genau Inferenz praktisch durchgeführt werden kann. Daher entwickeln wir sowohl ein approximatives Expectation-Maximization-Schema, das über die einzelnen Trials des Experiments mittelt, als auch ein auf fortgeschrittenen Markov Chain Monte Carlo Methoden basierendes Trial-für-Trial Schema.

1 Introduction

The challenges of mental health are enormous: Not only is the brain the most complex organ and far from being understood well, but it is also interacting extensively with its complex environment at multiple levels, from molecules to neurobiological circuits, cognition, behavior and finally the social environment. One consequence is that because our current understanding of the brain is so limited, psychiatric diseases are identified solely based on their symptoms according to some manual such as the Diagnostic and Statistical Manual of Mental Disorders (DSM-5; Ebmeier et al. (2006)). This is problematic for several reasons. First, such syndrome-based classification has no connection to underlying pathophysiology. At best, research concerning the symptom-based psychiatric classification of diseases confirms existing classification schemata, and does so at the expense of considerable resources, both in terms of time and money. But such symptom-based type of classification (which is how classification was done in other medical disciplines many decades ago) is itself flawed.

Not only can different diseases share the same symptoms, but individuals with the same disease can also have completely different symptoms. A substantial amount of research has been dedicated to identifying variables that differ significantly across psychiatric diagnostic categories (Stephan et al., 2017). In some situations, another complication is that the efficacy of available drugs is comparatively low, which might be caused by limitations of the existing classification schemata; several different neurobiological disorders might underlie the same (group of) symptoms. For example, in depression the medications that are most often prescribed are Selective Serotonin Reuptake Inhibitors (SSRIs). In the largest prospective study of treatment of acute depression, the SSRI Citalopram was effective in only 36.8 percent of cases in the sense that it reduced depressive symptoms below the clinical classification threshold for remission (Rush et al., 2006).

To complicate things even further, comorbidity (the joint occurrence of two or more mental disorders) is the rule rather than the exception (Kessler et al., 2005b). In practice, comorbidity is often investigated by analyzing the association between composite measures defined on sets of items (i.e., a correlation between scores on checklists). While this has yielded insights into which disorders co-occur (Kessler et al., 2005b; Merikangas et al., 1998) or what the underlying risk factors are (Beekman et al., 2000; Kendler et al., 2007), symptoms in this view are merely indicators of latent disorders that do the actual causal work (Borsboom, 2008). In classic data-driven approaches in psychiatry, the focus is usually on stratification based on symptom severity within a single diagnostic category, for example, schizophrenia (Sun et al., 2015), psychotic disorders (Clementz et al., 2016), depression (Lamers et al., 2012; Rhebergen et al., 2012; van Loo et al., 2012, 2014), attention-deficit/hyperactivity disorder (Mostert et al., 2015) and autism (Veatch et al., 2014). But this focus cannot address the need to characterize the heterogeneity and overlap of symptoms across diagnostic categories. Finally, not only are psychiatric diseases and their symptoms over-

lapping, but so is their treatment; for example, severely depressed patients often develop additional psychotic symptoms, which are being treated with separate drugs (Kupfer et al., 2012).

In major depressive disorder (MDD) there are ongoing efforts to find differences between groups of individuals using genomic data (Schatzberg et al., 2015) or structural and functional magnetic resonance imaging (Schmaal et al., 2015), but performance in predicting outcomes remains modest (Trivedi, 2013; Trivedi et al., 2016). The main problem, as mentioned before for psychiatry in general, is that due to the heterogeneity of depressive symptoms, two patients suffering from depression might not share more than a single out of nine symptoms used for assessing major depressive disorder (Fried and Nesse, 2015; Olbert et al., 2014). A treatment might be effective for one group of symptoms, but not for another, potentially explaining mixed results from large comparative efficacy meta-analyses (Cipriani et al., 2009; Gartlehner et al., 2011). For example, patients who were abused as children have been shown to respond better to cognitive behavioral therapy than to antidepressant medication (Nemeroff et al., 2003). A study investigating if it was possible to distinguish unipolar from bipolar depression was able to reach 87 percent classification accuracy by aggregating many minor variations - for example, slightly more pessimistic thoughts in bipolar patients, slightly more apparent sadness in unipolar patients (Perlis et al., 2006). Given the overwhelmingly many ways in which any two patients may be dissimilar, the number of rules to remember in order to successfully perform a classification vastly exceeds human capacity (Perlis, 2016).

Computational psychiatry (CP) is a relatively new field within psychiatry that aims to improve functional understanding of disease as well as predictive power. New statistical tools that have only recently become available enable in principle prediction of, say, treatment success for *individual patients* suffering from the same disease. Scientific studies have not investigated such individual differences of treatment efficacy much. As a result, current clinical practice usually consists of randomly assigning patients to what is deemed to be the most effective medication across individuals. CP to enable studies of inter-individual differences in order to answer the questions that clinicians usually ask; these might sound deceptively simple but are hard for scientists to answer (Rush, 2015): Which treatment has the best efficacy? Which clinical symptoms are associated with better outcomes of one treatment as opposed to another? What to do when first-line treatments fail?

CP, unlike previous approaches, directly addresses the fundamental computational nature of brain function (Montague et al., 2012; Huys, 2018). It is a pragmatic discipline with the goal of obtaining clinically immediately useful results. While traditional models are not suitable for the task of individual prediction, because they are unable to handle the kind of highly complex models that are necessary, CP uses novel statistical techniques which are often referred to under the label machine learning. The crucial difference is that specific combinations of variables do not have to be manually predetermined any more, rather algorithms iteratively comb through the data and determine on their own which

variables are relevant (Chekroud, 2017).

CP is not the first scientific effort which aims to find individual differences for the purpose of prediction. In the last decades, large efforts have been spent on the identification of biomarkers that may establish the presence of or risk for a particular psychiatric disorder (Singh and Rose, 2009; Davis et al., 2015; Treadway and Leonard, 2016). However, advances have been much slower than expected, as most of the promising biomarker candidates have been found to have low sensitivity and specificity. Here, again, significant comorbidity across disorders and the sheer complexity of the phenotypes have hindered the identification of disorder-specific pathophysiology in individuals (Singh and Rose, 2009; Kapur et al., 2012).

The absence of markers for symptomatic states in individuals can be difficult to reconcile with a number of reliable group-level findings in psychiatric populations. For example, many studies and meta-analyses have found that anxiety is associated with increased amygdala responsivity (Etkin and Wager, 2007; Ipser et al., 2013), or that patients with major depression exhibit structural reductions in brain matter in prefrontal and hippocampal areas (Koolschijn et al., 2009; Bora et al., 2011). However, unfortunately, these effects only exist on average, and vanish at the level of the individual (McMahon, 2014).

There are two different types of approaches within CP: theory-driven and data-driven (Huys et al., 2016), corresponding to the two main sections of this work. Theory-driven methods use mathematical and statistical tools to understand the functional mechanism of psychiatric illnesses (Huys et al., 2016). They may elucidate behavioral and neurobiological processes underlying a disease, which is necessary in order to discover novel targets and interventions (Krystal et al., 2017). In contrast, the data-driven approach to CP makes use of a wide range of computational techniques which are largely neurobiologically and cognitively agnostic. The aim is to identify predictive patterns, both for small and big data (Chekroud, 2017).

In theory-driven methods, the model space is huge, ranging from cellular biophysical models to algorithmic models that address the algorithms that the brain might use to implement certain computations, and how behavior is a result of those computations. Already, a large part of current research is focused on finding mechanistic explanations of mental illness (Rothman and Greenland, 2005; Weiskopf, 2011). From such an information-processing perspective, building formal models of specific brain processes involves 3 levels of analysis (Marr and Poggio, 1976). The levels describe the nature of a problem (computational level), the algorithm that solves it (representational level), and how the algorithm is implemented (physical level). Such characterization of relevant processes should lead to identification of targets for clinical intervention. Examples are the characterization of cognitive processes and how to target them with cognitive modifications (Borsboom et al., 2011), or how to target task-related neural activations through neurostimulation (Wan Lee et al., 2014). There are already attempts that offer a complete formal computational taxonomy for understanding psychiatric disease (Petzschner et al., 2017). The authors propose a general statistical framework

for behavior by conceptualizing it in terms of loops between beliefs and observations. Psychiatric phenotypes might effectively be described by isolating specific components of this framework. Ideally, such models produce parameters that are varying across individuals, which can be integrated into predictive data-driven machine learning models to improve clinical decision-making.

A specific theory-driven tool that is being used in CP is generative models. They produce simulated data which are similar to actual data obtained in an experiment. Statistically inverting these models (=fitting them to data) involves fitting their parameters such that the generated data becomes similar to the observed data. This is in contrast to standard approaches to data analysis, which usually involve focusing on often very small parts of the data (Huys et al., 2016). In comparison to classic approaches, generative models increase robustness and generalizability, as well as allowing for quantitative assessments of model complexity, such that the most parsimonious models can be chosen.

While theory-driven models have the potential to greatly increase our understanding of mechanisms of brain function and disease, from a practical perspective such explanatory research is strictly speaking not necessary for treatment. Currently, there still does not exist any biological diagnostic or prognostic test in psychiatry, mainly because the task of making neuroscientific research useful for clinical psychiatry is an extremely difficult problem (Paulus, 2015). It might be a long while before any practical benefits from improved understanding of the functional mechanisms of the brain may become apparent, because of the complexity of the task at hand. Instead, it might be useful to pragmatically focus on prediction.

The data-driven approach to CP makes use of a wide range of computational techniques which are largely neurobiologically and cognitively agnostic. The aim is to identify predictive patterns, both for small and big data (Chekroud, 2017). The pragmatic approach here is that predictive relationships between data and relevant outcomes matter even if they do not offer mechanistic insight. As a consequence, nearly any data can be analyzed. Big data especially enables researchers to ask more powerful questions: with hundreds of thousands of data points, new and meaningful insights might be discovered to help physicians make more informed decisions (Torous and Baker, 2016). Big data methods have just recently been employed to explore associations between the symptom profile of a depressed patient, and treatment outcomes (Chekroud et al., 2016, 2017a). Crucially, the symptom profiles in these studies consisted of several hundred variables, too many for classical approaches. The algorithm identified a small set of clinical features that predicted treatment success, and can easily be administered in the form a questionnaire. Tools such as these, unlike classic analytic approaches, are able to handle complex, high-dimensional, data sets. In current research, such tools are usually labeled with the term machine learning (Chekroud, 2017). Not only do they enable new, sophisticated experiments, but they also have the potential to find new patterns in clinical trial data that have already been collected (Chekroud, 2017).

A recent example aimed at improving disease classification is based on the

network perspective on mental disorders (Borsboom and Cramer, 2013; Borsboom, 2015; Cramer et al., 2010). It starts out by assuming that symptoms form a network in which symptoms can directly be caused by other symptoms (Keller et al., 2007; Cramer et al., 2016), and individuals may differ in terms of the strength of connections between symptoms. This differs from the classic perspective where symptoms are viewed as indicators of latent disorders. In such a network approach to mental disorders and comorbidity, symptoms are viewed as components in a network, so comorbidity is hypothesized to result from direct relations between symptoms (Cramer et al., 2010; Borsboom et al., 2011). For example, if a person suffers from insomnia for a while, this person might start experiencing fatigue. A disease can be viewed as such a network of directly related symptoms. From this perspective, the network and its dynamics can more generally be referred to as a complex dynamic system (Schmittmann et al., 2013): It is complex because relations between symptoms might result in outcomes that are impossible to predict from any single individual symptom, and dynamic because this network is hypothesized to evolve over time. For example, major depressive disorder could be a bistable system with two attractor states: a 'non-depressed' and a 'depressed' state (Fried et al., 2016).

It is possible to use a purely data-driven approach to identify transdiagnostic subtypes across mood, anxiety and trauma disorders, as there is large overlap across these disorders (Kessler et al., 2005a; Zbozinek et al., 2012; Afzali et al., 2017). Recent studies have started using clustering approaches with the aim to replace diagnostic categories with newly discovered symptom clusters for the purpose of disease classification or prediction (Grisanzio et al., 2017; Chekroud et al., 2017a). Best results may be obtained by combining the two approaches in CP, by extracting relevant parameters from theory-driven models and plugging them into data-driven models to increase their predictive power. This in principle offers predictive power far beyond previous approaches.

The method developed in the first part of this work is purely data-driven and predicts disease trajectories by proposing that each trajectory is a combination of multiple latent influence trajectories, where each latent trajectory describes how the influence of a covariate changes in time. Here, the term latent is used to describe that the influence trajectories are not observed. The method is applied to the problem of predicting a clinical marker of depression for classification of patients into those that remit (or relapse) and those that do not.

The model developed in the second part of this work belongs to the category of generative data-driven models and describes an inference scheme for the metadecision making / planning problem. Consecutive decisions can be seen to span a decision-tree which is being searched during planning. How to solve the metadecision problem in an optimal way is very tough, as the number of subtrees that can potentially be evaluated is enormous, and good heuristics are difficult to come by. We apply the method to data collected from a behavioral sequential decision-making experiment. The application is again motivated by clinical considerations: Decision-making is fundamentally of an affective nature, because affection determines valuation which underlies the relative weighting of choices.

Models that describe the decision process (and its failures) better might be able to extract meaningful variables that can be used to get improved prediction / classification from mechanistically agnostic models.

1.1 A short note on notation

In this work, we specify scalars x in lower-case italics, and constants C in upper-case italics. Vectors $\boldsymbol{x}$ are denoted bold and lower-case and are always column vectors; matrices $\boldsymbol{M}$ are bold and upper-case. Indices to matrices follow the standard convention, i.e., the first index refers to the row and the second index refers to the column. For example, $\boldsymbol{M}_{ij}$ refers to the i-th row and j-th column of $\boldsymbol{M}$. Further, we specify model parameters as greek letters, e.g. α, β, γ, and so on.

Throughout, we define constants N, M, T to be the total number of individuals, features and time points. We use n, m, t as integer indices for enumeration, such that $n \in [1, N]$, $m \in [1, M]$, and $t \in [1, T]$. Elements of vectors are written in square brackets; for example $\boldsymbol{x} = [x_1, x_2, ..., x_N]^\top$, for a vector of length N. The T-like superscript denotes the transpose. This notation might cause slight confusion as square brackets can both denote an interval as well as a vector, but we believe that the correct interpretation is always clear from the context.

2 Trajectory prediction

Panel or longitudinal data occur when measurements on a set of individual units (in psychological studies mostly study participants) are repeated over multiple occasions (Hox et al., 2010). Such data is also referred to as repeated measures data, while cross-sectional data refers to data that is collected at only one particular occasion. Longitudinal data is hierarchical in nature, where the first level captures relationships between multiple observations made over time for each participant, while the second level captures variation amongst participants. From this perspective, such data can be seen as a generalization of time series data, with one time series for each participant.

Longitudinal data are important. For instance, in economic applications (Rossi et al., 2006), questions regarding the participation or brand choice of individual customers might be of interest. In clinical trials, the goal is often to compare treatment effects, with each patient typically assigned to one of a number of possible treatments, and measures taken during a certain follow-up period; these measures could be disease progression or trends of health status. Most often, interest lies in studying the aggregated change over time in populations of individuals, with the aim to find significant differences between the populations regarding a particular question of interest, such as "is one particular treatment more effective than another on average?". Models that are used to answer questions such as this are the classic repeated measures analysis of variance or its generalization, multivariate analysis of variance. In this work, we are instead interested in individual differences, in order to formulate questions such as "is a particular treatment more effective for a given individual patient?".

Panel models often study the association between independent variables (predictive variables, covariates, features) that are collected at the beginning of a study, and dependent variables (response variables, observations, targets), which could simply be the repeated measures themselves. The association between covariates and response variables could be time-varying, as could be the covariates themselves. When considering the repeated measures as samples from a continuous underlying process (which we aim to characterize by the model), they can be seen as noisy measurements of a trajectory. Our interest lies not only in constructing models for estimation of the characteristics of such a longitudinal process, but also in predicting how the process will evolve for new data that was not used to estimate the model.

One often expects positive covariation between observations that are close to each other. In modeling the underlying process, such expectations can be implemented by choosing structured priors for the temporal process which explicitly recognize proximity. For example, time series forecasts place higher weight on recent observations, assuming an autoregressive process. In spatial applications, a frequent goal is interpolation of a modeled surface based on proximity between observed locations (Gotway and Wolfinger, 2003). Exactly how this is done depends on the particular approach, and is indeed one of the main differences between methods.

First, we describe the different approaches to longitudinal data analysis. Then we develop a new method that allows for estimation and prediction of individual trajectories that arise as a linear combination of many latent (i.e. unobserved) trajectories. Finally, we apply the method to the important clinical problem of predicting longitudinal outcomes in major depressive disorder.

2.1 Survey of methods

In this section, we describe the different approaches to the analysis of longitudinal data; each of these consists of a variety of specific methods which can be classified in a number of ways. On the broadest level, the distinction is between parametric and non-parametric models, as well as between Bayesian and frequentist models.

The latter distinction specifies two principled ways in which parameters can be treated when performing statistical inference (but most modern frameworks seem to blend the two in some way). In either case one specifies a distribution for the observed data $\boldsymbol{x}$ given parameters $\boldsymbol{\beta}$. Viewed as a function of the parameters this distribution is termed the likelihood function. Those parameters we wish to make inferential statements about are distinguished from so-called nuisance parameters, which solely exist for absorbing confounding effects that are not of interest but must be accounted for to explain the data and obtain veridical estimates of parameters of interest. The likelihood is common to both the Bayesian and frequentist viewpoint, but the parameters of interest are interpreted in different ways. From a frequentist perspective they are fixed but unknown quantities, while from a Bayesian viewpoint they are random variables. This in turn leads to specification of priors which determine the proposed distribution of the parameters. In this context, Bayes' theorem (Bayes and Price, 1763) describes parameter inference as how our belief about the parameters should rationally change to account for available evidence. This is specified as computation of the posterior distribution of parameters based on prior (belief) and likelihood,

$$\underbrace{p(\boldsymbol{\beta}|\boldsymbol{x},\boldsymbol{\theta})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{\beta}|\boldsymbol{\theta})}^{\text{prior}}\overbrace{p(\boldsymbol{x}|\boldsymbol{\beta},\boldsymbol{\theta})}^{\text{likelihood}}}{\int p(\boldsymbol{\beta}|\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\beta},\boldsymbol{\theta})d\boldsymbol{\beta}} = \frac{p(\boldsymbol{\beta}|\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\beta},\boldsymbol{\theta})}{\underbrace{p(\boldsymbol{x}|\boldsymbol{\theta})}_{\text{marginal likelihood}}} = \frac{\overbrace{p(\boldsymbol{x},\boldsymbol{\beta}|\boldsymbol{\theta})}^{\text{joint}}}{p(\boldsymbol{x}|\boldsymbol{\theta})}. \tag{1}$$

The product of prior and likelihood is known as the joint distribution of data and parameters, or simply as the joint. The marginal likelihood, also known as model evidence, is the integral of the product of prior and likelihood, or simply the integral of the joint. The prior distribution of parameters can itself be parametrized by a set of hyper-parameters $\boldsymbol{\theta}$. It is worth pointing out here that the marginal likelihood is independent of $\boldsymbol{\beta}$; this makes inference convenient, because when optimizing the posterior with respect to $\boldsymbol{\beta}$, we can ignore the marginal likelihood and simply optimize the joint. Finally, predictions for new values x_* are obtained from the posterior predictive distribution,

$$p(x_*|\boldsymbol{x},\boldsymbol{\theta}) = \int p(\boldsymbol{\beta}|\boldsymbol{x},\boldsymbol{\theta})p(x_*|\boldsymbol{\beta},\boldsymbol{\theta})d\boldsymbol{\beta}. \tag{2}$$

The integral essentially evaluates predictions for all possible parameter values, weighing them by how likely they are according to the posterior distribution. In contrast to classic frequentist methods, this explicitly takes into account uncertainty about $\boldsymbol{\beta}$.

While the choice of prior can seem rather arbitrary in cases where not much is known about the parameters and their distributions, it should be noted that this can simply be reflected by the particular form of the distribution; for instance, one may chose a Gaussian distribution with a wide variance if knowledge is uncertain. In cases when no informed guess about the prior can be made at all, a non-informative prior might be chosen, such as the Jeffrey's prior (Jeffreys, 1946). It should be noted that uniform priors are not uninformative, as they are not uniform under reparametrization (Gelman et al., 2004). Finally, the posterior distribution of the parameters given data can be computed from prior and likelihood, which is defined as the probability of data given model parameters.

In the case where something is already known about the parameters, it can be a tremendous advantage to be able to embody this knowledge into the prior (Bernardo and Smith, 2008). While Bayesian methods are usually much more computationally intensive, modern methodological advancements and increased computational power have enabled full Bayesian inference for some problems. Furthermore, often it is possible to choose a conjugate form of the prior that combines with its corresponding likelihood such that the posterior can be calculated analytically. Additional advantages of Bayesian methods include that answers are easily interpretable and new observations can readily be incorporated. Inferences are conditional on the data, which in turn means that methods provide exact answers that do not rely on asymptotic arguments. For more detailed discussion of the advantages and disadvantages of Bayesian statistics, see for example Wasserman (2013).

Parametric models for longitudinal data include generalized linear models, and mixed effects / hierarchical linear models. These all assume a finite set of parameters, which capture all non-random structure and hence function as sufficient statistics. As such, future predictions are independent of past observed data after conditioning on the parameters (Murphy, 2012). While it might seem attractive at first glance that the model encapsulates all information, the complexity of these models is bounded by their parametrization, and often it turns out that the model is either not flexible enough, or too complex. In other words the amount of information that a parametric model can capture is independent of the amount of data that is available.

This limitation is overcome by non-parametric models. They assume that the data distribution cannot be defined in terms of a finite set of parameters (Russell and Norvig, 2010). Instead, a function is employed which implicitly defines an infinite number of parameters. This function may be further constrained by another set of parameters, which in this context we refer to as hyper-parameters in order to prevent confusion. Although in principle it is possible to model observations entirely non-parametrically, as it happens when choosing a function which is not further constrained, such an approach is often impractical. This is because the amount of data samples required for reliable estimation rises very quickly with the dimensionality of the samples (the principle is known as curse of dimensionality; see Bellman (1961)). Moreover, numerical results obtained from fitting high-dimensional nonparametric models can be difficult to interpret.

These problems motivate the consideration of nonparametric models that have meaningful as well as mathematically tractable structures that are constrained by a low-dimensional vector of hyper-parameters. The key advantage of non-parametric models is that the amount of information that they capture about the data grows as the amount of data grows. As a result, they combine flexibility with simplicity, and work well both for small and large data sets. However, this comes at the price of computationally challenging model estimation and inference procedures.

For longitudinal data, the most widely used methods are the Hierarchical Linear Model (HLM; see (Harville, 1977; Laird and Ware, 1982)) and Structural Equation Modeling (SEM; see (Chou et al., 1998; Skrondal and Rabe-Hesketh, 2007)). We give here a short overview of these and related methods, and describe each of them in more detail in subsequent section. The simplest HLM includes time points as covariates and observations as dependent variables. This is known as growth curve modeling. Often, further covariates are included that are hypothesized to explain both intra-individual and inter-individual variation. Time effects are modeled either as fixed effects of the covariates representing time, or as random effects at the first level, while inter-individual differences are modeled at the second level. The HLM enables explicit estimation of how parameters change in time, but prediction for out of sample data remains a problem. SEM can be viewed as a further generalization of the HLM: longitudinal observations are modelled as a linear combination of multiple underlying latent growth curves, whose functional shape is assumed to be known. A major limitation of SEM (and all its special cases) is that as a family of parametric methods the number of variables is necessarily finite, and must be small for most practical purposes, so that the method is not very flexible in comparison to non-parametric approaches. Another consequence is that only discrete-time models can be expressed (which require that the interval between any two successive measurements is the same), although time is naturally continuous.

One example of continuous-time modeling is Gaussian Process Regression (GPR; see (Rasmussen and Williams, 2006)), a non-parametric method that specifies priors for mean and covariance of time series data, and can be extended to longitudinal data. Covariates can be included in a number of different ways, but interpretation is not straight-forward. Varying Coefficient (VC; see (Hastie and Tibshirani, 1993)) models instead specify linear effects of covariates (just like the HLM), but let these effects change in time by explicitly modeling time-varying regression coefficients. One way of doing this is to assume Gaussian Process priors over the regression coefficients. As our method combines GPR and VC, we defer discussion of these methods to section 3.

Finally, Survival Analysis (Lee et al., 2003) is an alternative approach to longitudinal data modeling with the aim of predicting time to event (failure, sickness, death, etc). The connection to all other models mentioned above is that one way such an event may be triggered is when the observed variable reaches a certain threshold.

Analysis of longitudinal data is, amongst other reasons, complicated by the

fact that subjects might have been measured at different times, and the total amount of observations might differ between them. More generally, there often are missing observations. An important question relevant to model estimation is whether such omissions are random or if some structure can be discerned (such as dependence on the response that would otherwise have been observed). We discuss advantages and disadvantages for each model family with respect to these issues, as well as each method's constraints and computational efficiency.

2.2 General Linear Model

The General Linear Model is the most basic formulation connecting all approaches that we discuss in this section. Thus, we describe the model here in some detail, as well as introduce necessary mathematical concepts on the way.

The (Multiple) Linear Regression Model for longitudinal data defines a conditional distribution of the dependent variables / targets / responses / outcomes y_{nt} given column vectors of observations for the independent variables / regressors / features / covariates $\boldsymbol{x}_{nt}$ (Bishop, 2006), where $n \in [1, N]$ indicates individuals and $t \in [1, T]$ indicates time. It can be written as

$$\mathcal{Y}_{nt}|\boldsymbol{x}_{nt} \sim \{\mathcal{N}(\boldsymbol{x}_{nt}^{\top}\boldsymbol{\beta}, \sigma^2) : (\boldsymbol{\beta}, \sigma) \in \mathbb{R}^K \times \mathbb{R}\}, \tag{3}$$

where $\mathcal{Y}_{nt}$ is a random variable whose realizations are y_{nt}. Each $\boldsymbol{x}_{nt}$ is a column vector of K covariates. Further, let $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma)$ denote the parameters to be estimated, $\boldsymbol{\beta}$ the vector of regression coefficients and σ^2 the variance around the mean given by $\boldsymbol{x}_{nt}^{\top}\boldsymbol{\beta}$. The use of the term *multiple* refers to the fact that the model supports more than one covariate. Given observed data $\mathcal{D} = \{\boldsymbol{y}, \boldsymbol{X}\}$ with $\boldsymbol{y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_N]^{\top}$, $\boldsymbol{y}_n = [y_{n1}, ..., y_{nT}]^{\top}$, the matrix $\boldsymbol{X}$, which is also called the design matrix, is defined as $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_N]^{\top}$ and $\boldsymbol{X}_n = [\boldsymbol{x}_{n1}, ..., \boldsymbol{x}_{nT}]^{\top}$ with dimensionality $NT \times K$. It contains feature column vectors for different participants stacked on top of each other, with columns corresponding to time points. Making the individual entries explicit, we can write

$$\boldsymbol{X} = \begin{bmatrix} (x_{11})_1 & \dots & (x_{11})_K \\ \vdots & & \vdots \\ (x_{1T})_1 & \dots & (x_{1T})_K \\ (x_{21})_1 & \dots & x_{21K} \\ \vdots & & \vdots \\ (x_{2T})_1 & \dots & (x_{2T})_K \\ \vdots & & \vdots \\ (x_{N1})_1 & \dots & (x_{N1})_K \\ \vdots & & \vdots \\ (x_{NT})_1 & \dots & (x_{NT})_K \end{bmatrix}, \tag{4}$$

where the first, second and third subscript to x refer to participants, time points, and features, respectively. Note that each $\boldsymbol{y}_n$ is itself a vector containing all

repeated measures of individual n. The corresponding statistical model is

$$\boldsymbol{\mathcal{Y}}|\boldsymbol{X} \sim \{\mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}_y) : (\boldsymbol{\beta}, \sigma) \in \mathbb{R}^K \times \mathbb{R}\}. \tag{5}$$

The spherical shape of the covariance $\boldsymbol{\Sigma}_y = \boldsymbol{\Sigma}_e = \sigma^2\boldsymbol{I}$ implies that the observations are assumed to be independent and identically distributed conditional on their underlying distributional form; such sequences of random variables are also called exchangeable. Covariance patterns are ignored by design, i.e. they cannot be estimated. Also usually the model is introduced in this way, it is not necessary in general to restrict the covariance $\boldsymbol{\Sigma}_y$ to be spherical. However, in most cases a constrained functional shape has to be imposed in order to be able to estimate it. For example, one may decompose it into a linear sum of fixed shapes $\boldsymbol{Q}_i$, whose weights λ_i are being estimated (Friston et al., 2002),

$$\boldsymbol{\Sigma}_y = \sum_i \lambda_i \boldsymbol{Q}_i. \tag{6}$$

The multiple regression model specifies the relationship

$$y_{nt} = \boldsymbol{x}_{nt}^\top \boldsymbol{\beta} + \epsilon_{nt}, \tag{7}$$

with residuals ϵ_{nt} being the single source of random variation. This is the formulation that best facilitates comparisons with other approaches; we will refer back to it whenever appropriate.

While there is no universal agreement on the difference between multiple regression and what is known as the General Linear Model, the latter is usually thought of as an extension of the multiple regression model to multivariate observations and categorical predictor variables (Mardia et al., 1979). As such, it specifies the relationship

$$\boldsymbol{Y} = \boldsymbol{X} \cdot \boldsymbol{B} + \boldsymbol{E}. \tag{8}$$

It should be noted that the only difference between this formulation and equation 7 is the dimensionality of $\boldsymbol{Y}, \boldsymbol{B}$ and $\boldsymbol{E}$. Since it is possible to reshape matrices into vectors by simply stacking columns on top of each other, the two formulations are effectively equivalent for all purposes except for ease of interpretation.

Effects of time are modeled in exactly the same way as effects between participants: The only way to incorporate them in the model is through the covariates. This is a case of observed heterogeneity: Differences are explained by independent variables. All other heterogeneity has to be accounted for by the residuals; this is also known as unobserved heterogeneity.

The model can be modified in a number of ways. By letting covariates be proposed functions of time (instead of repeated measures) we arrive at an approach known as growth models, which we discuss in see section 2.5. More generally, the design matrix can contain any number of functions of covariates, which essentially allows to model non-linear effects w.r.t. the original covariates. Kernel methods employ a mathematical trick to be able to estimate non-parametric

models that effectively represent an infinite set of such basis functions. Gaussian Process regression, discussed in section 3.1.1, is such an example. Varying Coefficient Models consider regression coefficients $\boldsymbol{\beta}$ as smooth functions of time (see section 3.1.2). Finally, the model could be extended such that regression coefficients are considered to be random effects of time; this leads to the Hierarchical Linear Model, which is discussed in section 2.3.

One way in which the General Linear Model might violate the assumption of independent residuals is that errors might be correlated in time. An example of how this can be modeled is an autoregressive AR(1) model, where the observations / residuals at time t depend on the observations / residuals at time $t-1$. Let ρ denote the correlation between two consecutive observations. Then the corresponding covariance matrix for the whole data set $\boldsymbol{X}$ becomes

$$\boldsymbol{\Sigma_y} = \begin{bmatrix} \boldsymbol{\Sigma_{y_1}} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma_{y_2}} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{\Sigma_{y_{N-1}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Sigma_{y_N}} \end{bmatrix}, \quad \boldsymbol{\Sigma_{y_n}} = \sigma^2 \begin{bmatrix} 1 & \rho & \dots & \rho^{T-2} & \rho^{T-1} \\ \rho & 1 & \dots & \rho^{T-3} & \rho^{T-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho^{T-2} & \rho^{T-3} & \dots & 1 & \rho \\ \rho^{T-1} & \rho^{T-2} & \dots & \rho & 1 \end{bmatrix}. \tag{9}$$

Such a covariance-matrix is called block-diagonal. The approach can easily be extended even further: A Markov model extends the AR(1) model to unequally spaced time points; powers of ρ in $\boldsymbol{\Sigma_{y_n}}$ are then taken to be distances between time points.

The Generalized Linear Model (GLM) extends the General Linear Model to model non-linear dependence of $\boldsymbol{y}$ on $\boldsymbol{X}$ by using a link function $g(\cdot)$, such that

$$\boldsymbol{y} = g^{-1}(\boldsymbol{X\beta} + \boldsymbol{\epsilon}). \tag{10}$$

This relaxes the rather restrictive assumption that the residuals are Gaussian; instead they can be distributed according to any distribution from the exponential family, such as the Poisson or Binomial distributions. For clarity of presentation, we focus on the linear models corresponding to normally distributed residuals, keeping in mind that they can always be extended to the non-linear case.

Most parametric hypothesis tests as employed in psychological research (like the t-test and ANOVA) are specializations of the GLM, where the design matrix $\boldsymbol{X}$ consists of zeros and ones in the appropriate places, to check for proposed changes in some of the parameters. If all entries in $\boldsymbol{X}$ are either 0 or 1, the model corresponds to a (Multivariate) Analysis of Variance (MANOVA or ANOVA), see section 2.4. Allowing entries in some of the columns of $\boldsymbol{X}$ to be real numbers (corresponding to actual feature values) and entries in other columns to be either 0 or 1 corresponds to MANCOVA (or ANCOVA), where C is the abbreviation for covariates.

When used for the purpose of supervised learning (which enables prediction of dependent variables given previously unseen samples from the independent variables) the frequentist approach is to obtain a point estimate for regression

coefficients $\hat{\boldsymbol{\beta}}$. Assuming independent residuals, the standard approach is to compute the maximum likelihood estimator (MLE)

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \tag{11}$$

such that the prediction function becomes $\hat{f}(\boldsymbol{x}) = \boldsymbol{x}^\top \hat{\boldsymbol{\beta}} = \hat{\boldsymbol{y}}$. In case of the General Linear Model, the MLE coincides with the mean squared error. For the case of correlated errors, whose covariance might be of the form defined in equation 6, more advanced techniques need to be employed in order to estimate the coefficients, such as the Expectation-Maximization algorithm (Dempster et al., 1977).

In a Bayesian context, the full posterior distribution of $\boldsymbol{\beta}$ is either analytically computed or approximated. The prediction function becomes an integral of the product of posterior and likelihood, where the integral is computed over all possible values of $\boldsymbol{\beta}$; this is called the posterior predictive distribution (see Bishop (2006) for an extensive discussion).

The GLM is too simple for most longitudinal data analysis settings, as no subject-specific effects are modeled. However, as detailed above, it provides a good starting point for exploration of other more involved methods.

2.3 Hierarchical Linear Model

One extension of the GLM is the Hierarchical Linear Model (HLM). This is a very general model, and has been investigated in many different fields. Hence, it is known under a variety of names, such as multilevel model, linear mixed model, random coefficient model or covariance component model (Wishart, 1938; Box, 1950; Rao, 1958; Harville, 1977; Laird and Ware, 1982). It is built upon the idea that regression coefficients are allowed to vary randomly (level 1) across another entity of interest (level 2). These random variations are called *random effects*. Coefficients that are not allowed to vary are known as *fixed effects*. Models including both random and fixed effects are called mixed (effects) models. For longitudinal data the first level models repeated measures of one individual, and the second level inter-individual differences such that

$$y_{nt} = \boldsymbol{x}_{nt}^\top \boldsymbol{\beta} + \boldsymbol{z}_{nt}^\top \boldsymbol{\gamma}_n + \epsilon_{nt}. \tag{12}$$

While $\boldsymbol{x}_{nt}$, as before, contains features that account for fixed effects across time (these are now effects on the second level), we have introduced a second set of features $\boldsymbol{z}_{nt}$, which accounts for random effects, meaning that these effects change on the first level. The random effects $\boldsymbol{\gamma}_n$ themselves are usually constrained by the assumption that they all follow the same distribution $\boldsymbol{\gamma}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{D})$. The residuals are assumed to be correlated on the first level, but independent on the second: $\boldsymbol{\epsilon}_n = [\epsilon_{n1}, ..., \epsilon_{nT}]^\top \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_n)$, such that the resulting covariance matrix for the whole vector $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_N]^\top$ is block-diagonal.

We can slightly rewrite the model to make its hierarchical nature more explicit,

$$y_{nt} = \boldsymbol{z}_{nt}^\top \boldsymbol{\beta}_n + \epsilon_{nt} \tag{13}$$
$$\beta_{nk} = \boldsymbol{a}_{nk}^\top \boldsymbol{\beta} + \gamma_{nk}. \tag{14}$$

By substituting the second line into the first and letting $\boldsymbol{A}_n = [\boldsymbol{a}_{n1}^\top, ..., \boldsymbol{a}_{nK}^\top]$ and $\boldsymbol{X}_n = \boldsymbol{Z}_n \boldsymbol{A}_n$, we obtain equation 12. The model implies

$$\boldsymbol{\Sigma}_{\boldsymbol{y}_n} = \boldsymbol{Z}_n \boldsymbol{D} \boldsymbol{Z}_n^\top + \boldsymbol{R}_n. \tag{15}$$

The random effects effectively constrain the shape of the covariance matrix. Indeed, any positive intraclass correlation in linear regression can be modeled as random effects by augmenting the regression with intraclass variables, i.e. variables that are one if a measurement belongs to the given class and zero otherwise (Gelman et al., 2004). For example, we might have only one parameter which is proposed to differ between groups. Let

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}^\top \boldsymbol{\beta}, \sigma^2 \boldsymbol{I}) \quad \text{and} \quad \boldsymbol{\beta} \sim \mathcal{N}(\mathbf{1}\alpha, \sigma_\beta^2 \boldsymbol{I}), \tag{16}$$

where $\boldsymbol{x}$ is a vector of zeros and ones indicating group membership and $\mathbf{1}$ is a vector of ones. By deriving the marginal likelihood of $\boldsymbol{y}$, averaging over $\boldsymbol{\beta}$ (Gelman et al., 2004), this can be shown to be equivalent to

$$\boldsymbol{y} \sim \mathcal{N}(\mathbf{1}\alpha, \boldsymbol{\Sigma}_y), \quad \text{where} \quad (\boldsymbol{\Sigma}_y)_{ij} = \begin{cases} \sigma^2 + \sigma_\beta^2 & \text{if } i = j \\ \sigma_\beta^2 & \text{if } i, j \text{ are in same group} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

From the Bayesian perspective, fixed effects are components of $\boldsymbol{\beta}$ that are assigned priors with infinite prior variance.

The HLM can easily be extended to more than two levels. If subjects can be assigned to different groups, a three level HLM may be more appropriate. Alternatively, group membership may be indicated by adding columns in the design matrix whose entries equal one if a subject belongs to a certain group and zero otherwise.

HLMs are usually employed as a frequentist method. From a Bayesian perspective the first level could be thought of as providing a prior for the regression coefficients on the second level. The typical specification of a Bayesian hierarchical model would add one more level to include additional priors on $\boldsymbol{\gamma}, \boldsymbol{D}$ and $\boldsymbol{R}_n$. With respect to inference, the introduction of hierarchy, which couples parameters between levels, leads to two effects on the parameters: First, information flows up the hierarchy to inform inferences about variables in higher levels. This effect is known as 'pooling'. Second, estimates at lower levels are shrunk towards the mean estimates at higher levels, an effect appropriately called 'shrinkage'. Bayesian parameter estimation can be done via likelihood-based approaches, such as restricted ML (Kenward and Roger, 1997), or the Expectation-Maximization algorithm (Dempster et al., 1977). Such inference schemes support

missing covariates, so that individuals for which not all covariates are available do not need to be excluded.

If all priors are known, a fully Bayesian scheme can be employed. Otherwise, the unknown priors must be estimated from the data; this approach is known as empirical Bayes (Friston et al., 2002). Usually, this proceeds in an iterative way where the parameters are estimated with fixed priors, and then the priors are estimated with fixed parameters. This process repeats until convergence. Philosophically, the empirical Bayes approach has often been criticized, because the prior should reflect our assumptions about the data, and thus should not be informed by the data. However, practically, it works very well.

Characterization of individual trajectories is possible by using such an empirical Bayes estimator for the random effects, which essentially evaluates conditional expectations $\mathbb{E}[\boldsymbol{\gamma}_n|\boldsymbol{y}_n]$(Reinsel, 1984). The expectation can also be evaluated only over one part of $\boldsymbol{y}_n$, enabling prediction of the remaining, and even further future values. Uncertainty in the estimates obviously grows if less observations are available; if no observations are observed, the random effects estimated via classical frequentist methods are the same for each individual. While the Bayesian perspective would allow simulation of $\boldsymbol{\gamma}_*$ from the posterior predictive density, individual differences would be random instead of systematic. Thus, predicting both $\boldsymbol{\gamma}_*$ and $\boldsymbol{y}_*$ from $\boldsymbol{X}_*, \boldsymbol{Z}_*$ for new individuals is not possible.

In contrast, in Varying Coefficient Models (see section 3.1.2), the time-varying change in regression weights is explicitly modeled, so prediction can be done without complications.

In summary, the multilevel approach to panel data is flexible enough to handle incomplete data, and has the ability to model covariation between observations in a flexible way. However, while characterization of individual trajectories is possible, the certainty of prediction for new individuals that have not been used to estimate the model parameters depends on the number of observations; without any observations, no predictions can be made.

2.4 Analysis of Variance

Before proceeding to discuss modern approaches to longitudinal data analysis, we complete the survey of classic approaches by discussing the principle of analysis of variance (ANOVA). The idea is to group predictor variables and their coefficients into batches, where each batch corresponds to a separate source of variation; all coefficients within a batch are assumed to be exchangeable. Usually, a batch corresponds to an experimental block, a factor or an interaction of factors. Phrased in the framework of HLMs, methods based on this principle were the first approaches to panel data analysis (Fisher, 1925). The usual setup is slightly different in that the focus is on testing hypotheses between differences of groups of individuals. For example, one group might consist of subjects who were given a certain medication while another group might consist of subjects who received placebo. Generally, testing the group by time interaction is of primary interest, as it addresses whether the change in mean response differs across

groups. Another question of interest may be whether the mean response is in fact constant over time.

There are two principled approaches for applications to longitudinal data: Repeated Measures Univariate Analysis of Variance (rmANOVA) and Multivariate Analysis of Variance (MANOVA). The term univariate here means that a separate model is considered for each individual observation.

2.4.1 Repeated Measures Univariate Analysis of Variance

Repeated Measures Univariate Analysis of Variance (also known as mixed effects ANOVA with one random effect) is formally a special case of the HLM and can be expressed through the relationship

$$y_{ngt} = \mu + \gamma_g + \tau_t + \zeta_{gt} + \pi_{ng} + \epsilon_{ngt}, \tag{18}$$

where the new index g denotes group membership, μ is the grand mean, γ_g the effect of group g, τ_t the effect of time t, ζ_{gt} the interaction of time t with group g, π_{ng} the individual difference component for subject n in group g and ϵ_{ngt} the residual. The residuals are often assumed to be normally distributed, $\epsilon_{ngt} \sim \mathcal{N}(0, \sigma_\epsilon^2)$, as are the individual difference components for a group, $\pi_{ng} \sim \mathcal{N}(0, \sigma_\pi^2)$. In other words, subjects are random, but group and time fixed. While models of this type are relatively easy to deal with, they are too restrictive for many problems of interest (Tabachnick and Fidell, 2012): The implied structure for the covariance of the observations is known as *compound symmetry*, and is only justified if the within-subject factor (in our application this is time) is randomly allocated; but this is most likely not the case for most longitudinal data.

However, just as in the GLM, it is possible to model more complicated covariance effects by imposing a structure on the covariance matrix, such as described in equation 6. The main advantage of rmANOVA in comparison to MANOVA is that due to its relative simplicity the model supports powerful and accurate tests of group trends.

2.4.2 Multivariate Analysis of Variance

A related approach is the Multivariate Analysis of Variance (MANOVA), which considers the data in form of vectors $\boldsymbol{y}_{ng}$ (hence multivariate). The model can be written as

$$\boldsymbol{y}_{ng} = \boldsymbol{\mu} + \boldsymbol{\gamma}_g + \boldsymbol{e}_{ng}, \tag{19}$$

where all vectors are of dimension $T \times 1$, $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\gamma}_g$ the vector effect for the population from which the g-th group was drawn, and $\boldsymbol{e}_{ng} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_e)$ the vector of residuals.

As a consequence of the multivariate approach, MANOVA hypotheses are about differences of mean *vectors* for each group. A rejected null means that at

least one of the mean vectors differs in at least one component. This is not a very informative measure. Worse, lots of comparisons need to be taken into account so it has the tendency to be underpowered, meaning that such departures from the null hypothesis are difficult to detect. As in the univariate case, incorporation of covariates is possible; the corresponding method is called Multivariate Analysis of Covariance (MANCOVA).

MANOVA and rANOVA can be extended in different ways, but all approaches suffer from the limitations that the dataset needs to be balanced (all subjects must have the same number of measurements, which must occur at the same time points) and complete (no missing data). As the focus of the analysis is on hypothesis testing, these models cannot be used for prediction. Indeed, subject-specific trends cannot even be inferred.

2.5 Growth Models

Growth curve analysis is used to obtain a description of the mean growth in a population over a specific period of time (Box, 1950; Rao, 1958). The model on which growth curve analysis is based, the growth curve model, can be approached from several perspectives. On the one hand, the model can be constructed as a standard two-level multilevel regression model by treating time as a random effect (Hox and Stoel, 2005; Raudenbush and Bryk, 2002). On the other hand, the model can be constructed as a structural equation model (SEM), which incorporates time as latent factors whose relative loadings are to be estimated. (McArdle, 1988).

2.5.1 Multilevel Growth Models

In multilevel growth modeling (MGM), an overall mean change function (e.g. linear, quadratic, cubic etc.) is fitted to the whole data set around which individual's trajectories are allowed to vary in a random fashion. While in principle any functional shape in time is supported, most often, only the slope and intercept may be allowed to vary across individuals as random effects (Hoffman and Rovine, 2007).

Multilevel Growth Models are formally equivalent to HLMs. The difference is that columns of the design matrix are constructed to consist of functions of time (but other covariates can be included, too). For example, $[1, 2, 3, ..., T]^\top$ can be used to model linear effects of time, or $[1, 2, 4, ..., T^2]^\top$ to model quadratic trends.

In the simplest linear growth model, each subject may have their own regression curve with the intercepts and slopes varying randomly by subject, around a common linear trajectory. In growth curve terminology, models that do not include any subject-specific covariates are known as unconditional models. Conditional models, on the other hand, can include time-varying covariates (TVC) or time invariant covariates (TIC) (Goldstein, 2011; Omar et al., 1999).

2.5.2 Latent Growth Curve Models

The Latent Growth Curve (LGC) approach adopts a different perspective by incorporating functions of time in a design matrix of latent variables that represent the shape of the growth curve (McArdle, 1988). In other words, observations are considered to be linear combinations of underlying latent growth curves.

LGC is an application of Structural equation modeling (SEM) to longitudinal data. There are different ways of specifying such a model (Chou et al., 1998; Skrondal and Rabe-Hesketh, 2007), which lead to subtle differences (see e.g. Curran et al. (2010) for more details). One way of formalizing LGC analysis is to complement a linear measurement model of the same form as the HLM in equation 12 with a so-called *structural model*, which specifies causal links between latent factors (Raudenbush and Bryk, 2002). In statistics, latent factors are variables that are not observed themselves, and thus hidden; as we shall see, this corresponds to the implication of random effects.

The mean and covariance structure of the latent variables in LGC analysis correspond to the fixed and random effects in MGM, and this makes it possible to specify the same model as an LGC or MGM (Hox and Stoel, 2005). In fact, it can be shown that under many broad conditions SEM is analytically equivalent to a Hierarchical Linear Model (Willet and Sayer, 1994; Raudenbush and Bryk, 2002; Curran, 2003). However, it should be noted that the two approaches optimize different objectives. While the objective of the HLM is to minimize the error between predictions $\hat{\boldsymbol{y}}$ and observations $\boldsymbol{y}$, SEM seeks to minimize the error between predicted covariance $\hat{\boldsymbol{\Sigma}}_y$ and sample covariance $\boldsymbol{\Sigma}_y$.

The measurement and structural components of the LGC model for longitudinal data can be written as

$$\begin{aligned} y_{nt} &= \boldsymbol{x}_{nt}^\top \boldsymbol{\beta} + \boldsymbol{\lambda}_t^\top \boldsymbol{\gamma}_n + \epsilon_{nt} \\ \gamma_{nm} &= \boldsymbol{z}_{nm}^\top \boldsymbol{\alpha} + \sum_{k=1}^{M} b_{mk}\gamma_{nk} + \zeta_{nm}, \end{aligned} \tag{20}$$

where $\boldsymbol{x}_{nm}$ and $\boldsymbol{z}_{nm}$ are vectors of covariates, m indicates the m-th of M latent factors, and $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ the corresponding coefficients. The residuals are considered to be independent across subjects, but unconstrained otherwise: $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ and $\boldsymbol{\zeta}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta})$. The vector $\boldsymbol{\lambda}_t$ contains the values at time t of all latent growth trajectories. Subject-specific latent variable terms $\boldsymbol{\gamma}_n$ describe the amount to which each of the latent growth trajectories specified as columns in the loading matrix $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_T]^\top$ contributes to the individual's observed trajectory. Typically, $\boldsymbol{\Lambda}$ consists only of two columns, specifying constant and linear trends in time. In other words, time enters the model through $\boldsymbol{\Lambda}$ in the same way as it enters the MGM through $\boldsymbol{X}$. Addition of the structural model effectively allows for two effects: Additional covariates $\boldsymbol{z}_{nt}$ may influence the latent variable terms, and these terms may influence each other.

Note that sometimes the special case without $\boldsymbol{x}_{nt}$ and no structural model ($y_{nt} = \boldsymbol{\lambda}_t^\top \boldsymbol{\gamma}_n + \epsilon_{nt}$) is termed a latent growth model (Preacher, 2008). This special case is fully equivalent to a multilevel growth model.

The functional shape of growth curves is usually assumed to be known and fixed. However, depending on the sample size some of them can be parametrized, and consequently estimated. If that is the case, the problem is essentially a factor analytic one, suffering from the same mathematical identification problems.

Covariates can be incorporated in two ways: either through $\boldsymbol{x}_{nt}$, where they directly influence the observations, just like in the HLM. Or through $\boldsymbol{z}_{nm}$, thereby affecting the latent terms in such a way that they can be thought of as causing individual-specific variations in the latent growth curves specified in $\boldsymbol{\Lambda}$. Additional weights b_{mk} describe the amount to which latent terms influence each other.

Classic structural equation modeling is a frequentist framework, and estimation is done via maximum likelihood approaches. However, Bayesian variants exist, that scale better to small sample sizes (Lee, 2007). While modern extensions to growth curve modeling can deal with missing data, unequally spaced time points and complex trajectories (Curran et al. (2010)), the functional shape of the trajectories must still be specified parametrically in $\boldsymbol{\Lambda}$. To estimate complex temporal courses would require temporal basis sets with many parameters.

Model specification in LGCM (and special cases like HLM, GLM, ANOVA) consists of formulating a set of linear equations that describe the hypothesized relationships between a finite set of variables parametrically. Accordingly, the number of parameters is necessarily finite, and so development over time cannot be modeled as a continuous-time process. In order to do that, non-parametric methods need to be employed, such as Gaussian Process Regression, which we introduce in section 3.1.1.

2.6 Survival Analysis

A different perspective on modeling longitudinal data is provided by survival analysis. Survival models predict time to an event, and are usually employed to compare the survival time of two populations or to test for significant risk factors affecting survival (Lee et al., 2003). Typically, event times are not observed for everyone, an effect which is known as right-censoring (Tobin, 1958), attrition (Schafer and Graham, 2002) or dropout (Hogan et al., 2004). However, they can also be used for predicting the event of individual survival. As such they yield a binary prediction (life or death), instead of whole time courses. The ability to deal with censoring of longitudinal data in a principled way by a frequentist method is unique to survival analysis (In the Bayesian context, one can marginalize over missing data). For completeness, we quickly review here parametric survival models, as well as the Cox Proportional Hazards model, which is the most famous method that is not fully parametric (to be precise, it is semi-parametric rather than non-parametric, as we shall see).

2.6.1 Parametric

Let $f(t)$ denote the probability density function of time. The cumulative density is given by $F(t) = \int_0^t f(u)du$ and the survival function by $S(t) = 1 - F(t)$.

Further, define the hazard rate h(t) as the rate of occurrence of an event

$$h(t) = \frac{f(t)}{S(t)} = -\frac{d}{dt}\log S(t) = e^{-\int_0^t h(u)du}. \tag{21}$$

In words, the hazard rate equals the density of events at a time point divided by the probability of surviving until that time. We note that $h(t), S(t)$ and $f(t)$ are coupled through the relations defined in equation 21, such that specifying one determines the other two.

The simplest hazard is constant in time, $h(t) = h$. This leads to an exponential survival, as the corresponding survival function and p.d.f. are $S(t) = \exp(-ht)$ and $f(t) = h\exp(-ht)$, respectively. However, this is too simple to be useful in most cases. Instead, often the two-parameter Weibull model, which is a generalization of the exponential model, is employed. It can be parametrized as

$$f(t) = \kappa\lambda t^{\kappa-1}e^{-\lambda t^\kappa} \qquad S(t) = e^{-\lambda t^\kappa} \qquad h(t) = \kappa\lambda t^{\kappa-1}. \tag{22}$$

Covariates could be included by letting either of the model parameters κ, λ depend on them; for example, we could let $\log\lambda = \boldsymbol{x}(t)_n^\top\boldsymbol{\beta}$. If covariates are somehow incorporated into the model, the hazard function can be split into two parts, where the first part, the baseline hazard $h_0(t)$, describes how the risk of event per time unit changes over time independently of the covariates. Model parameters can be estimated using likelihood-based approaches.

Parametric survival models necessarily prescribe a functional form for the proposed survival function, such that a trade-off always has to be made between flexibility / complexity of the model and how well it can be fit to the data. In any case, prior knowledge about the general shape of the survival curves is necessary to decide for any particular model. If such knowledge is not available, a semi-parametric approach should be considered.

2.6.2 Cox Proportional Hazards

The Cox Proportional Hazards model (Cox, 1972) is arguably the most well known semi-parametric survival model. In its most common form, the hazard function is defined as

$$h(t|\boldsymbol{x}_n) = h_0(t)e^{\boldsymbol{x}_n^\top\boldsymbol{\beta}}, \tag{23}$$

where $h_0(t)$ is the baseline hazard which reflects the underlying hazard for subjects with all covariates equal to zero. We observe that $\log\frac{h_n(t)}{h_0(t)} = \boldsymbol{x}_n^\top\boldsymbol{\beta}$ for individual n. This means that each individual's hazard function is proportional to the baseline hazard (and so all individual's hazards are proportional to each other; hence the name of the method).

There are several approaches to the analysis of such a model. The simplest is to assume $h_0(t)$ constant; this is equivalent to assuming an underlying exponential distribution. Alternatively, some parametrization might be chosen to be able

to fit a wider range of shapes, such as the Weibull distribution. Then, standard methods such as maximum likelihood can be used. Cox's model takes a different approach: Due to the particular form of equation 23, it is possible to allow that $h_0(t)$ be any arbitrary function of time.

Following Cox's model, the estimated hazard for individual n with covariate vector $\boldsymbol{x}_n$ has the form $\hat{h}_n(t) = \hat{h}_0(t)e^{\boldsymbol{x}_n^\top \hat{\boldsymbol{\beta}}}$, where $\hat{\beta}$ is found by maximizing the so-called pseudo-partial likelihood, while $\hat{h}_0(t)$ follows from the Breslow estimator. Suppose there are N distinct death times $t_1, ..., t_N$, assuming that there are no tied death times (but this assumption can be relaxed; see Hertz-Picciotto and Rockhill (1997)). Let $R(t) = \{i : x_i \geq t\}$ denote the risk set at time t, defined as the set of indices of subjects that are alive (and thus at risk) just before the observed time t. Then the parameters $\boldsymbol{\beta}$ can be estimated by maximizing the pseudo-partial likelihood (Cox, 1972)

$$\mathcal{L}^{\text{partial}}(\boldsymbol{\beta}) = \prod_{n=1}^{N} \frac{e^{\boldsymbol{x}_n^\top \boldsymbol{\beta}}}{\sum_{l \in R(t_k)} e^{\boldsymbol{x}_l^\top \boldsymbol{\beta}}}, \tag{24}$$

where the individual terms in the product are the conditional probabilities that a particular subject n would fail at t_n given risk set $R(t_n)$. This is a *partial* likelihood in the sense that it considers probabilities for censored subjects only implicitly via definition of the risk set. Cox provided a justification of $\mathcal{L}^{\text{partial}}$ as the part of the full likelihood that contains most of the information about $\boldsymbol{\beta}$ (Cox, 1975). Having obtained coefficient estimates $\hat{\boldsymbol{\beta}}$, the baseline hazard $h_0(t)$ can be approximated. The Breslow Estimator (Breslow, 1975) is the most frequent way of achieving this. It is defined as

$$\hat{h}_0(t_i) = \frac{d_i}{\sum_{j \in R_i} e^{\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}}}, \tag{25}$$

with d_i the number of deaths at t_i. The estimator is obtained as a maximum likelihood solution assuming that the baseline hazard is piecewise constant between failure times. For further details, see (Lin, 2007).

The model can be generalized to include time-varying covariates $\boldsymbol{x}_n^\top(t)$ as well as time-varying coefficients $\boldsymbol{\beta}(t)$; this is called the General Hazard Rate Model. The strengths of this model family is that it rests on a minimal set of assumptions and is thus quite robust. It is unique in the sense that coefficients $\boldsymbol{\beta}$ that determine inter-individual differences can be estimated without having to know the mean growth shape. In comparison to the parametric approach, we pay for the additional flexibility by only being able to predict individual survival via approximation of the baseline hazard, which requires a cumbersome two step procedure and somewhat defeats the very purpose of the model.

3 Gaussian process trajectory prediction

In this section we develop Gaussian Process Trajectory Prediction (GPTP), a Gaussian-process (GP) based approach to longitudinal data prediction (Renz et al., 2018). The observed longitudinal data is modeled as a linear combination of hidden feature weight trajectories multiplied with their respective feature values. Each latent trajectory describes how the influence of a feature changes in time.

Consider the example shown in Figure 1, which focuses on one of the most burdensome diseases world-wide: major depressive disorder. The number of previous depressive episodes a person has had might predict a slow re-emergence of depression over time (green line), while the current symptom severity might only predict depression to a certain degree for a briefer period and then no longer hold predictive value (red line). By performing a weighted sum of such contributions (Figure 1b), it might be possible to predict disease trajectories. By inferring the temporal contributions of individual baseline features, GPTPs allow for complex, non-linear disease trajectories and hence generalize currently existing approaches in clinical use.

Apart from the intuitive appeal, the approach has a number of additional advantages. The application of Gaussian Processes enables us to easily handle missing observations and covariates, as they can be integrated out using standard marginalization rules for Gaussians. The method is robust against even highly correlated covariates, and the most parsimonious covariates for prediction can be identified through the use of Automatic Relevance Determination (ARD). While GPTP can in principle handle time-varying covariates as well, it is useful to restrict covariates to be time-invariant, and to be measured at the beginning of the observation period, as this enables prediction of trajectories of future responses from covariates for out-of-sample participants. Finally, GPTP allows partial information about the marker trajectory to be included and conditioned upon and thereby allows for dynamical updates of predictions.

Note that the difference between this approach and standard GP regression (Rasmussen and Williams, 2006) is that the GP prior is placed on the regression weights, whereas in GP regression the prior is placed on the observations. As such, our model combines the Bayesian perspective (by using GP priors) with the idea of time-varying coefficients, as first proposed by Hastie and Tibshirani (1993).

Before specifying our model, we first introduce the concepts of Gaussian Process Regression and Varying Coefficient models, which we combine in GPTP.

3.1 Prerequisites

3.1.1 Gaussian Process Regression

In this section we present Gaussian Process Regression (GPR) as a non-parametric extension of the GLM. While GPR itself can be used to model longitudinal data by assuming that observations are realizations of a Gaussian Process, the re-

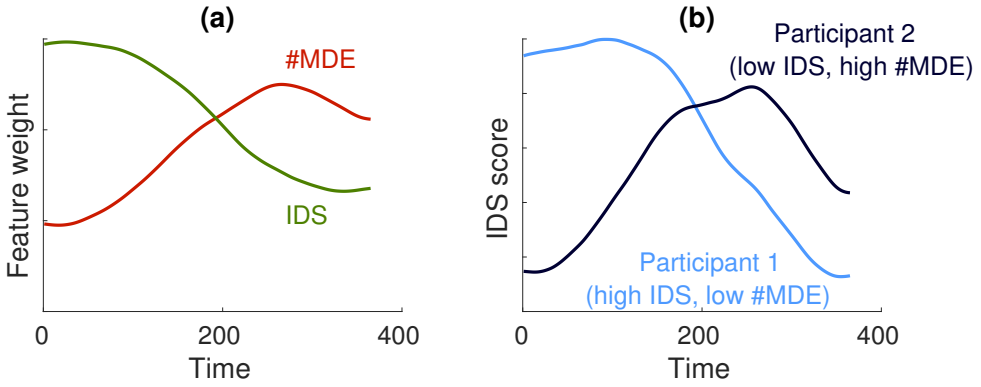


Figure 1: GPTP illustration with two covariates and two patients: Two covariates that are relevant in depression are the IDS, an index of total depressive severity, and #MDE, the number of previous depressive episodes. The IDS that was measured at the beginning of the observation period might be a good indicator of the disease activity in the near future (as measured by IDS scores at future time points), gradually becoming less important with time (red curve, left plot). On the contrary, the number of previous episodes might be a rather good indicator of the long-term development of the illness (green curve, left plot). The predictions for two quite different patients (one having high IDS and low #MDE, and the other having low IDS and high #MDE) are shown in the right plot.

sulting model is not readily interpretable. Varying Coefficient Models, which we present in the next section, instead model regression coefficients as functions of time, and one way of achieving this is to place a Gaussian Process (GP) prior on the coefficients. This way, coefficients can directly be calculated and interpreted.

Before becoming famous in the machine learning community, GPR was heavily used in the field of geostatistics, where it is known as Kriging (Matheron, 1963). However, GP analysis itself is much older - it was first applied to time series data by T.N. Thiele in 1880 (Lauritzen, 1981), and the underlying theory was further developed by N. Wiener and A.N. Kolmogorov (Wiener, 1948). Gaussian processes are already well established models for various temporal (and spatial) problems. Examples include Wiener processes / Brownian motion, Langevin processes and Kalman filter models.

In GP modeling - just as in other nonparametric methods - predictions are obtained without explicitly parameterizing the unknown function of the data. To achieve this, a Gaussian Process prior is placed directly on the space of functions. A GP can be thought of as the generalization of a Gaussian distribution to a function space of infinite dimension. Applying the GP to a particular data set gives rise to a function $f(\boldsymbol{x})$, which represents a single sample from this process.

We first introduce the so-called weight-space view, which allows interpretation of GPR as an extension of the GLM. One such mode of extension of the GLM as described in equation 5 is to include non-linear functions of individual covariates in the design matrix, which effectively allows for modeling of a weighted linear combination of non-linear relationships of the original covariates. For this pur-

pose, we define a basis function $\phi(\boldsymbol{x})$ as a mapping of covariates into a different space. In order to preserve clarity of notation and presentation of the methodology, we introduce GPR here as a time-series regression problem ($N = 1$) of $\boldsymbol{X}$ to $\boldsymbol{y}$, where each row of $\boldsymbol{X}$ corresponds to the covariates measured at one particular time point; we make the connection to longitudinal data at the end of this section.

3.1.1.1 Weight-space perspective When using basis functions, inference is equivalent to the linear model without basis functions (Rasmussen and Williams, 2006). The statistical model is thus the same as for the standard linear model in equation 5. In a full Bayesian treatment, the probability of observed values given parameters is called the likelihood, and is augmented with a prior on the weight vector $\boldsymbol{\beta}$. Assuming independent errors, we get

$$\begin{aligned} \boldsymbol{y} &\sim \mathcal{N}(\boldsymbol{\Phi\beta}, \sigma^{-2}\boldsymbol{I}) \\ \boldsymbol{\beta} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_\beta), \end{aligned} \tag{26}$$

where $\boldsymbol{\Phi}$ is the matrix whose entries are made up of basis function mappings of the covariates and $\boldsymbol{\Sigma}_\beta$ is the unconstrained variance of the regression weights. In the most typical use-case, it is assumed that nothing is known a priori about the mean of $\boldsymbol{\beta}$ and thus the prior mean is specified as $\boldsymbol{0}$, a vector of zeros.

This Bayesian linear basis function regression corresponds to the weight-space perspective of the Gaussian Process Regression Model. The 'trick' is - as we shall see shortly - that these basis functions need not necessarily be specified explicitly. Inference proceeds by calculating the posterior distribution of parameters given data according to Bayes' theorem in equation 1. Our convenient model formulation implies that the posterior, posterior predictive and marginal likelihood are all Gaussian, and can thus be derived analytically. It is easy to show (Rasmussen and Williams, 2006) that for Gaussians the posterior over regression coefficients is

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{\Phi}, \boldsymbol{y}) &= \frac{p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\boldsymbol{y}|\boldsymbol{\Phi})} \\ &= \frac{p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\beta})p(\boldsymbol{\beta})}{\int p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\beta})p(\boldsymbol{\beta})d\boldsymbol{\beta}} \\ &= \mathcal{N}(\sigma^{-2}\boldsymbol{A}^{-1}\boldsymbol{\Phi y}, \boldsymbol{A}^{-1}), \end{aligned} \tag{27}$$

where $\boldsymbol{A} = \sigma^{-2}\boldsymbol{\Phi\Phi}^\top + \boldsymbol{\Sigma}_\beta^{-1}$ is the inverse of the posterior variance, also known as posterior precision; it is a weighted average of the sample covariance matrix and the prior precision.

For the purpose of prediction, it is useful to define the noise-free observation function

$$f(\boldsymbol{x}_*) = \phi(\boldsymbol{x}_*)^\top\boldsymbol{\beta} = y_* - e_*, \tag{28}$$

where $\boldsymbol{x}_*$ is the vector of covariates at a new time point. The posterior predictive distribution for predicting a new observation $f(\boldsymbol{x}_*)$ from new covariates $\boldsymbol{x}_*$ can

also be derived analytically and is of the form

$$p(f(\boldsymbol{x}_*)|\phi(\boldsymbol{x}_*), \boldsymbol{\Phi}, \boldsymbol{y}) = \int p(\boldsymbol{\beta}|\boldsymbol{\Phi}, \boldsymbol{y})\ p(f(\boldsymbol{x}_*)|\phi(\boldsymbol{x}_*), \boldsymbol{\Phi}, \boldsymbol{y}, \boldsymbol{\beta})\ d\boldsymbol{\beta}$$
$$= \mathcal{N}(\sigma^{-2}\phi(\boldsymbol{x}_*)^\top \boldsymbol{A}^{-1}\boldsymbol{\Phi}\boldsymbol{y}, \phi(\boldsymbol{x}_*)^\top \boldsymbol{A}^{-1}\phi(\boldsymbol{x}_*)). \tag{29}$$

Finally, the model evidence / marginal likelihood in the denominator of equation 27 can be shown to be

$$p(\boldsymbol{y}|\boldsymbol{\Phi}) = \int p(\boldsymbol{\beta})\ p(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{\beta})\ d\boldsymbol{\beta}$$
$$= \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Phi}\boldsymbol{\Sigma}_\beta\boldsymbol{\Phi}^\top + \sigma^2\boldsymbol{I}). \tag{30}$$

This formulation of the marginal likelihood implies that in the Bayesian linear regression setup all observations $\boldsymbol{y}$ have a joint multivariate Gaussian distribution.

3.1.1.2 Function-space perspective Because observations $\boldsymbol{y}$ are jointly Gaussian, they can be considered a sample from a GP. More formally, a (univariate) GP is a random function $f : \mathbb{X} \to \mathbb{R}$ such that

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right), \tag{31}$$

where $m(\cdot)$ is the mean function, $k(\cdot, \cdot)$ the covariance function (also known as the kernel function) and $\boldsymbol{x} \in \mathbb{X}$ is a vector of covariates for a certain time point (each of these vectors makes up one row in $\boldsymbol{X}$). Finally, $\mathbb{X}$ is the set containing all values of $\boldsymbol{X}$, e.g. the set of real numbers, integers, or even letters or words (some kernels take non-numeric data as input).

The function-space view of Gaussian Process regression employs a GP as a prior over the noise-free $\boldsymbol{f} = [f(\boldsymbol{x}(t_1), ..., \boldsymbol{x}(t_T))]^\top$ as defined in equation 31; this implicitly defines the linear regression formulation. Let $p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$ and $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}, \sigma^2\boldsymbol{I})$. $\boldsymbol{C}$ specifies the kernel matrix, which is equivalent to the kernel function evaluated at all data points,

$$\boldsymbol{C} = \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_T) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_T, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_T, \boldsymbol{x}_T) \end{bmatrix}. \tag{32}$$

Then the marginal likelihood / model evidence is given by

$$p(\boldsymbol{y}|\boldsymbol{X}) = \int p(\boldsymbol{f}|\boldsymbol{X})\ p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{f})\ d\boldsymbol{f}$$
$$= \mathcal{N}(\boldsymbol{0}, \boldsymbol{C} + \sigma^2\boldsymbol{I}), \tag{33}$$

where we have assumed that the prior mean $m(\cdot)$ is zero everywhere. This is the typical assumption in GP regression, and is not a big restriction, as we shall see shortly. We note that equations 30 and 33 are equivalent if we let $\boldsymbol{C} = \boldsymbol{\Phi}\boldsymbol{\Sigma}_\beta\boldsymbol{\Phi}^\top$.

This means that we can use certain kernel functions $k(\cdot, \cdot)$ that implicitly define basis functions $\phi(\cdot)$. Conveniently, the posterior predictive distribution, too, can be rewritten such that all occurrences of $\boldsymbol{\Phi}$ can be replaced by $\boldsymbol{C}$. Consider the joint distribution of $\boldsymbol{y}$ and $f(\boldsymbol{x}_*)$,

$$\begin{bmatrix} \boldsymbol{y} \\ f(\boldsymbol{x}_*) \end{bmatrix} = \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_y^2 \boldsymbol{I} & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_*) \\ \boldsymbol{K}(\boldsymbol{x}_*, \boldsymbol{X}) & k(\boldsymbol{x}_*, \boldsymbol{x}_*). \end{bmatrix}\right). \tag{34}$$

The conditional distribution of $f(\boldsymbol{x}_*)$ is multivariate normal and follows from some basic linear algebra (see also Rasmussen and Williams (2006)),

$$\begin{aligned} p(\boldsymbol{x}_*|\boldsymbol{y}, \boldsymbol{X}) &= \mathcal{N}\left(f(\boldsymbol{x}_*)|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right) \\ \boldsymbol{\mu}_* &= \boldsymbol{K}(\boldsymbol{x}_*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{y} \\ \boldsymbol{\Sigma}_* &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{K}(\boldsymbol{x}_*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_*). \end{aligned} \tag{35}$$

Importantly, because we have replaced all occurances of $\boldsymbol{\Phi}$ by $\boldsymbol{C}$, inference can be done without ever having to explicitly compute the basis functions. This is known as the *kernel trick* (Rasmussen and Williams, 2006). It is a very powerful method because a simple kernel function may correspond to a very large set of basis functions. More precisely, *Mercer's theorem* states that for every positive definite covariance function $k(\cdot, \cdot)$, there exists a (possibly infinite) expansion in terms of basis functions (Mercer, 1909).

Usually, polynomial basis functions are employed to introduce the concept, such that $\phi_1(x) = x, \phi_2(x) = x^2$ and so on for all $x \in \boldsymbol{x}$. A more interesting choice is the set of radial basis functions that are centered around fixed points $\boldsymbol{c}_h$,

$$\phi_h(x) = a_h \exp\left[-\frac{(\boldsymbol{x} - \boldsymbol{c}_h)^2}{2r^2}\right]. \tag{36}$$

Such a set of basis functions corresponds to a simple feed-forward neural network with one hidden layer (MacKay, 2005). Interestingly, a GP with a kernel function of the same shape as equation 36 (which in this context is usually called the squared exponential kernel), corresponds to an infinite set of such radial basis functions (Rasmussen and Williams, 2006).

The formulation of the posterior predictive in equation 35 implies that the kernel function determines the generalization capabilities of the GP. Because the prior mean was chosen to be zero, it is possible to get a closed-form solution. Importantly, it becomes clear that a prior mean of zero does not imply that the posterior predictive be biased towards zero, at least for prediction close to the training data. However, if prediction needs to be done far away from the training data, the GP's prediction does revert to zero; depending on the chosen kernel this can happen quite rapidly. In such cases, or if one wishes to have a more readily interpretable model, one may model the mean explicitly; this approach can be thought of as letting the GP model the residual after applying the mean function, but has in general no closed-form solution, such that it is necessary to employ approximative methods.

The kernel function is typically parametrized. These hyper-parameters are obtained by optimization of the marginal likelihood, which effectively corresponds to model selection with respect to the space of models spanned by the hyper-parameters. Most often, a maximum-likelihood estimate is obtained, which is then used to compute the posterior predictive distribution.

3.1.1.3 Longitudinal data Our description of GPR so far has assumed simple time series data ($N = 1$), where at every time t a vector of covariates $\boldsymbol{x}(t)$ was measured. The most simple application of GPR to longitudinal data ($N > 1$) is to use time itself as the only covariate, and assume the same GP model for each participant,

$$y_{nt} - e_{nt} \sim \mathcal{GP}\left(m(t), k(t, t')\right), \tag{37}$$

in which case GP models correspond to a non-parametric version of growth curve modeling. As such, they are a much more powerful and flexible tool for estimating non-linear growth curves, as their functional shapes do not need to be specified. Only the parametric shape of the kernel function needs to be chosen. While a linear kernel results in estimation of a linear $f(t)$, much like a linear regression without basis functions, a more interesting choice of kernel might simply concern how smooth $f(t)$ should be, allowing for wildly different shapes.

The above mentioned squared exponential is a famous choice. However, it assumes that the process be infinitely differentiable, resulting in very smooth trajectories; this has often been shown to be a false assumption. The Matérn kernel contains an additional parameter that controls the level of smoothness, making it very popular in practical applications (Bishop, 2006); we define it in equation 49.

Since we have specified the same GP for each participant, all individual time series are forced to follow the same distribution. Consider a very simple example of such a GP with non-zero mean function,

$$\begin{aligned} m(t) &= a + bt \\ k(t_1, t_2) &= \delta(t_1, t_2)\sigma^2, \end{aligned} \tag{38}$$

where the delta function $\delta(\cdot, \cdot)$ equals one if the arguments match and zero otherwise. This corresponds to a linear trend with independent Gaussian observation noise.

Subject-specific effects can be introduced in a number of different ways. First, subject-specific covariates can be incorporated by concatenating time points and covariates into one long vector, which becomes the input for the mean and covariance function. However, the model is then difficult to interpret, as it defines some non-linear function on a combined space of time and covariates. Second, the mean function can be modified to incorporate covariates in a fixed effects fashion; for example, consider replacing t in equation 38 by some covariate. Third, random effects can be included as linear functions of the mean, but only if they are Gaussian, because a linear function of their covariance matrix can then simply be added to the kernel matrix in order to obtain a valid GP (Karch, 2016).

3.1.2 Varying Coefficient Models

In Varying Coefficient (VC) models (Hastie and Tibshirani, 1993) the coefficients $\beta_m(t)$ for covariates $m \in [1, M]$ are allowed to vary with so-called 'effect modifiers' γ_m; such effect modifiers could be time, or space, for example. VC models can be viewed as a generalization of the GLM, but also of generalized additive models (constant x_m), and dynamic generalized linear models (where the parameters follow an autoregressive process; see Fan and Zhang (2008)). The most common linear form of this model, where the task modifier is the same for all covariates (as is the case in spatial and temporal data), can be written as

$$y_n(t) = \boldsymbol{x}_n(t)^\top \boldsymbol{\beta}(t) + \epsilon_n(t), \tag{39}$$

where $\epsilon_n(t)$ is a white noise process, that is, $\forall t, t' : \mathbb{E}[\epsilon_n(t)] = 0$, $\mathrm{Var}[\epsilon_n(t)] = \sigma_\epsilon^2$ and $\mathrm{Cov}[\epsilon_n(t), \epsilon_n(t')] = 0$. We have changed notation slightly in comparison to the GLM by making y, x, β, ϵ functions of time to indicate that VC models enable (just like GPR) modeling time as a continuous process (the main advantage of which is that granularity of time can be greatly increased). Since we have one $\boldsymbol{\beta}(t)$, for each time point, the vector $\boldsymbol{\beta} = [\boldsymbol{\beta}(1)^\top, ..., \boldsymbol{\beta}(T)^\top]^\top$ has a total of MT entries. We can reshape and obtain M vectors $[\beta_m(1), ..., \beta_m(T)]^\top$ of length T, which can be interpreted as latent influence trajectories of the corresponding covariates.

One perspective on the parameters of the VC model can be obtained by comparison to a random effect model where $\boldsymbol{\beta}_t = \boldsymbol{\beta} + \boldsymbol{\zeta}_t$. In the VC model, the random effects $\boldsymbol{\zeta}_t$ become parameters that are explicitly being estimated, which in turn enables predicting $y_*(t)$ for out of sample participants with covariates $\boldsymbol{x}_*(t)$.

Due to the large number of parameters, the VC model is usually too general in the sense that coefficients cannot reliably be estimated without further constraints (Hastie and Tibshirani, 1993). The simplest solution is to directly propose specific functional shapes for the influence of covariates on target variables. This can greatly reduce the number of free parameters in the model, but at the cost of severely restricting the way in which covariates can influence target variables. Moreover, this approach suffers from poor generalization properties: predicting new target variables which fall outside the training intervals of the task-modifying variables can result in big errors.

Another approach, which has a long tradition in geospatial modeling is called Geographically Weighted Regression (GWR). It consists of specifying ensembles of local regression models, meaning that a different model is estimated for each time point (Brunsdon et al., 1996; Huang et al., 2010). The solution to each of the models is given by

$$\boldsymbol{w}_t = (\boldsymbol{X}'\boldsymbol{V}_t\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}_t\boldsymbol{y}. \tag{40}$$

This resembles weighted least squares, with the difference that the weight matrices $\boldsymbol{V}_t$ change with time instead of with $\boldsymbol{X}$. The weight matrices usually are diagonal with non-zero entries corresponding to $\exp(-\alpha d_j)$, where d_j is the

distance of the j-th data point from the current point and α is chosen using cross-validation. For computational reasons, a tapering function is often used to further simplify the problem, such that all data outside a local neighborhood of t are ignored. Lesage (2004) notes that local linear estimates based on a distance-weighted subsample of data may suffer from weak identification as the effective number of observations used to produce estimates for some points in space may be small. This problem can be somewhat alleviated under a Bayesian approach by incorporating prior information (LeSage, 2008). However, since this is a discrete time model, observations from each participant must be measured at the same time points when applied to longitudinal data.

Another possibility is to estimate a non-Bayesian non-parametric model and employ local smoothing methods; this is the approach taken in Temporally Varying Coefficient Models (Wu and Yu, 2002; Fan and Zhang, 2008). Estimation of parameters can be done in several ways (Park et al., 2013). For example, one could employ local polynomial kernels, where functions are locally approximated by Taylor expansion (but the bandwidths of the kernels need to be chosen). Lin and Carroll (2000) show that this method usually inflates variability. Alternatively, smoothing splines express the function as a linear combination of spline bases. This is a more flexible approach, but the number of knots and their locations need to be somehow chosen, which is difficult to do in a principled way. In either case, coefficients are estimated which are then applied to some set of basis functions. Since the amount of coefficients is necessarily always finite, these methods are not truly non-parametric.

Finally, it is also possible to adopt a Bayesian perspective; this is the approach taken in Spatially Varying Coefficient models (SVC; see Gelfand et al. (2003)). The model specification is completed by assuming some prior distribution over $\boldsymbol{\beta}$. Perhaps the most natural way this can be done is to specify a *multivariate* Gaussian Process prior (Wheeler and Calder, 2006; Gelfand et al., 2003). In the context of longitudinal data, this process would imply covariation between covariates as well as between time points. A multivariate GP is therefore characterized by a cross-covariance function $\boldsymbol{K}(\boldsymbol{X}(t), \boldsymbol{X}(t'))$, which returns a matrix for each pair of arguments. Such a function is also known as a 'random field'. Assuming that all individuals have been measured at the same time points, this function yields the covariance matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}(1), \boldsymbol{X}(1)) & \cdots & \boldsymbol{K}(\boldsymbol{X}(1), \boldsymbol{X}(T)) \\ \vdots & \ddots & \vdots \\ \boldsymbol{K}(\boldsymbol{X}(T), \boldsymbol{X}(1)) & \cdots & \boldsymbol{K}(\boldsymbol{X}(T), \boldsymbol{X}(T)) \end{bmatrix}. \tag{41}$$

Several methods have been proposed to construct valid cross-covariance functions. The most popular approach is to combine univariate covariances; this is called coregionalization for stationary random fields (Bourgault and Marcotte, 1991). It consists of representing the random field as a linear combination of independent univariate GPs. The resulting cross-covariance functions take the

form

$$\boldsymbol{K}(\boldsymbol{X}(t_1), \boldsymbol{X}(t_2))_{m_1,m_2} = \sum_{j=1}^{J} \rho_k(t_1, t_2) \boldsymbol{A}_{m_1,j} \boldsymbol{A}_{m_2,j}, \tag{42}$$

where $\rho_k(t_1, t_2)$ are valid stationary correlation functions. This is only feasible for a limited amount J of processes. In any case, the smoothness of any component is restricted to that of the roughest one (Genton and Kleiber, 2015), so $J > 1$ is rare. When $J = 1$, the cross-covariance function is termed 'separable', because it is possible to write the resulting covariance matrix in the form

$$\boldsymbol{C} = \boldsymbol{R} \otimes \boldsymbol{A}, \tag{43}$$

where $\boldsymbol{R}$ is the time-invariant covariance of regression weights between features, $\boldsymbol{A}$ is the feature-invariant covariance of regression weights between time points and $\otimes$ symbolizes the Kronecker product, which is defined as

$$\boldsymbol{R} \otimes \boldsymbol{A} = \begin{bmatrix} r_{11}\boldsymbol{A} & \cdots & r_{1T}\boldsymbol{A} \\ \vdots & \ddots & \vdots \\ r_{T1}\boldsymbol{A} & \cdots & r_{TT}\boldsymbol{A} \end{bmatrix}. \tag{44}$$

The on-diagonal matrices $r_{tt}\boldsymbol{A}$ are also called marginal covariances and off-diagonal matrices are called cross-covariances. While separability simplifies the problem of estimating the cross-covariance computationally, an obvious limitation is that all features share the same covariance structure $\boldsymbol{A}$.

We pause here for a moment to make a connection to transfer learning (Caruana, 1997), where knowledge about one task is applied to another task. Instead of having time or space as task modifiers, more generally any *task* variable can be employed (Bonilla et al., 2007; Bussas et al., 2015). An example of transfer learning via VC models is presented in Bonilla et al. (2008), where a free-form task-similarity matrix is learned. If we redefine $\boldsymbol{R}$ as the task kernel and $\boldsymbol{A}$ as the feature kernel, we get a decomposition into learning task similarity and learning feature similarity.

Other approaches of specifying the cross-covariances are rare. A possibility is to extend the univariate Matérn kernel function, which takes two parameters, to the multivariate case by parametric coupling of univariate kernels (Apanasovich et al., 2012; Gneiting et al., 2010). For example, smoothness parameters for the cross-covariances are set to the arithmetic average of the respective marginal variances, while length-scales are assumed to be the same across all covariances (Gneiting et al., 2010). In any case, within the VC model this just specifies the prior on the regression coefficients, so in cases where enough data is present to sufficiently determine the posterior, the particular choice for the covariance of the prior should not affect predictions too much.

There are recent implementations for inference in such models; an example is spBayes for the R language (Finley et al., 2007, 2015). The package implements MCMC inference for cross-covariance matrices specified through the linear model

of coregionalization, which the authors argue to be the only feasible approach to date. Still, without further restricting assumptions any such general method that specifies cross-covariance globally for all measurements scales cubically with the number of measurements (due to expensive matrix inversions). Within an MCMC inference scheme, this quickly becomes prohibitively slow, and we could not find any application to more than $M = 2$ features. Other attempts have been made to specify valid multivariate GP priors and perform inference, but these approaches do not scale to many features either (Alvarez and Lawrence, 2009; Melkumyan and Ramos, 2011).

VC models allow the estimation of time-varying coefficients, and consequently enable prediction of new observations $\boldsymbol{y}_*$ given new covariates $\boldsymbol{X}_*$. This is not possible in the HLM family, as the regression coefficients are considered to be random effects. There are different ways in which coefficients can be estimated, but from a Bayesian perspective the most principled approach is to employ a GP prior whose hyper-parameters can be estimated by optimizing the marginal likelihood. This, again, is unfortunately computationally expensive, so applications to more than just a few covariates are rare. However, when employing recent advances in Gaussian Processes (Lázaro-Gredilla et al., 2010), such a method can scale to hundreds of features and time points as well as thousands of individuals, as we will show in the forthcoming sections. This should be sufficient for all but the biggest studies in the field of computational psychiatry.

3.2 Model formulation

As stated before, we combine a GP prior with an SVC model in order to estimate feature influence trajectories $[\beta_m(1), ..., \beta_m(T)]^\top$ for covariates $m \in [1, M]$, such that observed disease trajectories $[y_n(1), ..., y_n(T)]^\top$ arise as a weighted sum of covariates, where the weights are the latent feature influence trajectories. Let M be the number of covariates $\boldsymbol{x}_n \in \mathbb{R}^M$ which are available at baseline for subjects $n \in [1, N]$. Our goal is to estimate the clinical disease trajectory $y_n(t)$,

$$p(y_n(t)|\boldsymbol{\beta}(t), \boldsymbol{x}_n) \sim \mathcal{N}(\boldsymbol{x}_n^\top \boldsymbol{\beta}(t), \sigma_y^2). \tag{45}$$

This mirrors the linear SVC form that we presented in equation 39. For simplicity, we have assumed a Gaussian likelihood with a spherical covariance, but this model can be readily extended to the non-Gaussian non-spherical case if needed (Wang and Neal, 2012). We place a multivariate GP prior over the vector of regression weights:

$$\boldsymbol{\beta}(t) \sim \mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{K}(t, t')), \tag{46}$$

where $\boldsymbol{\mu}(t)$ is the M-dimensional mean vector and $\boldsymbol{K}(t, t')$ is any valid $M \times M$-dimensional cross-covariance function as discussed in section 3.1.2.

The corresponding graphical model is depicted in figure 2; it can be interpreted as a continuous transfer-learning problem: We have a standard Bayesian linear regression (Gaussian likelihood and prior) for each time instant and the Gaussian process prior over the weights captures the correlation between the

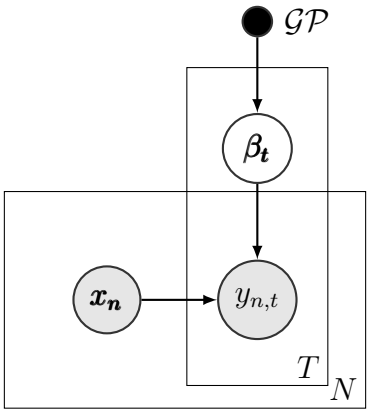


Figure 2: Graphical model for Gaussian Process Trajectory Prediction.

regression problems over time, with closer time instants having more correlated weights. Examining the posterior distributions over the weight trajectories $\boldsymbol{\beta}$ might allow insights into how the different covariates contribute to the temporal course of the overall response, and allow for predictions at future time points.

We choose a separable cross-covariance function, such that for the kernel matrix $\boldsymbol{C} = \boldsymbol{R} \otimes \boldsymbol{A}$, where $\boldsymbol{R}$ is the time-invariant covariance of regression weights between features with dimensionality $M \times M$ and $\boldsymbol{A}$ is the feature-invariant covariance of regression weights between time points with dimensionality $T \times T$, such that dimensionality of $\boldsymbol{C}$ is $MT \times MT$ (see section 3.1.2 for discussion of different types of cross-covariance functions). Further, we assume a diagonal $\boldsymbol{R} = \text{diag}([\sigma_1^2, ..., \sigma_M^2])$, because we expect that different weights are independent for a given time instant (this resembles the standard assumption in linear regression). This shape of $\boldsymbol{R}$ is very useful, as it enables us to perform feature selection, as we discuss in section 3.2.3.

The resulting kernel matrix has the following form:

$$\boldsymbol{C} = \begin{bmatrix} \sigma_1^2\boldsymbol{A} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_2^2\boldsymbol{A} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \sigma_{M-1}^2\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \sigma_M^2\boldsymbol{A} \end{bmatrix}. \tag{47}$$

For matrix $\boldsymbol{A}$, which specifies covariance in time, any valid kernel function may be chosen that produces a positive semi-definite covariance matrix for all possible inputs. For example, constant trends would be captured by a kernel that produces a matrix of ones for any data. Very smooth trajectories are captured by a squared exponential kernel of the form

$$k_{\text{RBF}}(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_f^2 \exp\left[-\frac{1}{2}(\boldsymbol{x}_p - \boldsymbol{x}_q)^\top \boldsymbol{G} (\boldsymbol{x}_p - \boldsymbol{x}_q)^\top\right], \tag{48}$$

where usually $\boldsymbol{G} = \ell^{-2}\boldsymbol{I}$. The length scale ℓ specifies at which scale the trajectories are expected to vary with the features. One kernel that is particularly often

used for real-world data (and which we will use later in our application) is the Matérn kernel

$$k(t_1, t_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{||\boldsymbol{x}_p - \boldsymbol{x}_q||\sqrt{2\nu}}{\ell} \right)^{\nu} K_\nu \left(\frac{||\boldsymbol{x}_p - \boldsymbol{x}_q||\sqrt{2\nu}}{\ell} \right) \tag{49}$$

with positive parameters ν and ℓ. $\Gamma(\nu) = \int_0^\infty z^{\nu-1} e^{-z} dz$ is the gamma function and K_ν is the modified Bessel function of the second kind; see Rasmussen and Williams (2006) for details. The roughness of the process is specified by ν, where higher ν indicates less roughness. In the limit of $\nu \to \infty$, this function equals the squared exponential kernel. Both the squared exponential and Matérn kernels are examples of 'universal kernels', which under broad conditions are capable of learning any continuous function given enough data (Micchelli et al., 2006).

We use the same ν and ℓ for all M features to reduce the number of hyper-parameters that need to be learned. This is not a strong restriction, as we show in section 3.7 - it is still possible to estimate very different types of feature influence trajectories.

Following the formulation above, we can reformulate the prior - perhaps more simply - as M independent univariate GP priors (one per covariate):

$$\beta_m(t) \sim \mathcal{GP}(0, \sigma_m^2 k(t, t')) \quad \forall m \in [1, M]. \tag{50}$$

We have assumed that the mean function is zero, because we have no prior knowledge whether the weights are positive or negative. If we used the same variance for all features such that $\boldsymbol{R} = \text{diag}([\sigma^2, ..., \sigma^2])$, the model would contain a total of four hyper-parameters $\boldsymbol{\theta} = (\sigma, \nu, \ell, \sigma_y)$ that need to be estimated from data through optimization of the marginal likelihood. This is a relatively easy optimization problem even for small datasets, as the marginal is only mildly non-convex. Thus, a few random restarts of the optimization are sufficient to find the global optimum with high certainty (Rasmussen and Williams, 2006).

3.2.1 Inference

Assume we have a dataset $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$. For the sake of notational simplicity, we further assume here that there are no missing observations and that all time series have been sampled at the same time instances even though the method readily extends to the case of uneven sampling.

Let the vector of all concatenated observations $\boldsymbol{y} = [\boldsymbol{y}_1^\top, ..., \boldsymbol{y}_N^\top]^\top$ with $\boldsymbol{y}_n = [y_n(t_1), \ldots, y_n(t_T)]^\top$. Further, let $\boldsymbol{X} = [\boldsymbol{X}_1^\top, ..., \boldsymbol{X}_N^\top]^\top$, where

$$\boldsymbol{X}_n = \begin{bmatrix} x_{n1} & \cdots & x_{n1} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & x_{n2} & \cdots & x_{n2} & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & x_{nM} & \cdots & x_{nM} \end{bmatrix}. \tag{51}$$

This allows us to rewrite equation 45 in the vectorized forms

$$p(\boldsymbol{y}_n | \boldsymbol{\beta}, \boldsymbol{X}_n) \sim \mathcal{N}(\boldsymbol{X}_n \boldsymbol{\beta}, \sigma_y^2 \boldsymbol{I}) \quad \text{and} \quad p(\boldsymbol{y} | \boldsymbol{\beta}, \boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma_y^2 \boldsymbol{I}). \tag{52}$$

In case we do have missing values, standard marginalization rules for Gaussians can be applied, assuming that the data are missing at random, i.e. there is no relation between response variables and absence of data. This turns out to be very simple: missing covariate values can just be replaced by the value zero in $\boldsymbol{X}$, and columns of $\boldsymbol{X}$ corresponding to missing entries of $\boldsymbol{y}$ may be removed (Rasmussen and Williams, 2006).

Although we assume here that covariates are constant in time, an extension to time-varying covariates is trivial by redefining $\boldsymbol{X}_n$ appropriately,

$$\boldsymbol{X}_n = \begin{bmatrix} x_{n11} & \cdots & x_{n1T} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & x_{n21} & \cdots & x_{n2T} & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & x_{nMT} & \cdots & x_{nMT} \end{bmatrix}. \tag{53}$$

However, extending the approach in this way precludes the possibility of forecasting the response $\boldsymbol{y}$ for future time points, as covariates for these time points are not available yet.

Inference proceeds by optimization of the marginal likelihood. We derive an analytic closed-form expression for it from the expression of the joint probability of regression weights and observations,

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\beta} | \boldsymbol{X}, \boldsymbol{\theta}) &= p(\boldsymbol{\beta}|\boldsymbol{\theta}) p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{\theta}) \\ &= p(\boldsymbol{\beta}|\boldsymbol{\theta}) p(\boldsymbol{y}_1, ..., \boldsymbol{y}_n | \boldsymbol{\beta}, \boldsymbol{X}_1, ..., \boldsymbol{X}_N, \boldsymbol{\theta}) \\ &= p(\boldsymbol{\beta}|\boldsymbol{\theta}) \prod_n p(\boldsymbol{y_n}|\boldsymbol{\beta}, \boldsymbol{X}_n, \boldsymbol{\theta}), \end{aligned}$$

where we assumed that individuals are independent, such that their joint factorizes into components $p(\boldsymbol{y_n}|\boldsymbol{\beta}, \boldsymbol{X}_n, \boldsymbol{\theta})$. Plugging in our model assumptions, we can write

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\beta} | \boldsymbol{X}, \boldsymbol{\theta}) &= p(\boldsymbol{\beta}|\boldsymbol{\theta}) \prod_n p(\boldsymbol{y_n}|\boldsymbol{\beta}, \boldsymbol{X}_n, \boldsymbol{\theta}) \\ &\propto \exp\left(-\frac{1}{2} \left(\boldsymbol{\beta}^\top \boldsymbol{C}^{-1} \boldsymbol{\beta} \right) \right) \cdot \exp\left(\sum_n \sigma_y^{-2} (\boldsymbol{y}_n - \boldsymbol{X}_n \boldsymbol{\beta})^\top (\boldsymbol{y}_n - \boldsymbol{X}_n \boldsymbol{\beta}) \right) \\ &= \exp\left(-\frac{1}{2} \left(\boldsymbol{\beta}^\top \boldsymbol{C}^{-1} \boldsymbol{\beta} + \sum_n \sigma_y^{-2} (\boldsymbol{y}_n - \boldsymbol{X}_n \boldsymbol{\beta})^\top (\boldsymbol{y}_n - \boldsymbol{X}_n \boldsymbol{\beta}) \right) \right) \\ &= \exp\left(-\frac{1}{2} \left(\boldsymbol{\beta}^\top \boldsymbol{C}^{-1} \boldsymbol{\beta} + \sigma_y^{-2} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \right) \right) \\ &= \exp\left(-\frac{1}{2} \left(\boldsymbol{\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} - 2 \boldsymbol{\mu_\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} + \sigma_y^{-2} \boldsymbol{y}^\top \boldsymbol{y} \right) \right), \end{aligned} \tag{54}$$

where $\boldsymbol{\theta} = [\boldsymbol{\zeta}, \sigma_y]$ is a vector containing all hyper-parameters, $\boldsymbol{\zeta}$ being the hyper-parameters that define the kernel matrix $\boldsymbol{C}$. For example. if we choose the Matérn kernel for the feature-invariant covariance of regression weights between

time points $\boldsymbol{A}$, we have $\boldsymbol{\zeta} = [\sigma_1, ..., \sigma_M, \ell, \nu]$ and $\dim(\boldsymbol{\zeta}) = M + 2$. In the last line of equation 54, we have substituted

$$\begin{aligned} \boldsymbol{\mu}_\beta &= \sigma_y^{-2} \boldsymbol{V}_\beta \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{V}_\beta &= (\boldsymbol{C}^{-1} + \sigma_y^{-2} \boldsymbol{X}^\top \boldsymbol{X})^{-1}. \end{aligned} \tag{55}$$

Since the sum of Gaussians is another Gaussian, it follows that the joint is Gaussian. The posterior is proportional to the joint up to a normalizing constant, and thus Gaussian, too. The relation between the different distributions is given by Bayes' theorem, which we defined in equation 1. Here, we repeat it in a slightly modified form that incorporates the data / parameters of our method. Assuming for the moment that $\boldsymbol{\theta}$ is known, we can write

$$\underbrace{p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{\beta}|\boldsymbol{\theta})}^{\text{prior}} \overbrace{p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{\theta})}^{\text{likelihood}}}{\underbrace{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}_{\text{marginal likelihood}}} = \frac{\overbrace{p(\boldsymbol{y}, \boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{\theta})}^{\text{joint}}}{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}. \tag{56}$$

With respect to the coefficients $\boldsymbol{\beta}$, the posterior is proportional to the joint, as the marginal likelihood does not depend on $\boldsymbol{\beta}$. So starting from the last line in equation 54, we can complete the square,

$$\begin{aligned} &p(\boldsymbol{y}, \boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{\theta}) \\ &\quad \propto \exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} - 2\boldsymbol{\mu_\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} + \sigma_y^{-2} \boldsymbol{y}^\top \boldsymbol{y}\right)\right) \\ &\quad = \exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} - 2\boldsymbol{\mu_\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} + \sigma_y^{-2} \boldsymbol{y}^\top \boldsymbol{y} + \boldsymbol{\mu}_\beta^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\mu}_\beta - \boldsymbol{\mu}_\beta^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\mu}_\beta\right)\right) \\ &\quad = \underbrace{\exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} - 2\boldsymbol{\mu_\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} + \boldsymbol{\mu}_\beta^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\mu}_\beta\right)\right)}_{\text{(I)}} \cdot \underbrace{\exp\left(-\frac{1}{2}\left(\sigma_y^{-2} \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{\mu}_\beta^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\mu}_\beta\right)\right)}_{\text{(II)}}. \end{aligned} \tag{57}$$

Note the proportional sign in the second line, which we need because we ignore the normalizing constant of the Gaussian distribution. Term (II) is constant with respect to $\boldsymbol{\beta}$, and term (I) is the exponential of a quadratic function and thus proportional to a Gaussian; we can derive that the functional shape of the posterior must be

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}) &= \frac{1}{Z} \exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} - 2\boldsymbol{\mu_\beta}^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\beta} + \boldsymbol{\mu}_\beta^\top \boldsymbol{V}_\beta^{-1} \boldsymbol{\mu}_\beta\right)\right) \\ &= \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_\beta, \boldsymbol{V}_\beta), \end{aligned} \tag{58}$$

where Z indicates all terms that are independent of $\boldsymbol{\beta}$. Term (II) can further be

used to calculate the marginal likelihood,

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) &= \frac{p(\boldsymbol{y},\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{\theta})} \\ &= \frac{1}{Z}\exp\left(-\frac{1}{2}\left(\sigma_y^{-2}\boldsymbol{y}^\top\boldsymbol{y} - \boldsymbol{\mu}_\beta^\top\boldsymbol{V}_\beta^{-1}\boldsymbol{\mu}_\beta\right)\right) \\ &= \frac{1}{Z}\exp\left(-\frac{1}{2}\left(\sigma_y^{-2}\boldsymbol{y}^\top\boldsymbol{y} - \sigma_y^{-4}\boldsymbol{y}^\top\boldsymbol{X}\boldsymbol{V}_\beta\boldsymbol{X}^\top\boldsymbol{y}\right)\right) \\ &= \frac{1}{Z}\exp\left(-\frac{1}{2}\boldsymbol{y}^\top\left(\sigma_y^{-2}I - \boldsymbol{X}\left(\sigma_y^4\boldsymbol{C}^{-1} + \sigma_y^2\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\right)\boldsymbol{y}\right) \\ &= \frac{1}{Z}\exp\left(-\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{V}_y^{-1}\boldsymbol{y}\right) \\ &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{V}_y), \end{aligned} \tag{59}$$

where Z now indicates terms that are independent of $\boldsymbol{y}$. Note that the covariance $\boldsymbol{V}_y$ can be simplified further using the well-known Woodbury matrix identity,

$$(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{C}\boldsymbol{V})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})\boldsymbol{V}\boldsymbol{A}^{-1}. \tag{60}$$

It follows that

$$\begin{aligned} \boldsymbol{V}_y &= \left(\sigma_y^{-2}\boldsymbol{I} - \boldsymbol{X}\left(\sigma_y^4\boldsymbol{C}^{-1} + \sigma_y^2\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\right)^{-1} \\ &= \sigma_y^2\boldsymbol{I} + \sigma_y^4\boldsymbol{X}\left(\sigma_y^4\boldsymbol{C}^{-1} + \sigma_y^2\boldsymbol{X}^\top\boldsymbol{X} - \sigma_y^2\boldsymbol{X}^\top\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top \\ &= \sigma_y^2\boldsymbol{I} + \boldsymbol{X}\boldsymbol{C}\boldsymbol{X}^\top. \end{aligned} \tag{61}$$

We can gain an interesting insight by looking at the first line of equation 61. In the innermost bracket, weighted versions of the inverse of the kernel matrix $\boldsymbol{C}^{-1}$ and the covariance $\boldsymbol{X}^\top\boldsymbol{X}$ are added together. This implies that for a large number of subjects N, marginal likelihood optimization is *independent of* the chosen kernel function. This makes perfect sense from a Bayesian perspective where the GP is used as a prior on the regression weights. The influence of the prior is high for small amounts of data and decreases as the amount of data grows. However, in *standard* Gaussian Process regression, where the GP prior is on $\boldsymbol{y}$ instead of $\boldsymbol{\beta}$, the marginal likelihood cannot become independent of the kernel even in the limit of infinite data (such that for example a linear kernel would always constrain the model of the data to be linear). The same argument also holds for the posterior means $\boldsymbol{\mu}_\beta$ in equation 56.

3.2.2 Prediction

Finally, the posterior predictive distribution for a new set of baseline features $\boldsymbol{x}_*$ follows from the fact that all data are jointly multivariate Gaussian,

$$\begin{bmatrix} \boldsymbol{y} \\ f(\boldsymbol{x}_*) \end{bmatrix} = \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{X}\boldsymbol{C}\boldsymbol{X}^\top + \sigma_y^2\boldsymbol{I} & \boldsymbol{X}\boldsymbol{C}\boldsymbol{X}_*^\top \\ \boldsymbol{X}_*\boldsymbol{C}\boldsymbol{X}^\top & \boldsymbol{X}_*\boldsymbol{C}\boldsymbol{X}_*^\top \end{bmatrix}\right). \tag{62}$$

where $\boldsymbol{X}_*$ is defined in the same manner as $\boldsymbol{X}_n$ in equation 51 and$f(\boldsymbol{x}_*) = \boldsymbol{y}_* - \boldsymbol{e}_*$ denotes the noise-free observations. The conditional distribution of $f(\boldsymbol{x}_*)$ follows from the calculation of the Schur complement, which arises as the result of performing a block Gaussian elimination (see also Rasmussen and Williams (2006) for further details),

$$\begin{aligned} p(f(\boldsymbol{x}_*)|\boldsymbol{y}, \boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{\theta}) &= \mathcal{N}\left(f(\boldsymbol{x}_*)|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right) \\ \boldsymbol{\mu}_* &= \boldsymbol{X}_* \boldsymbol{C} \boldsymbol{X}^\top (\boldsymbol{X} \boldsymbol{C} \boldsymbol{X}^\top + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{y} \\ \boldsymbol{\Sigma}_* &= \boldsymbol{X}_* \boldsymbol{C} \boldsymbol{X}_*^\top - \boldsymbol{X}_* \boldsymbol{C} \boldsymbol{X}^\top (\boldsymbol{X} \boldsymbol{C} \boldsymbol{X}^\top + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{X} \boldsymbol{C} \boldsymbol{X}_*^\top. \end{aligned} \tag{63}$$

3.2.3 Automatic Relevance Determination

So far, we have assumed that all covariates are somehow known to be relevant for the prediction. This may not always be the case. In fact, for some applications there may be a large number of baseline covariates and it may specifically be of interest to infer which baseline features are most informative.

Feature selection can be performed in Gaussian Processes by using an automatic relevance determination approach (ARD; Williams and Rasmussen (1996)). For example, consider the squared exponential kernel for regular Gaussian Process regression

$$k_{\text{RBF}}(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_f^2 \exp\left[-\frac{1}{2}(\boldsymbol{x}_p - \boldsymbol{x}_q)^\top \boldsymbol{G} (\boldsymbol{x}_p - \boldsymbol{x}_q)^\top\right]. \tag{64}$$

Instead of just defining one length scale parameter with $\boldsymbol{G} = \ell^{-2}\boldsymbol{I}$, each covariate can be allowed to have its own length scale,

$$\boldsymbol{G} = \begin{bmatrix} \ell_1^{-2} & 0 & \cdots & 0 & 0 \\ 0 & \ell_2^{-2} & \cdots & 0 & 0 \\ \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \ell_{M-1}^{-2} & 0 \\ 0 & 0 & \cdots & 0 & \ell_M^{-2} \end{bmatrix}. \tag{65}$$

Since the inverse of the length-scale determines how relevant a feature is, the covariance will change very little for features whose length-scales have very large values. This in turn effectively removes the corresponding features from the inference.

We adopt this approach to the problem of varying coefficients: Each univariate GP in equation 50 specifies one features' influence trajectory over time. ARD results from allowing a separate variance term σ_m^2 for each covariate, $\boldsymbol{R} = \text{diag}([\sigma_1^2, ..., \sigma_M^2])$, just as we defined in section 3.2: Features whose variances σ_m^2 are estimated to be very small effectively do not influence prediction as the variances scale the output of the univariate kernel function $k(t, t')$. Thus each σ_m^2 can be interpreted as an inverse length scale.

Sparse solutions can now be encouraged by placing a sparse prior on each σ_m (Williams and Rasmussen, 1996). We here use a spike and slab prior, which

consists of a mixture of Gaussians,

$$p(\sigma_m^2) = \frac{1}{2}\mathcal{N}(0, \sigma_{\text{spike}}^2) + \frac{1}{2}\mathcal{N}(0, \sigma_{\text{slab}}^2). \tag{66}$$

The spike component has a much smaller variance than the slab component, such that parameter values are shrunk towards zero. While we could have chosen a different weighing of spike versus slab prior, additional simulations showed only very minor effects on the results for several different weighing schemes.

Starting with $\sigma_{\text{spike}} = 1.0$, the marginal likelihood is optimized with respect to the hyper-parameters $\boldsymbol{\theta}$ via gradient descent using MATLAB's `fminunc` with standard settings (this uses an algorithm which is a variant of the interior-reflective Newton method). This is performed multiple times with different random initializations for the hyper-parameters, as the optimization problem is (weakly) non-convex, and thus we are not guaranteed to find the optimum from every set of initial conditions. Note that the result is a maximum likelihood point estimate of the hyper-parameters $\hat{\boldsymbol{\theta}}$. While full Bayesian inference is possible, there is no closed form solution for the posterior over $\boldsymbol{\theta}$, and consequently it must be approximated. This, however, is generally slow (MacKay, 2005).

We note that as the spike variance decreases, more and more features are set (very close) to zero. Thus, by shrinking the variance step by step, we get a sequence of consecutively sparser models until no feature survives. Specifically, we use a procedure which iteratively halves the width of the spike on each iteration and drops the features with variances below some threshold. Removing features is not necessary, but greatly accelerates the procedure. For each setting of σ_{spike}, the parameters are estimated again without the tight prior (setting $\sigma_{\text{spike}} = \sigma_{\text{slab}}$) in order to get estimates that are not shrunk towards zero.

The threshold below which features can be discarded depends on the data at hand. It can automatically be selected such that discarding features changes the optimal marginal likelihood value only by a negligible amount. Listing 1 shows the pseudo-code for feature selection through ARD.

Amongst all models that we collect along the sparsification path, the best model can be chosen by using the Bayesian Information Criterion (BIC, aka Schwarz criterion; see Schwarz (1978)), which adds a penalization term to the likelihood, in order to punish models with too many parameters. This is a well established model selection criterion, and as such successfully prevents overfitting. It is defined as

$$\text{BIC} = \|\boldsymbol{\theta}\|_0 \log\|\boldsymbol{y}\|_0 - 2\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}), \tag{67}$$

where $\|\cdot\|_0$ denotes the 0-norm, which simply returns the size of its input, and $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$ is the marginal likelihood. Thus, the first term scales linearly with the number of parameters and acts as a penalty for complex models with many parameters.

Listing 1: Pseudo-code for feature selection through ARD.

```
function shrinkParams(data, M, sigmaThresh)
    slabPriorVar = 3
    spikePriorVar = 1
    m = M
    i = 1
    while m > 0
        zetaSpike, sigmaSpike, fvalSpike = optimMarginal(
           spikePriorVar, data)
        for k=1=length(sigmaSpike)
            li = []
            if sigmaSpike(k) < sigmaThreshold
                li.append(k)
                m = m - 1
            end
        end
        # remove covariates for which sigmaSpike(k) < sigmaThreshold
        data = deleteCovars(data, li)
        zeta(i), sigma(i), fval(i) = optimMarginal(slabPriorVar,
           data)
        spikePriorVar = spikePriorVar / 2
        i = i + 1
    end
    return zetaVec, sigmaVec, fval
end

function optimMarginal(sigmaPriorVar, data)
    # maximize marginal with multiple random initializations
    for i=1:nRestarts
        fval(i), zeta(i), sigma(i)  = maximize(marginal,
           sigmaPriorVar, data)
    end
    # sort in descending order and get indices to ordered elements
    sortIdx = argsort(fval)
    return zeta(sortIdx(1)), sigma(sortIdx(1)), fval(sortIdx(1))
end
```

3.2.4 Complexity

The marginal covariance $\boldsymbol{V}_y$ that we defined in equation 61 is of dimension $NT \times NT$. During optimization of the marginal likelihood with respect to the hyper-parameters, the most expensive operation is inversion of this matrix (an operation that scales cubically with the size of the matrix), which - depending on the problem at hand - might be very large. In the case $M < N$ a simple solution is to compute $\boldsymbol{V}_y^{-1}$ via the first line in equation 61. This operation requires two inversions of matrices of size $MT \times MT$, so it scales cubically with MT instead of NT.

Using pseudo-samples procedures (see Lázaro-Gredilla et al. (2010) and references therein), where essentially only part of the data is used for computing the matrix, the complexity can be reduced to $O(M^3T)$ (or $O(N^3T)$, respectively. Here, we assume that M^3 is not a limiting factor, as this would be the typical complexity of any linear regression method. For large M we would need to rely

on distributed regression procedures, which are outside the scope of this work.

3.3 Cross-Validation

In machine learning, available data is split into three parts: training, test and validation data. The training data is used to fit the model, which is evaluated on the test data. This is often repeated many times in the quest for finding better models. After having decided for one model (family), it is applied to the validation data, in order to get a reasonable estimate of the generalization ability to previously unseen data. When using a procedure called k-fold cross-validation, the data that was not set aside as validation data is partitioned into k parts of equal size, and k separate models are estimated; for fitting the k-th model, the k-th partition is used as test data; the remaining $k-1$ partitions are combined into the training data. A model score is computed for each test partition, and the total model score is computed as the average of each partition's score. Most common is $k = 10$, which we also use in this work. A schematic representation is depicted in Figure 3.

Here, the prior shrinkage procedure described in section 3.2.3 yields a set of models. In order to prevent overfitting, it is important that only one model per CV fold is scored on the test data. This is why we use the BIC to compare the models that we obtained along the shrinkage path. The best model is then applied to the left-out data partition; predictions for observations $\hat{\boldsymbol{y}}$ are computed for each participant. Since we assume Gaussian residuals, the log-likelihood is proportional to the mean squared error - however, here, we choose the root mean squared error $\text{RMSE}(\boldsymbol{y}) = \sqrt{\mathbb{E}[(\hat{\boldsymbol{y}} - \boldsymbol{y})^\top (\hat{\boldsymbol{y}} - \boldsymbol{y})]}$ for scoring the model, as the RMSE values have the same scaling as the data.

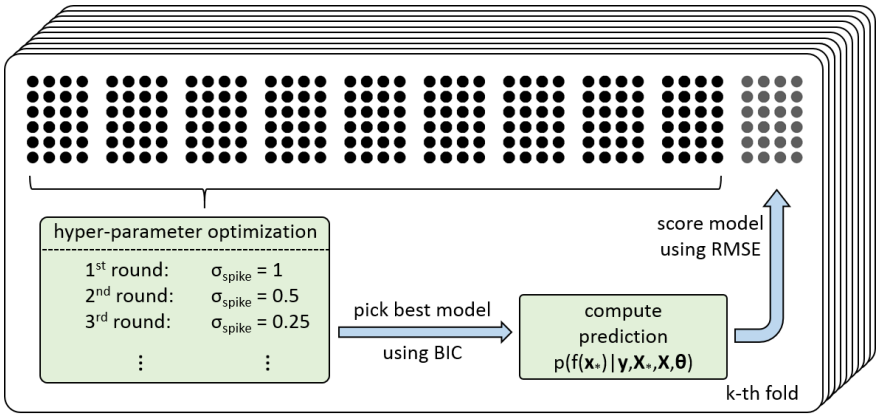


Figure 3: Model fitting procedure: We propose to use 10-fold cross-validation. In each fold, a model is trained on 90 percent of the non-validation data; the model is then scored on the remaining 10 percent. Hyper-parameter optimization is performed using prior shrinkage and the best model according to the BIC selected. For this winning model, the posterior is computed in order to score the model.

3.4 Prediction of Survival

It is possible to predict (time to) an event for individual n which is a result of the observation trajectory $\boldsymbol{y}_n$ passing a certain threshold ω. The individuals are thus split in two groups: those passing the threshold and those that do not. Although this discards most of the useful data in the trajectory itself, GPTP can be used for the purpose of such a binary classification similar to survival models, by thresholding the predicted mean trajectories $\hat{\boldsymbol{y}}_n$ where the mean is given by equation 63. We do not necessarily have to employ the same threshold for observed and predicted trajectories; instead, we can let ω_{pred}, which is the threshold that we apply to predicted mean trajectories $\hat{\boldsymbol{y}}$, be a variable.

By changing ω_{pred}, we effectively trade off sensitivity, which in this context is defined as the ratio of correctly identified survival events and all survival events, for specificity, which is the ratio of correctly identified death events and all death events. This results in a so-called Receiver Operating Characteristic curve, which plots sensitivity against specificity for all values of ω_{pred}. Its integral is known as the area under the curve (AUC); a perfect classifier has an AUC of 1, while a random classifier has an AUC of 0.5.

3.4.1 Majority voting

While the mean CV score $\mathbb{E}[\text{RMSE}(\boldsymbol{y})]$ is a good proxy for the model score on validation data, the models that were produced during CV are usually not being used for prediction of validation data, as the goal is to compute the CV score only. Instead, all of the k partitions are used to fit a new model. This model is applied to the validation data, in order to get the final model validation score. An alternative is to retain the separate CV models, especially if these models differ substantially from another. Predictions may then be obtained by somehow combining the models.

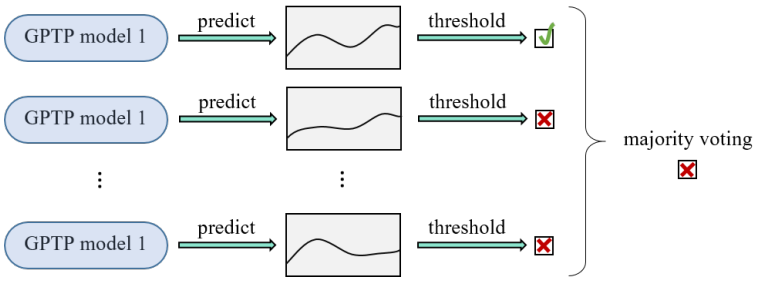


Figure 4: Majority voting for binary classification of validation data.

In section 4, we apply our method to the problem of predicting disease outcome in major depressive disorder. The models that were trained during CV differ quite substantially from another, so we report the average RMSE on validation data. Additionally, the predicted disease trajectories are thresholded. This results in a binary prediction. Situations arise in which the trajectory is very close to the threshold, yet the output must be a binary decision. This might

cause small changes in the model to have a big impact, so we use a majority classification scheme, were the models that were estimated during CV are retained. Applying them to the validation data yields k trajectories per participant, which are then thresholded to make k binary predictions. The final decision is taken to be whatever the majority of models recommends. A sketch of this approach is shown in Figure 4. An important advantage of such model averaging is that it generally leads to improved overall performance, especially if the models are not highly correlated (Bishop, 2006).

3.5 Online prediction

GPTP assumes a multivariate Gaussian distribution of observations; this enables using the method online by conditioning on the outcomes that were already observed. Mathematically, this is essentially equivalent to predicting new $f(\boldsymbol{x}_*)$ as described in section 3.2.2.

Let

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{y_1}\\ \boldsymbol{y_2}\end{bmatrix} | \begin{bmatrix}\boldsymbol{\mu_1}\\ \boldsymbol{\mu_2}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}}\\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}}\end{bmatrix}\right), \tag{68}$$

then the distribution of $\boldsymbol{y_2}$ conditional on $\boldsymbol{y_1} = \boldsymbol{a}$ is multivariate normal with

$$\begin{aligned} p(\boldsymbol{y_2}|\boldsymbol{y_1} = \boldsymbol{a}) &= \mathcal{N}\left(\boldsymbol{y_2}|\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\right)\\ \bar{\boldsymbol{\mu}} &= \boldsymbol{\mu_2} + \boldsymbol{\Sigma_{21}}{\boldsymbol{\Sigma_{11}}}^{-1}(\boldsymbol{a} - \boldsymbol{\mu_1})\\ \bar{\boldsymbol{\Sigma}} &= \boldsymbol{\Sigma_{22}} - \boldsymbol{\Sigma_{21}}{\boldsymbol{\Sigma_{11}}}^{-1}\boldsymbol{\Sigma_{12}}. \end{aligned} \tag{69}$$

In this way the method's predictions can be updated as new observations become available without having to re-estimate the hyper-parameters.

3.6 Feature preselection

We can gain further insight into the GP prior by comparing with the Bayesian perspective of ridge regression. In this standard Bayesian regression model with $y \sim \mathcal{N}(\boldsymbol{X\beta}, \sigma_y^2\boldsymbol{I})$ and prior $\boldsymbol{\beta} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$ the ratio $\lambda = \sigma_y^2/\sigma^2$ is known as the ridge shrinkage parameter. In analogy, for our model we have an unconventional $\lambda = \sigma_y^2\boldsymbol{C}^{-1}$. Thus, in GPTP the mean of the regression weights results as a minimization of the objective

$$\boldsymbol{\mu}_\beta = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\left(\|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \|\sigma_y^2\boldsymbol{C}^{-1}\boldsymbol{\beta}\|_2^2\right). \tag{70}$$

Since we defined $\boldsymbol{C}$ to be block-diagonal where the m-th block equals $\sigma_m^2\boldsymbol{A}$, it follows that a small σ_m^2 results in a high penalty for regression coefficients of the m-th covariate. As each σ_m^2 has a spike and slab prior, which forces values towards zero, this results in automatically selecting out features that are not predictive (in the linear sense) of response variables *of at least one point in time.*

This method of feature selection has its limits, as we show in section 3.7; the model overfits if the total number of features M is too high. In that case, one

possibility is to employ some feature pre-selection before application of GPTP (but after setting aside the validation data). One of the most popular approaches is the elastic net (Zou and Hastie, 2005). Running one elastic net per time point, we minimize the objective

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\|\boldsymbol{y}_t - \boldsymbol{X}_t\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_1 \right), \tag{71}$$

where $\boldsymbol{y}_t, \boldsymbol{X}_t$ contain data from all participants for a single time point t. Since the elastic net does not support missing data, corresponding features / individuals need to be somehow removed. Here, we simply remove features that are present in less than 75 percent of individuals and from the remaining individuals remove the ones with any missing features.

Any covariates that were not selected by any of the elastic nets can safely be discarded, as this implies that no linear functions of these covariates could be found that are predictive of the response variables (with the one caveat that some patients/features had to dropped before doing some of the elastic net regressions).

For completeness, Figure 5 shows a high-level overview of the method that we presented in this section.

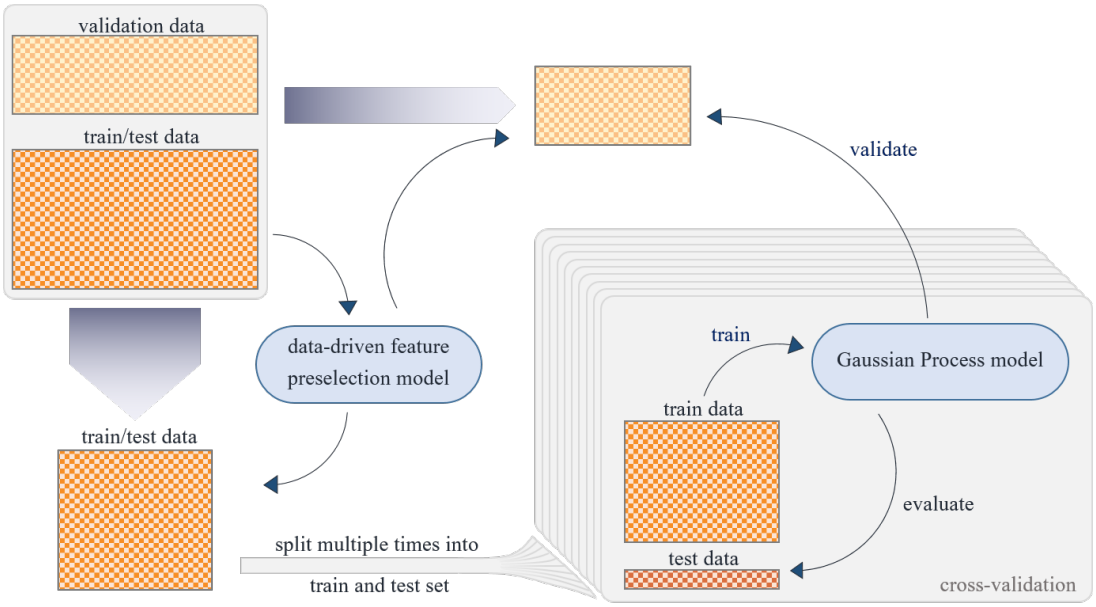


Figure 5: High-level overview of all components of GPTP: After (optionally) applying feature pre-selection via the elastic net on the non-validation data, CV is applied to train many GP models for each CV fold. The best model according to the predictive RMSE is picked for each fold and used to compute predictions and validation scores on the validation data.

3.7 Simulations

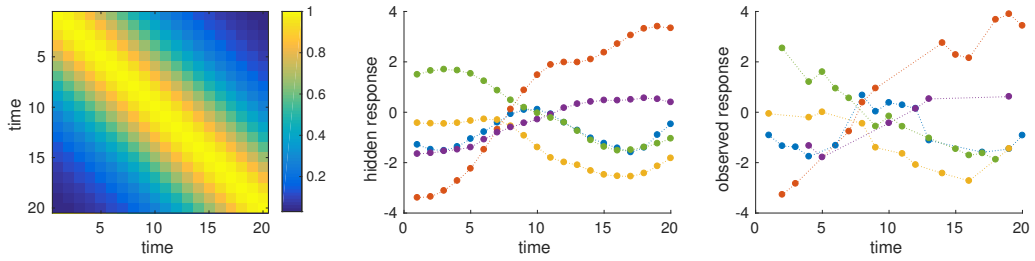


Figure 6: Example of data generated in the first set of simulations. Left: time covariance matrix $\boldsymbol{A}$. Middle: Full response data for 5 simulated participants; this was hidden from inference. Right: Response data that was available for inference; 50 percent of observations were missing, and a moderate amount of observation noise was added.

We designed several sets of simulations in order to test how GPTP performs in the face of critical issues that often arise in the analysis of longitudinal data. Specifically, we performed simulations investigating performance in the case of small datasets, missing data, correlated covariates and non-Gaussian data.

The first set of simulations assessed the model's ability to deal with missing data, while at the same time investigating the amount of data needed to prevent overfitting. The simulation setup was designed such that generated data were close to the kind of data that one could expect in a clinical or psychological longitudinal study. We used the Matérn kernel function with fixed $\nu = \frac{5}{2}$ for the GP prior, so that the kernel only had one free parameter, the length scale ℓ. This is one of the most widely used kernel functions for modeling smooth real-world data (Rasmussen and Williams, 2006), and is described in section 3.2. We performed 100 simulations each for $T = 20$, $M \in \{16, 32, 64\}$ and $N \in \{32, 64, 96, 128\}$, where N indicates both the amount of training and test participants, T indicates the number of time points and M indicates the number of features. In each simulation, 5-fold CV was used to assess training and test error. Further, half the covariates as well as half the marker observations were removed. Feature values were assumed to be measured by a hypothetical questionnaire, such that for each feature we first sampled uniformly from the interval $[2, 8]$ the number of possible values, and then sampled from those values uniformly for the simulated participants. This was done to simulate a real-world multiple-choice questionnaire in which each question had between 2 and 8 answer options. We also repeated all simulations for different feature intervals and also Gaussian features, and obtained identical results. For this set of simulations, we chose $\sigma_m^2 = \sigma^2$ for $m \in [1, M]$, such that feature selection was turned off. Thus, there were a total of 3 hyper-parameters $\boldsymbol{\theta} = (\ell, \sigma, \sigma_y)$, which were drawn randomly from a tight Gaussian distribution around $\ell = 0.4^2, \sigma = \exp(0.4), \sigma_y = \exp(-1)$ for each simulation. Figure 6 shows an example of the generated data. For inference, we

chose wide Gaussian priors on all hyper-parameters, $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, 3^2\boldsymbol{I})$.

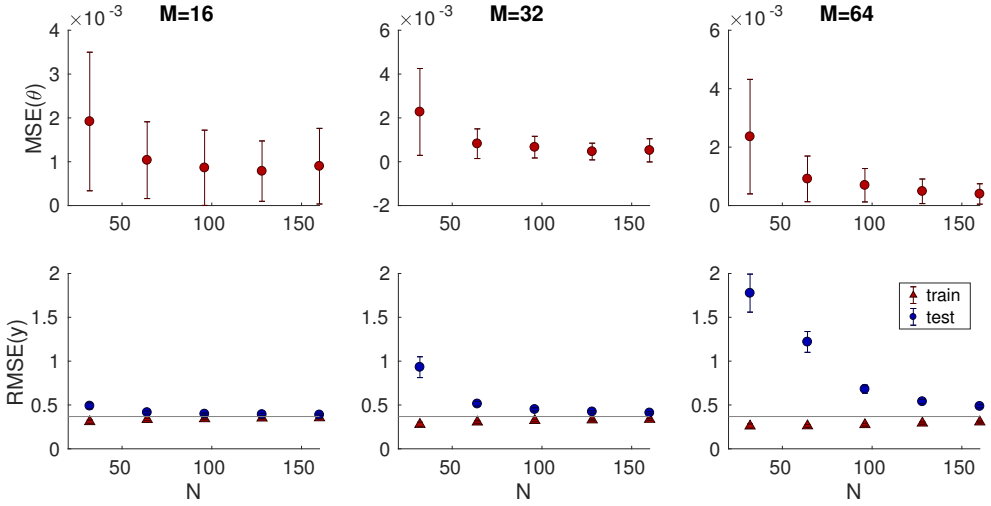


Figure 7: Performance of a GPTP model with three hyper-parameters $\boldsymbol{\theta} = (\ell, \sigma, \sigma_y)$ and a Matérn kernel function. Top row shows mean squared error for hyper-parameters, which were set to $\ell = a^2 = 0.4^2, \sigma = \exp(b) = \exp(0.4), \sigma_y = \exp(c) = \exp(-1)$. The reported errors are with respect to a, b, c, as these were the parameters that the unconstrained gradient optimization was run on. Bottom row shows train (red triangles) and test (blue circles) root mean squared error of observations $\boldsymbol{y}$. The grey line indicates noise level, $\sigma_y = \exp(-1)$.

Results are shown in Figure 7. While hyper-parameters could generally be recovered better for higher N, mean squared error MSE($\boldsymbol{\theta}$) was smaller than 0.004 in all simulations. This was less than one percent of the absolute values of each of the parameters, suggesting that hyper-parameters could be recovered very well in all simulations. We employed the root mean squared error (RMSE) as measure for how well response trajectories were estimated. Test RMSE was high when $M > N/2$ suggesting that although the model was identified well there was simply not enough data to compute the posterior accurately (see equation 56).

In the set of simulations above, the mean feature correlation (after Fisher transform) across all runs was $\bar{\rho} = 0.19$. However, even coefficients of highly correlated features can be estimated reliably: We repeated the simulations again for $\bar{\rho} = 0.4$ and $\bar{\rho} = 0.8$. Higher correlation did not affect results adversely, see Figure 8.

Third, we investigated the issue of non-Gaussian data. While it is possible to explicitly model non-Gaussian likelihoods, this slows down inference, as approximate solutions have to be found through sampling approaches. We simulated data with a heavy-tailed Student t as well as a light-tailed Laplace distribution of residuals and compared that to the standard Gaussian distribution. We chose hyper-parameters $\nu = 3.16$ for the Student t distribution and $\beta = 1.17$ for the Laplace distribution, such that the resulting noise variance $\sigma_y = \exp(0.5) = 1.65$ was the same for all distributions. As shown in Figure 9, non-Gaussian noise did

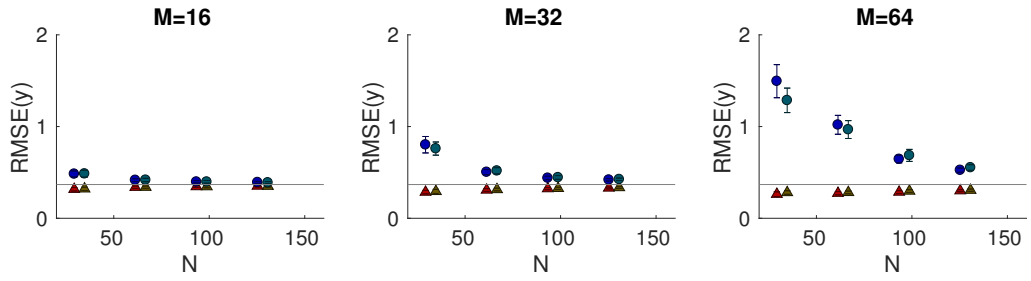


Figure 8: Performance of the non-ARD model with three hyper-parameters for highly correlated variables. Circles mark test errors, triangles mark train errors. Blue and red colors indicate errors for a set of simulations with $\bar{\rho} = 0.4$. Turquoise and brown colors indicate errors for a set of simulations with $\bar{\rho} = 0.8$.

not affect results adversely.

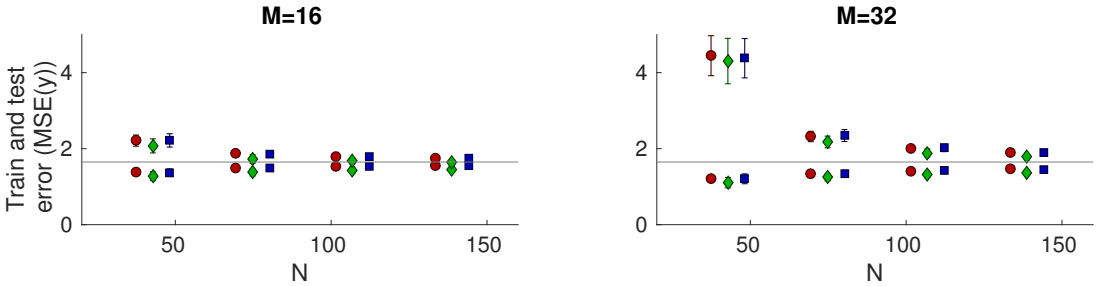


Figure 9: Robustness to likelihood misspecification. Comparison of model performance for Gaussian (red circles), Student t (green triangles), or Laplace data (blue squares). Predictive error was small for $N > 2M$, and was robust with respect to misspecification. The gray line indicates the true noise standard deviation $\sigma_y = 1.65$. Markers below the line correspond to training errors, and above the line to test errors.

In the next set of simulations we evaluated the performance of the full ARD model with a total of $M + 2$ hyper-parameters $\boldsymbol{\theta} = (\ell, \sigma_1, ..., \sigma_M, \sigma_y)$. For the feature variances σ_m, we used a Gaussian spike and slab prior with shrinkage as described in section 3.6. We set half of the feature weight trajectories to zero before generating $\boldsymbol{y}$, such that half the features were non-predictive of outcomes. Figure 10 shows that the method works well for $M > N/2$: hyper-parameters can be recovered well, features are selected correctly, and consequently validation error is close to optimal.

It might be argued that the simulation setup so far was too easy in the sense that the correct kernel function was assumed to be known: inference was carried out using the same (correct) kernel function with which regression weight trajectories were generated, such that 'only' hyper-parameter selection had to be done. The last set of simulations addresses this issue: For $T = 20$ and $M = 18$, half of features were chosen to be non-predictive, as before. For the other 9

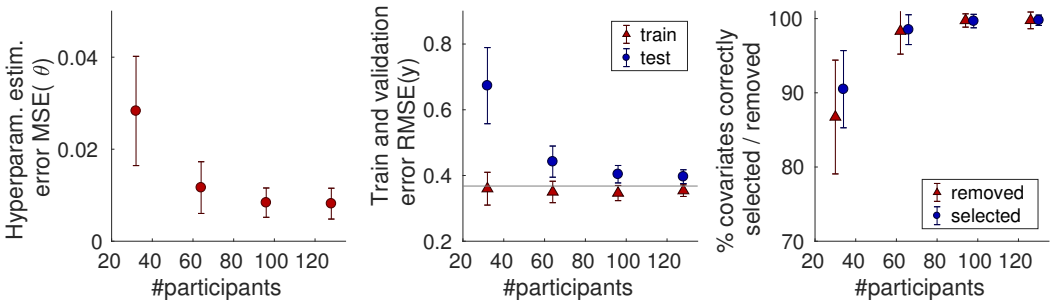


Figure 10: ARD model performance for $M = 32$. Left: MSE of hyperparameter estimation. Middle: RMSE of train (red triangles) and validation (blue circles) observations. The grey line indicates noise level, $\sigma_y = \exp(-1)$. Right: Number of covariates selected correctly by ARD. Red triangles indicate the fraction of truly predictive covariates that were correctly selected, and blue circles the fraction of irrelevant covariates that were correctly removed.

features, we manually designed weight trajectories that differed in general shape, but also specifically with respect to length-scale and variance. Importantly, these trajectories did not arise from a GP at all. We repeatedly estimated trajectories 5 times with $N = 32$, $\sigma_y = \exp(-1) = 0.3679$, and also with $N = 96$, $\sigma_y = \exp(-2) = 0.1353$. All weight trajectories could be approximated closely (see Figure 11), although the amount of data was very limited, and the trajectories differed in shape, amplitude and length scale. This is especially notable since the kernel function's length scale was forced to be the same for all features.

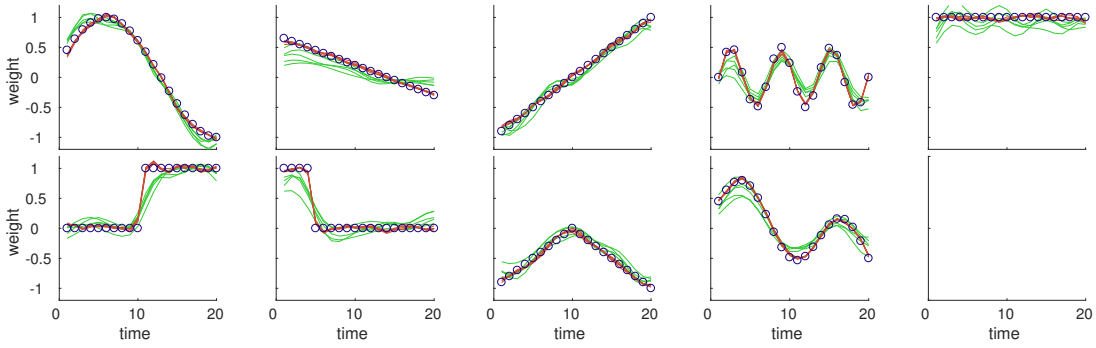


Figure 11: Estimation of the 9 non-GP weight trajectories. Each plot shows one simulated features' weight trajectory (black circles) as well as GPTP estimates for $N = 32$, $\sigma_y = \exp(-1) = 0.3679$ in green and for $N = 96$, $\sigma_y = \exp(-2) = 0.1353$ in red. Estimation was repeated 5 times, resulting in 5 lines each. Note that the Matérn kernel function was able to recover all weight trajectories very closely, although kernel length scales were fixed to be the same across covariates, and amount of data was limited.

4 Prediction of longitudinal outcomes in depression

Major depressive disorder (MDD) is a leading cause of morbidity and mortality worldwide (Ferrari et al., 2013). Currently, it is already the fourth leading cause of disability, and is projected to become the second by the year 2030 (Mathers and Loncar, 2006). It is a disconcertingly common (striking about one out of six people over their lifetimes) and often highly debilitating disorder and has severe consequences on quality of life for the majority of patients: not only is it chronic and recurrent, but also current lines of treatment are reported as inadequate by more than three quarters of those suffering from MDD (Ebmeier et al., 2006).

Despite its high prevalence, the disorder is very imprecisely characterized and much less understood. It is primarily associated with a persistently low mood, which can lead to feelings of low self-esteem, a lack of enjoyment for normally pleasurable activities (anhedonia) and is often accompanied by loss of appetite and feelings of helplessness, pessimism, decreased motivation and anxiety. Disturbances in homeostatic functions are common, too, and include insomnia, loss of appetite or loss of libido (Bentall and Beck, 2005). The content of depressive ruminations (an example of a maladaptive strategy of emotion regulation) is neatly summarized by Beck's cognitive triad; the self is worthless, life is pointless, the future hopeless (Beck, 1967).

Currently, there are two major lines of treatment: antidepressant medication (ADM) and psychotherapy. 70% of patients will eventually respond at least partly to treatment (Rush et al., 2006). About half of these achieve full remission, defined as absence of disease activity. The other half continues to suffer from a reduced level of symptoms. They are said to have achieved partial remission. Rates of relapse - the reappearance of a disease - are high: around 50% of people who achieve remission, relapse within 12 months. Unfortunately, however, most remitted patients choose to discontinue their medication, particularly after having been well for a certain time; among these, 40% relapse within three months (DeRubeis et al., 2008).

ADMs have shown convincing efficacy in reducing the risk of relapse or recurrence. Several meta-analyses have estimated that continuation of antidepressants reduces the odds of a relapse by around 70% (Viguera et al., 1998; Geddes et al., 2003; Glue et al., 2010). Therefore, the general recommendation is to continue medication for about three months after the first episode, but for very long periods after a few episodes. Furthermore, the patients that have suffered already from two or three episodes are the ones at highest risk of the more debilitating type of depression and the corresponding chronic symptoms, making them clinically a highly interesting target for research.

As a first step, purely descriptive machine-learning approaches should be applied to combine existing features and produce predictors of an individual's remission / relapse risk (Forand and DeRubeis, 2014). This is precisely our aim in this section: we apply GPTP to a large dataset of patients suffering from major depressive disorder, and investigate both remission during treatment, as well as

relapse during a 12 month follow up period. While prediction of remission has been attempted on these data before (Chekroud et al., 2016), we are not aware of any such attempt for the prediction of relapse. A possible next step could be to apply this approach to the differential prediction under pharmacological vs psychotherapeutic treatment.

4.1 STAR*D data

The data are from a large, multicentre clinical trial of major depressive disorder (STAR*D, ClinicalTrials.gov, number NCT00021528, see Rush et al. (2006)). This is the largest randomised controlled study of treatment in major depressive disorder to date. Patients, who were all adults aged 18-75 and had a primary clinical diagnosis of non-psychotic major depressive disorder, were recruited in the USA between 2001 and 2004. In a first stage, 3671 patients were treated with the Serotonin Re-Uptake Inhibitor (SSRI) Citalopram for 12-14 weeks; during this period an index of depression, the Quick Inventory of Depressive Symptomatology (QIDS; Rush et al. (2003)), was measured every two weeks. QIDS scores are integers between 0 and 23. Those 1475 patients who remitted (this was defined as no longer suffering from an acute depressive episode) entered a follow-up phase of one year during which QIDS scores were acquired every month. The medical recommendation was to continue antidepressant treatment during this period; however, this was not monitored and it is unknown whether, when and to what extent patients might have stopped taking the medication.

The 1430 patients who did not remit after 12-14 weeks entered a second stage in which they were randomly re-assigned to another medication or to psychotherapy. The study consisted of a total of five stages, but we focus here on the first stage only, as well as the follow-up period. Part of the total STAR*D trial flow-chart is shown in Figure 12. Amongst other results, the study showed that remission rate in the first stage was 36.8 percent, and the most successful strategy in the second stage was to switch to cognitive therapy, which resulted in a remission rate of 41.9 percent in the remaining patients (with the caveat that only $N = 62$ patients were assigned to this arm).

Since the acute treatment phase was 12-14 weeks with the first bi-weekly QIDS measurement at the beginning of the first week (=baseline), a total of 7-8 scores should have been collected per patient. However, the mean number of observations per participant was 3.99 ± 0.66, indicating lots of missing observations. During follow-up, on average 7.13 ± 3.62 scores were collected. Many patients, however, did not complete this phase, resulting in a high drop-out ratio, see Figure 13. The clinical criterion for remission during stage one was a QIDS of 5 or lower at the end of the treatment phase. The criterion for relapse back into another depressive episode during follow-up was a QIDS of 11 or higher at any time.

We investigate the prediction of the disease trajectory as a linear combination of feature weight trajectories during acute treatment, addressing the question of treatment response, and consequently remission prediction. Second, we

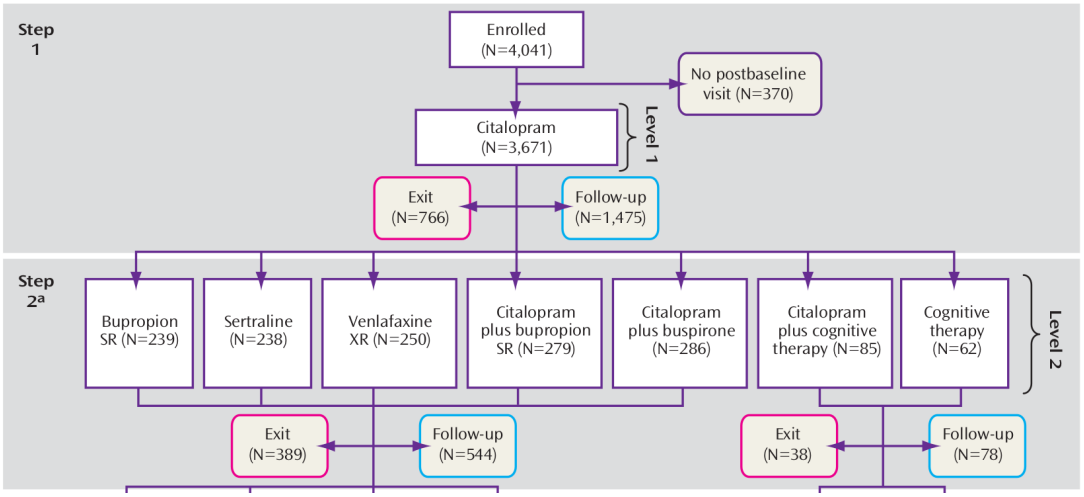


Figure 12: First two stages in the STAR*D study (Rush et al., 2006). Reprinted with permission from the American Journal of Psychiatry, (Copyright ©2006). American Psychiatric Association.

also predict the trajectory during follow-up, addressing the question of *sustained* treatment response, and consequently relapse prediction.

4.2 Pre-processing

4.2.1 Criteria for participants

We developed the GPTP model on 70% of the data, which we repeatedly used for cross-validating the models as described in section 3.3. The remaining 30% were set aside as validation data, and only used at the very end to determine generalizability of the model to previously unknown cases, as described in section 3.4.

We excluded all patients who did not finish the treatment phase in 12-14 weeks, as this was the treatment duration defined in the STAR*D study protocol. Additionally, we excluded any participants who did not finish the treatment phase. This resulted in a total of 1214 patients during treatment (after dropping 30 percent of the data to be used for validation), of whom 591 remitted at the end of treatment, and 623 did not remit. During follow-up, a total of 632 patients remained, of whom 229 relapsed and 403 did not relapse.

Although in the study protocol non-remission was an exclusion criterion for entering the follow-up phase, in fact 190 out of 632 patients who entered the follow-up were not fully remitted (see Figure 14). Note that while the criterion for remission necessarily required patients to remain in the study until the end of the treatment period, and thus there were no cases of drop-outs, this was not the case for the follow-up phase. In fact, the majority of patients dropped out before the end of the 12-month interval, see Figure 13. We also discarded all observations during follow-up that exceeded a follow-up duration of one year, making the longest possible observation period a total of 66 weeks (14 weeks of

treatment plus 52 weeks of follow-up). Last, we standardized QIDS scores to a mean of zero and a standard deviation of one. Note that this did not remove any trend in the data, as we standardized across all time points.

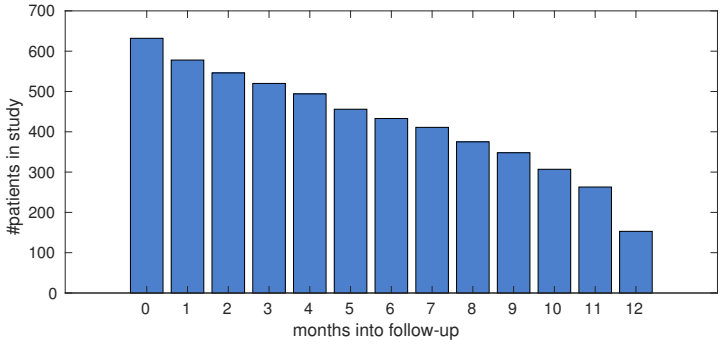


Figure 13: During follow-up, an average of 39.9 patients dropped out every month, so that only 153 out of 632 patients remained at the end of the follow-up period.

We note that, unlike suggested by the specifications in the study protocol, 165 participants had a QIDS score of less than 11 at the beginning of the study (i.e. they did not fulfill the study criterion for acute major depression), and 190 participants had a QIDS score greater than 5 at the end of level 1 prior to entering the follow-up phase (i.e. they did not fulfill the study criterion for remission), see Figure 14. Nevertheless, these data are suitable for the current purpose of predicting temporal trajectories and where hence not excluded.

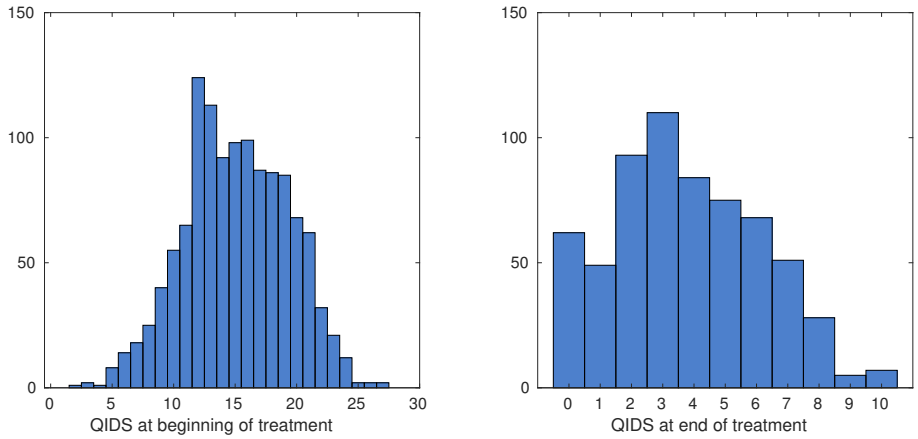


Figure 14: The left plot shows a histogram of QIDS scores at the beginning of the treatment period for all patients that we included in our analyses. Note the peak of QIDS scores just over the inclusion threshold. The right plot shows the scores at the beginning of the follow-up period. We can see that while none of the patients started the follow-up with a QIDS > 10, which is the clinical criterion for relapse, almost half of the patients were not remitted (QIDS > 5) either.

4.2.2 Feature engineering

We examined separately the prediction of trajectories during treatment and follow-up, respectively. For both prediction of remission as well as relapse, we included all available questionnaire measures at baseline; they are summarized in table 1. All items were included for training the two separate GPTP models for prediction of treatment success and prediction of relapse. Specifically for prediction of relapse we included the difference T2-T1 between the two measurements of each questionnaire item for all questionnaires that were available both at beginning (T1) and end (T2) of the acute treatment phase.

Table 1: Questionnaire measures available in the STAR*D trial. Questionnaires in italics were available both at beginning and end of the treatment period, others were only available at the baseline.

Cumulative Illness Rating Scale
Demographics Form
Hamilton Rating Scale for Depression (HRSD)
Patient-Rated Inventory of Side Effects
Protocol Eligibility
Psychiatric Diagnostic Screening Questionnaire
Psychiatric History
Quality of Life Enjoyment and Satisfaction Questionnaire
Quick Inventory of Depressive Symptomatology - Self Rated (QIDS-SR)
Quick Inventory of Depressive Symptomatology - Clinician Rated (QIDS-C)
Screening form
Short Form Health Survey
Side Effects Form
Work and Social Adjustment Scale Depression
Work Productivity and Activity Impairment

From the individual QIDS items, we calculated the QIDS subscales sleep / weight and psychomotor for the QIDS scores at baseline and beginning of follow-up and included those as additional features (Rush et al., 2003).

Some features required additional pre-processing: We set the maximum number of major depressive episodes (MDE) to 4, as some participants had a very high number of reported MDE (some more than 30). This feature - like most - was self-reported, and hence may not have adhered to standard definitions of episodes, as often patients overestimate the number of episodes. Further, we passed the features "onset of current MDE", "Monthly household income" and "Monthly employment income"' through the square root function, as we expected a non-linear effect. Last, we calculated the change in monthly household/employment income between baseline and follow up and converted this to a categorical variable, where -1 indicated a decreased income and +1 indicated an increased income.

Next, we calculated an approximate true drug initial response pattern. Pre-

vious research (e.g. Andrews et al. (2011)) suggests that pharmacological effects may explain half of the improvement observed in medication response while non-specific or placebo effects explain the other half. True drug efficiency in the so-called delayed persistent responders could be measured by a true *drug initial response pattern* because delayed persistent responses are three times more likely to occur on drug than on placebo. The pattern is essentially a late improvement after a 2-week delay and stable, non-fluctuating, course. This has been shown to be predictive of relapse (Quitkin et al., 1984; Andrews et al., 2011). Persistence was defined as an improvement that was not followed by worsening in the Clinical Global Impression scale. Delayed responses were defined as improvement that was first manifested after two weeks of treatment (thus, in week three or later). However, if there was no improvement in the first six weeks, we can safely assume that the drug did not have any effect. Looking at the first 6 weeks of treatment, the group of delayed-onset persistent responders is then characterized by the codes 001111, 000111, 000011, where 0 indicates an unimproved week and 1 indicates an improved week. Missing weeks were filled by assigning the same improvement score as the bracketing weeks, or a 0 if the bracketing weeks differed. Since in the STAR*D data set, QIDS scores were collected only every two weeks, so we approximated true drug response by using the patterns 011 and 001 for a delayed-onset persistent responder, where we defined an improved week as a QIDS score improvement of at least 3.

We also standardized each feature by subtracting the minimum feature value, so that the resulting lowest feature value was zero, and to have unit variance; this was done so that interpretation of estimated trajectories would be easiest (because lower feature value would then indicate lower influence of trajectory; this would not have been the case if we had subtracted the mean). Overall, a total of 403 and 555 baseline features were available for prediction of remission and follow-up, respectively.

4.3 Feature pre-selection

As we have shown in section 3.7, accurate estimation of posterior weights (and consequently prediction of trajectories) breaks down for $M \gtrapprox N/2$. Unfortunately, the number of features during treatment was close to this threshold, and above it for the follow-up period, so we decided to employ a feature pre-selection procedure as discussed in section 3.6. Across all elastic net regressions, a total of 211 out of 403 features remained for the treatment period, and 183 out of 555 features remained for the follow-up period. This reduction was big enough such that we could train the Varying Coefficient model on these remaining features.

Although we have shown in section 3.7 that feature correlations do not present a problem for GPTP, they do impact on the robustness of parameter estimates and thus the interpretability of the method when being run on different partitions of the data: For example, during cross-validation, models estimated on different folds might select one or the other of two highly correlated variables, changing the overall predictive error only slightly but resulting in different parameter / weight

estimates. Figure 15 shows the correlation matrices. The median correlation was relatively small (0.052 during treatment and 0.047 during follow-up). Although some of the features were highly correlated, we chose to not exclude them, cognizant that this came at the potential cost of more difficult interpretability.

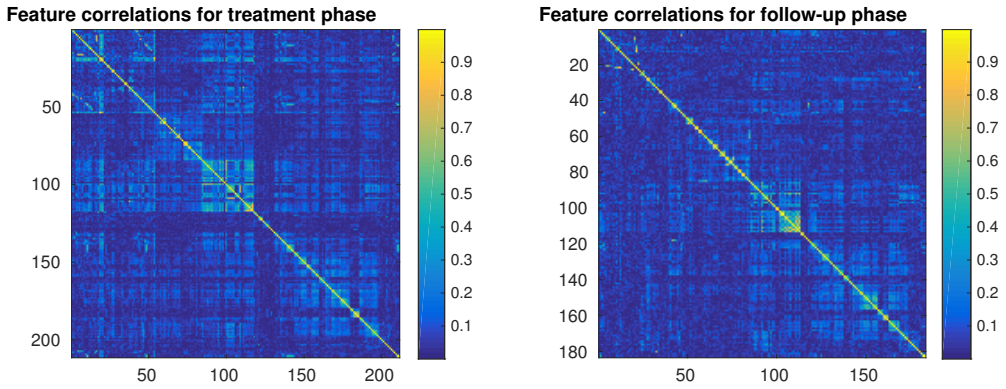


Figure 15: Feature correlations for all features that remained in the design matrix after pre-processing. While some features were highly correlated, this was not the case on average. The mean and median correlation (after Fisher transform) of the baseline variables used to predict treatment response were, 0.094 and 0.052. For the prediction of the follow-up data, the mean and median correlation were 0.087 and 0.047.

Table 2: Online validation RMSE in treatment phase, when conditioning on the first $t = 0, 1, 2, 3, 4$ observations as discussed in section 3.5. The first column shows the RMSE values for all observations after time point t. Thus, the first value in this column indicates the total RMSE on validation data. The second column contains RMSE values for all observations at time points later than t after conditioning on the first t time points.

	RMSE	Online RMSE
t=0	0.683 ± 0.023	-
t=1	0.740 ± 0.012	0.683 ± 0.022
t=2	0.754 ± 0.011	0.682 ± 0.023
t=3	0.761 ± 0.011	0.681 ± 0.022
t=4	0.772 ± 0.011	0.679 ± 0.021

4.4 Prediction of remission

4.4.1 QIDS trajectories

We assessed root mean squared error RMSE($\boldsymbol{y}$) of the *standardized* QIDS validation scores, meaning that a constant prediction of zero everywhere would result in an RMSE of 1. All reported scores on validation data are averages of predictions from the different models obtained through CV.

The method resulted in an RMSE of 0.683 on the validation dataset, implying that $100(1 - 0.683^2) = 53.4$ percent of the variance could be explained. Table 2 summarizes predictive RMSE as well as *online* RMSE after conditioning on the first 1,2,3 or 4 time points. As expected, online prediction results always improve on the results obtained without conditioning on previous time points. The reduction in RMSE can mostly be attributed to changes in prediction of the first time points after conditioning; Figure 16 shows some example trajectories. Also, the results in table 2 clearly show that observations at later time points can be fit less well, as the RMSE increases when discarding the first t time points. The effect of conditioning is strong enough that it reverses this pattern: Online RMSE decreases slightly with the number of observations that we condition on.

We compared our method to several other approaches. In order not to conflate the comparison results with the kind of model averaging that we performed when predicting validation data, we report here results on the test data of the different CV folds. In table 3, our method (which uses the Matérn kernel) is denoted by GPTP_{MAT}. The model GPTP_{CON} employs a constant kernel (where all entries are equal to one) to estimate trajectories that are time-invariant. This is equivalent to doing linear regression on the averages of individual patients' QIDS scores, and can thus show if prediction of trajectories in fact provides any advantage over a simple constant mean prediction.

As opposed to the global smoothing possible with the Matérn kernel, where observations at all time points may in principle influence each other, a local smoothing method can be formulated as a GPTP problem by using a different kernel function $k(\cdot, \cdot)$, where $k_(t, t) = 1$, $k(t, t') = \rho$ for $|t-t'| = 1$ and $k_m(t, t') = 0$

for $|t - t'| > 1$. As this recasts the dynamic linear system which is solved by the Kalman filter in GPTP terms, we denote the corresponding method GPTP_{KAL}.

We further compared to standard Gaussian Process Regression models that implement GP regression from some $\boldsymbol{Z}$ onto $\boldsymbol{y}$. For model $\text{GP}_{\text{x,t}}$ we have $\boldsymbol{Z} = [\boldsymbol{Z}_1^\top, ..., \boldsymbol{Z}_N^\top]^\top$ and the t-th row of $\boldsymbol{Z}_n$ equals $[\boldsymbol{x}_n, t]$. This model finds some non-linear mapping from a combined space of time and covariates to outcomes, and is thus difficult to interpret. For model GP_{t} the t-th row of $\boldsymbol{Z}_n$ equals $[t]$. This last model completely ignores the features, taking only time into consideration, and thus results in estimation of the mean trajectory across all participants.

Examination of Table 3 shows that GPTP_{CON} performed worst. This was the only model to disregard effects of time completely, clearly showing the need for a time-varying model. The other 'averaging' model, GP_{t}, performed better, but was itself outperformed by the other three approaches, indicating that there was substantial individual variation in the trajectories which could at least partly be captured by the more flexible models. The classical Gaussian Process Regression using both covariates and time, $\text{GP}_{\text{x,t}}$, was yet outperformed by both time-varying versions of GPTP, and finally GPTP_{MAT} outperformed the local smoother GPTP_{KAL} slightly, indicating that there are at least small effects that can not be explained by simply taking into consideration neighboring time points.

4.4.2 Treatment outcome

Although GPTP is not designed to predict binary outcomes, but rather continuous longitudinal trajectories, we nevertheless examined prediction of the treatment outcome "remission" (QIDS score of 5 or less) by thresholding the last predicted score during treatment (i.e. at week 14) of model GPTP_{MAT}. We report balanced accuracy, which is calculated as the mean of sensitivity and specificity. The reported results on the validation data are obtained by majority vote from the individual predictors, as described in section 3.4.1. The RMSE of 0.683 on the validation data translated into a balanced accuracy of 61.64 percent. When using the best possible QIDS threshold for classification according to the ROC

Table 3: Comparison of test RMSE for different models on treatment phase of STAR*D data. Subscripts denote the kernel being used (MAT:Matérn, CON: constant, KAL: kernel equivalent to Kalman filter). GP denotes a standard GP regression using the subscripted instances as covariates. See text for further explanation.

$\mathbf{GPTP_{MAT}}$	0.676 ± 0.043
$\mathbf{GPTP_{KAL}}$	0.693 ± 0.034
$\mathbf{GP_{x,t}}$	0.708 ± 0.022
$\mathbf{GP_{t}}$	0.849 ± 0.039
$\mathbf{GPTP_{CON}}$	0.950 ± 0.042

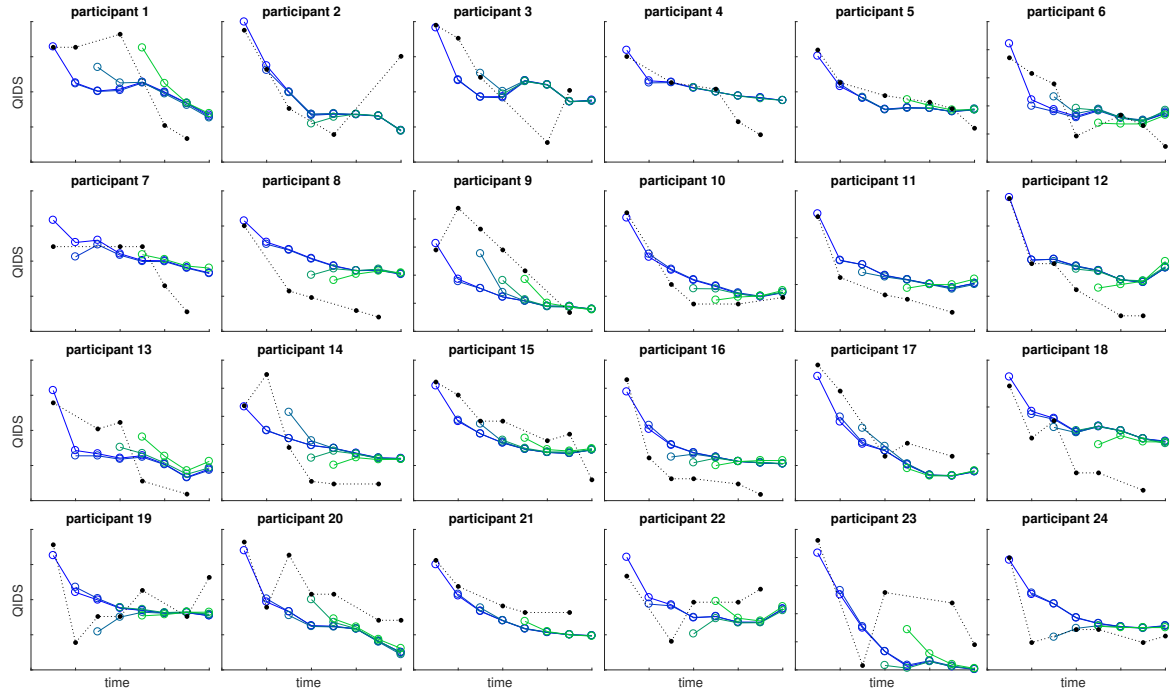


Figure 16: Some examples of predicted disease trajectories during treatment. Each plot shows one participant, black dots are actual measured QIDS values, and circles are predictions. Trajectories that were obtained without conditioning are shown in blue, and after conditioning on the first 1,2,3 or 4 observations in green with stronger coloring indicating conditioning on more observations.

Table 4: Prediction of remission. Results for the best model obtained by the prior shrinkage procedure, as selected by BIC. The scores marked with a star are the ones obtained by using the best possible classification threshold on the training data, according to the ROC curve.

	train	**test**	**validation**
rmse	0.664 ± 0.019	0.676 ± 0.043	0.683 ± 0.023
accuracy	63.35	62.49	63.98
accuracy*	66.62	66.31	68.14
sensitivity*	69.98	71.99	68.57
specificity*	63.26	60.63	67.71

curve shown in Figure 17, this improved to 68.14 percent for a QIDS threshold of 7, with a sensitivity of 68.57 percent and a specificity of 67.71 percent (see Table 4). For comparison, Chekroud et al. (2016) used an approach specifically to predict binary remission outcomes and achieved a balanced accuracy of 64.50 percent. Our improved accuracy results in a number to treat (patients needed to treat in order to prevent one case of sickness) of 18 in comparison to the method by Chekroud et al. (2016), which is the next-best known method.

4.4.3 Performance versus sparsity

We evaluated how well BIC values predicted test RMSE during CV (see Figure 18). Across all CV folds and shrinkage steps, the correlation between the two scores was 0.887, indicating that indeed BIC was a good proxy for generalization performance. Figure 18 also shows correlations between the number of features for each of the models and test RMSE. This was of interest to us because ideally we would have liked to pick a relatively low threshold number of features above which model performance does not deteriorate much. However, no such threshold exists, as the test RMSE steadily increases whenever features are dropped from a model.

4.4.4 Weight trajectory estimates

Across all CV folds, a total of 170 of 211 features were selected. Although none of the selected features can be dropped without severely affecting model performance, most of the corresponding feature trajectories stay close to zero all the time: There were only 88 features whose trajectories deviated from zero by at least 0.05 ($\text{any}(|\boldsymbol{\beta}_k|) > 0.05$), and only 22 features for which $\text{any}(|\boldsymbol{\beta}|) > 0.1$. These are shown in Figure 19. We can see that while all models share many similar tendencies, they also vary quite substantially, with some features selected only in some of the models (for the other models, their posterior weight trajectories become zero everywhere).

The 4 features whose trajectories deviated from zero by more than 0.15 in any of the models are: QIDS-SR at baseline, Thinking that one is controlled by some force, seeing/hearing things other people do not, and days that have passed

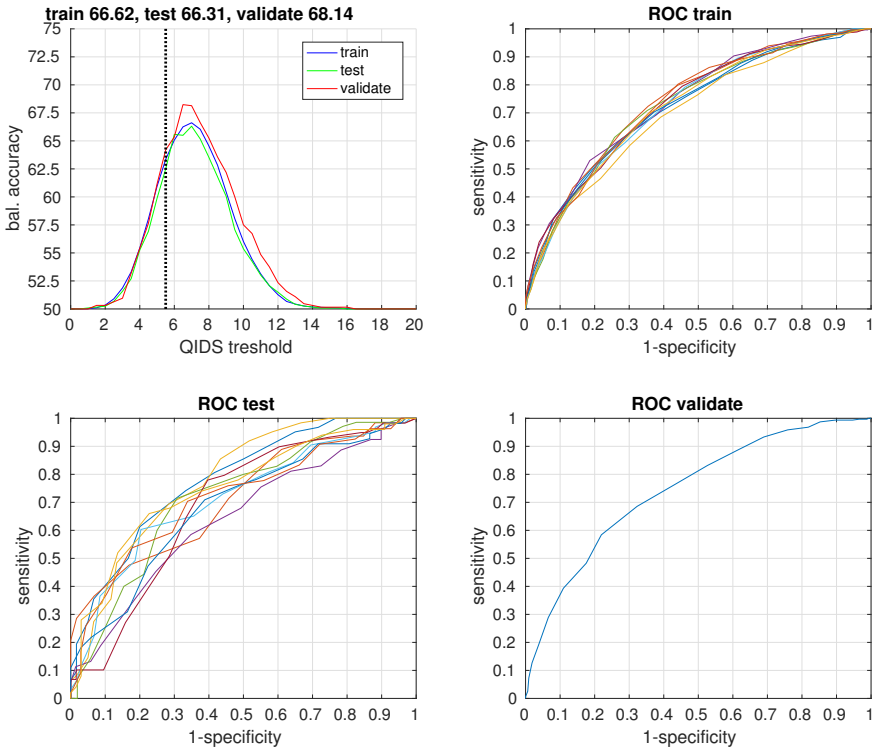


Figure 17: Top left: Remission classification accuracy versus QIDS threshold ω_{pred}. The vertical dotted black line marks the clinical criterion for remission. The threshold optimizing training accuracy is a QIDS of 7; the same threshold also improves test and validation accuracy. The top right (lower left) plot shows ROC curves for the predictors obtained from the different CV folds on train (test) data. The lower right plot shows the validation ROC curve obtained through majority voting from the 10 different predictors. Area under the curve for validation data was 0.745.

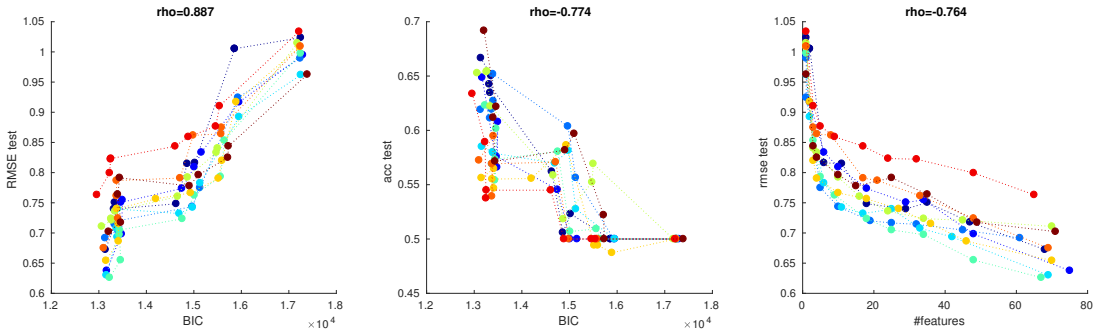


Figure 18: Left: Plot of BIC versus test RMSE. Each dot marks results from one shrinkage step. Dots of the same color, connected by dotted lines, indicate results from the same CV fold. Middle plot shows BIC versus test accuracy. Right: Number of features used for each model versus test RMSE. Models with fewer features generally perform much worse.

since the patients presented themselves for the first time at baseline. The last item refers to the time difference between accepting a patient to the study and start of treatment / measurement. These features stand out clearly in Figure 19.

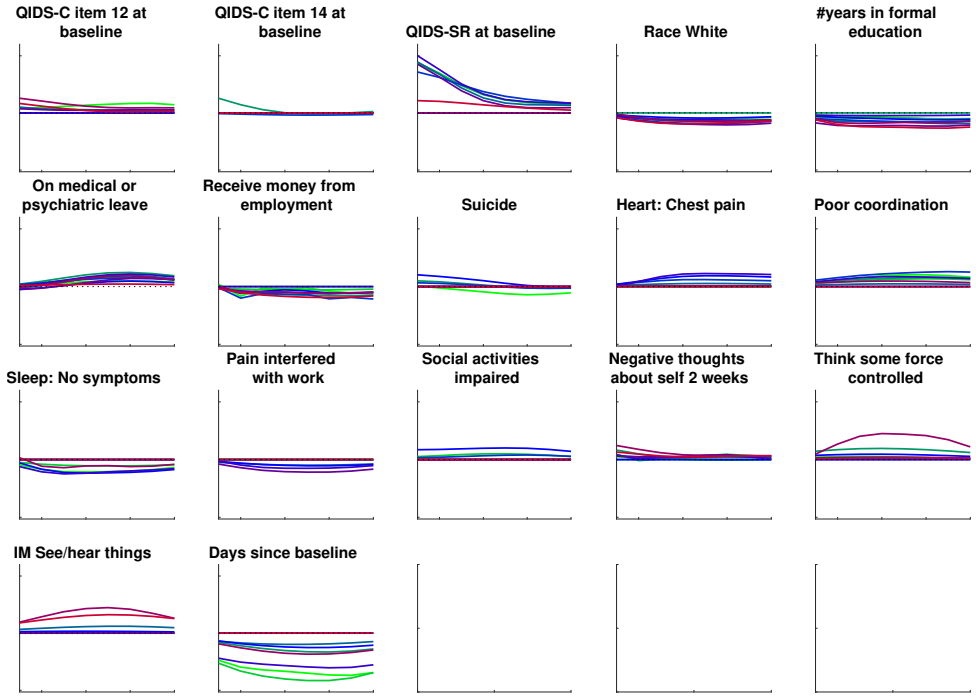


Figure 19: Posterior weight trajectories during treatment for all features for which any($|w|$) > 0.1. Each line corresponds to one CV fold. Average number of selected features per fold was 68.5 ± 3.7.

Interpretation is complicated by the fact that results differ between folds. However, a few clear trends emerge: First, the total QIDS-SR (self-rated) at baseline - and in two cases also the QIDS-C (clinician-rated) Item 14, which measures level of fatigue - are strong predictors early on. This makes obvious sense, as the predicted value itself is the QIDS-SR (self-rated). Second, and perhaps surprisingly - the trajectories for the features 'days since baseline' indicate that in the presence of all other covariates - delayed start of treatment reduces QIDS scores, i.e. the longer treatment was delayed, the *better* patients got during treatment. Being of white race, having spent many years in formal education, having no sleep symptoms and receiving money from unemployment generally reduce QIDS scores, i.e. they cause patients to get better. In some CV folds, psychotic symptoms 'Think some force controlled' and 'IM See/hear things' have a positive effect on QIDS trajectories in the middle of the treatment period; this effect diminishes as time progresses, and can be interpreted in the way that patients who suffer from these symptoms specifically experience a marked benefit from medication towards the end of the first third of the treatment period.

4.5 Prediction of relapse

4.5.1 Clustering according to residual symptom domains

Nierenberg et al. (2010) investigated residual symptoms during follow-up in STAR*D. Amongst others, they found that when clustering participants according to residual symptom severity, the corresponding (Kaplan-Meier) survival curves differed between clusters, with higher residual severity indicating lower survivability. They calculated the number of residual symptoms based on the QIDS in the following way: Each questionnaire item ranges from 0 to 3, so that a threshold of greater than zero recognizes even mild symptoms. Using this threshold, items were grouped into the 9 symptom domains as defined by the Diagnostic and Statistical Manual of Mental Disorders (4th Edition): sleep disturbance, sad mood, appetite/weight, concentration, outlook, suicidal ideation, involvement, energy/fatigue, psychomotor disturbance. Participants had a minimum of 0 residual domains and a maximum of 6, with the most frequent being sleep disturbance (71.7%), appetite/weigh disturbance (35.9%) and sad mood (27.1%).

The probability of relapse was shown to increase with the number of residual symptom domains. This is reflected in the QIDS trajectories of individual participants: the more symptom domains, the higher the average QIDS, see Figure 20. Our model predictions were able to pick up these patterns, showing that the model could capture important parts of the structure (see Figure 21).

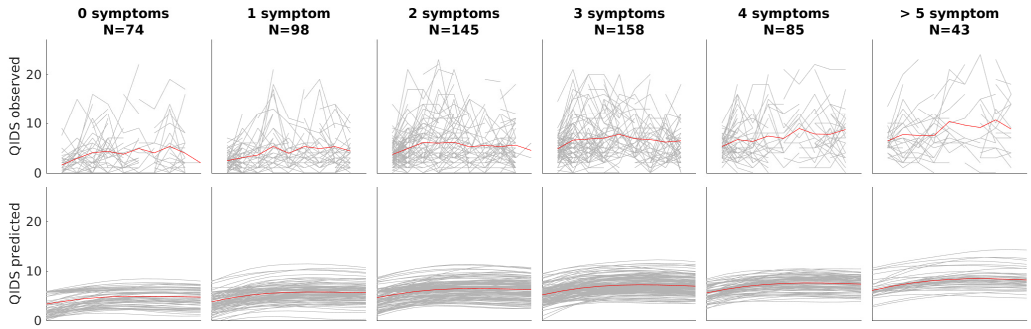


Figure 20: Trajectories for each participant (grey lines) and mean trajectories (red lines) for actual observations (first row) and predicted trajectories (second row), split by number of symptom domains.

4.5.2 QIDS trajectories

For the QIDS trajectories we were able to obtain a validation RMSE of 0.790, meaning that $100(1 - 0.79^2) = 37.6$ percent of the variance could be explained. This is much less than the 53.4 percent that could be explained during treatment. One reason for this difference is that while there is a clear downward trend in the QIDS scores during treatment, which was explained well on average, no such clear trend is visible in the follow-up period (see Figure 22).

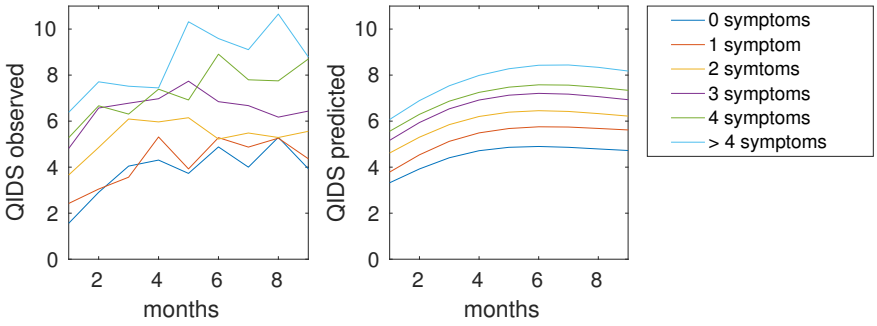


Figure 21: Mean trajectories from Figure 20, collapsed across columns. Similar to results reported in Nierenberg et al. (2010) we find that average QIDS scales with residual symptoms in domains (left plot). The trajectories predicted by the model capture this trend nicely (right plot).

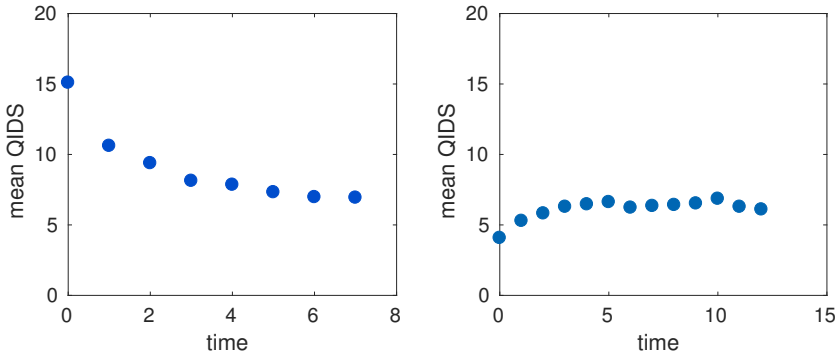


Figure 22: Left plot: Mean QIDS during treatment phase. Right plot: Mean QIDS during follow-up.

Table 5: Online validation RMSE during follow-up phase, when conditioning on the first $t = 0, 1, 2, 3, 4$ observations. The first column shows the RMSE values for all observations after time point t. Thus, the first value in this column indicates the total RMSE on validation data. The second column contains RMSE values for all observations at time points later than t after conditioning on the first t time points.

	RMSE	Online RMSE
t=0	0.790 ± 0.024	-
t=1	0.894 ± 0.014	0.829 ± 0.013
t=2	0.906 ± 0.014	0.830 ± 0.013
t=3	0.909 ± 0.014	0.823 ± 0.012
t=4	0.920 ± 0.013	0.820 ± 0.012

Table 5 summarizes predictive RMSE as well as *online* RMSE after conditioning on the first 1,2,3 or 4 time points. Similar to the treatment period, RMSE grows after discarding time points, while conditioning on those time points effectively reverses this pattern, such that online RMSE of the remaining time points in fact decreases. Figure 23 shows some example trajectories before and after conditioning.

Model comparison results are similar to results obtained from prediction of remission, with GPTP_{MAT} clearly being the best model, see Table 6.

4.5.3 Treatment outcome

We examined prediction of the binary treatment outcome "relapse", which was defined as a QIDS score of 11 or higher at any point during follow-up. The RMSE of 0.790 on the validation data translated into a balanced accuracy of 55.38 percent. When using the best possible QIDS threshold for classification according to the ROC curve shown in Figure 17, this greatly improved to 71.28 percent at a QIDS threshold of 6 (see Table 7). This is notable as it suggests that prediction of remission is more difficult than prediction of relapse.

Table 6: Comparison of test RMSE during follow-up phase for different models on STAR*D data. Subscripts denote the kernel being used (MAT:Matérn, CON: constant, KAL: kernel equivalent to Kalman filter). GP denotes a standard GP regression using the subscripted instances as covariates.

$\mathbf{GPTP_{MAT}}$	0.785 ± 0.047
$\mathbf{GP_{x,t}}$	0.845 ± 0.072
$\mathbf{GPTP_{KAL}}$	0.891 ± 0.078
$\mathbf{GPTP_{CON}}$	0.933 ± 0.069
$\mathbf{GP_{t}}$	0.979 ± 0.045

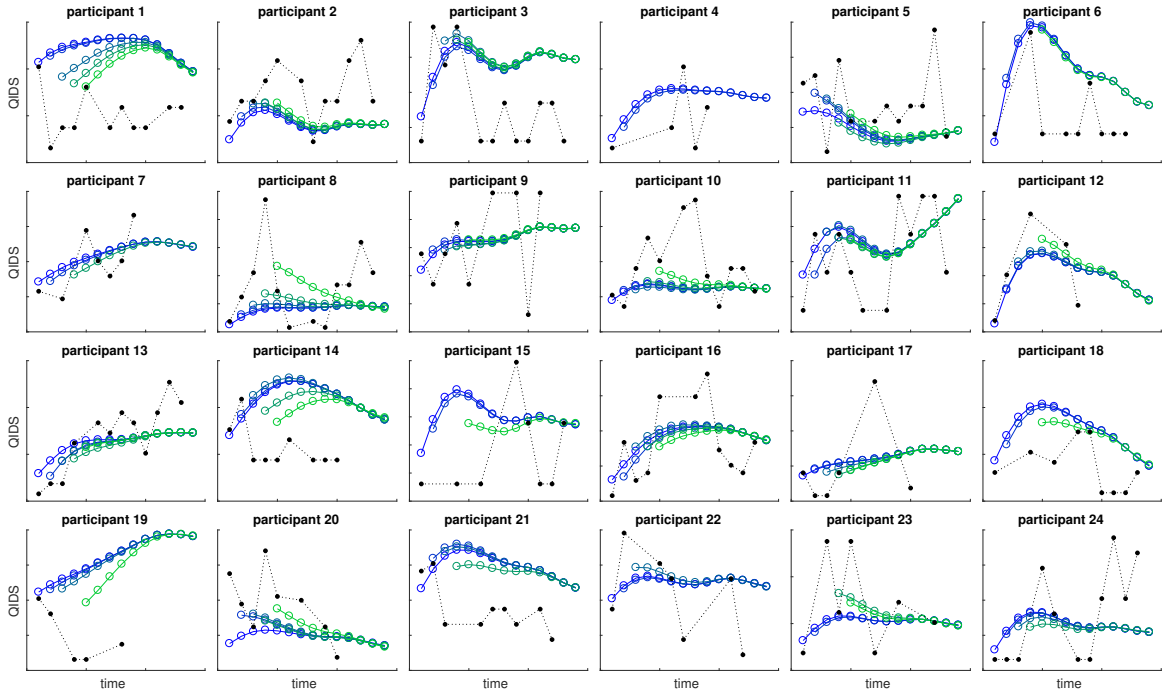


Figure 23: Some examples of predicted disease trajectories during follow-up. Each plot shows one participant, black dots are actual measured QIDS values, and circles are predictions. Trajectories that were obtained without conditioning are shown in blue, and after conditioning on the first 1,2,3 or 4 observations in green with stronger coloring indicating conditioning on more observations.

Table 7: Prediction of relapse. Results for the best model obtained by the prior shrinkage procedure, as selected by BIC. The scores marked with a star are the ones obtained by using the best possible classification threshold on the training data, according to the ROC curve.

	train	**test**	**validation**
rmse	0.781 ± 0.017	0.785 ± 0.047	0.790 ± 0.024
accuracy	58.57	57.77	55.38
accuracy*	70.37	69.01	71.28
sensitivity*	70.72	68.31	69.67
specificity*	70.02	69.71	72.90

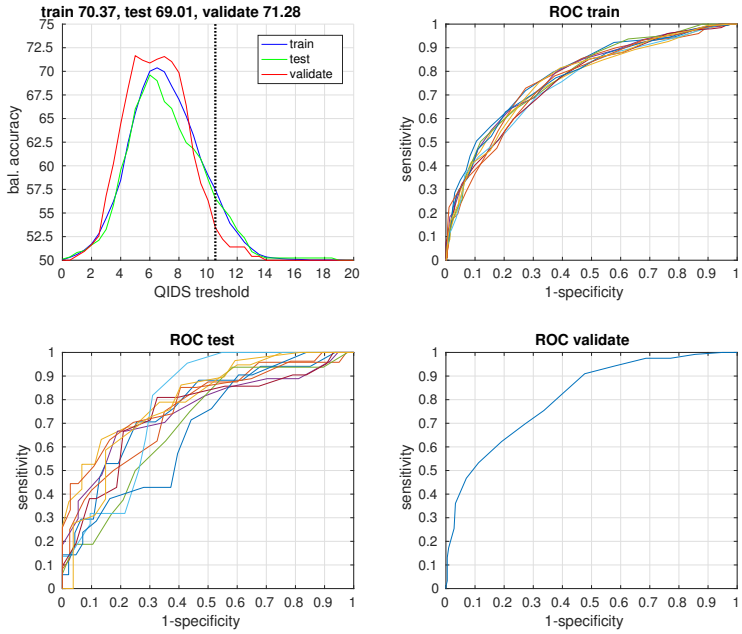


Figure 24: Top left: Relapse classification accuracy versus QIDS threshold ω_{pred}. The vertical dotted black line marks the clinical criterion for remission. The threshold optimizing training accuracy is a QIDS of 6; the same threshold also improves test and validation accuracy. The top right (lower left) plot shows ROC curves for the predictors obtained from the different CV folds on train (test) data. The lower right plot shows the validation ROC curve obtained through majority voting from the 10 different predictors. Area under the curve for validation data was 0.813.

4.5.4 Weight trajectory estimates

Across all folds, a total of 142 of 183 features were selected. The trajectories of 99 features deviated from zero by at least 0.05 (any($|\boldsymbol{\beta}_k|$) > 0.05),. Furthermore, there were 46 features for which any($|\boldsymbol{\beta}|$) > 0.1) and only 22 features for which any($|\boldsymbol{\beta}|$) > 0.15). These latter features are shown in Figure 25.

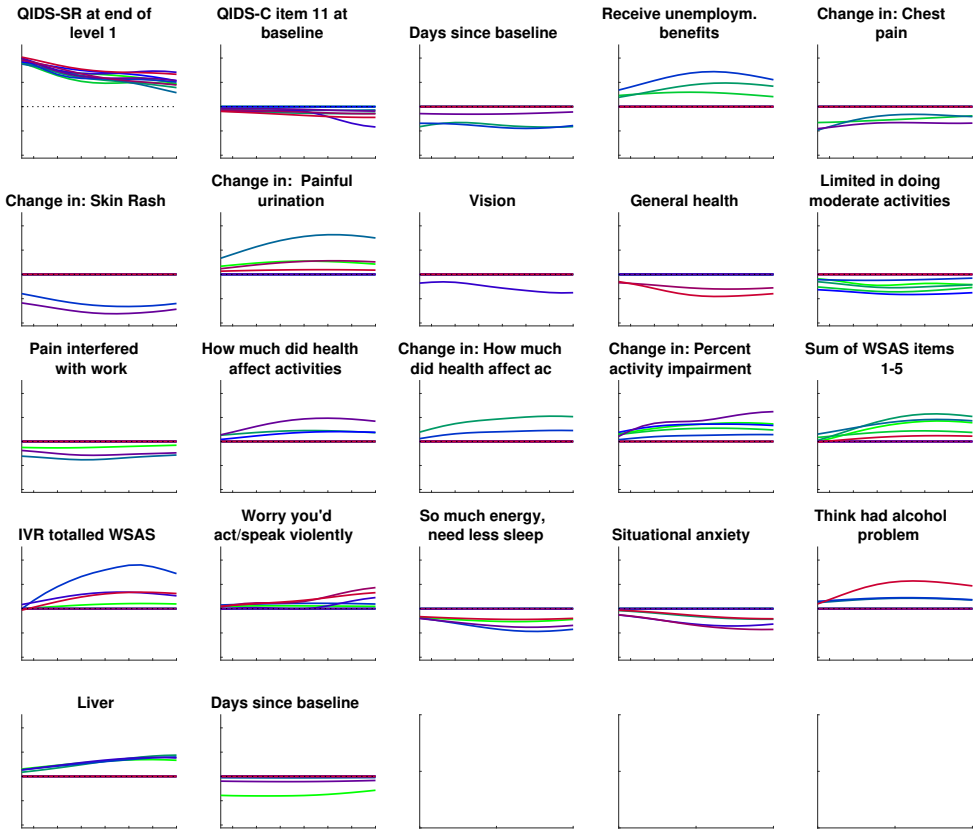


Figure 25: Posterior weight trajectories during follow-up for all features for which any($|w|$) > 0.15. Each line corresponds to one CV fold. Average number of selected features per fold was 36.5 ± 7.5.

As was the case for prediction of remission, the strongest predictor across all CV folds is the total QIDS-SR score (here at beginning of follow-up). While liver problems, total WSAS scores and more painful urination as compared to baseline specifically caused higher QIDS during late follow-up, alcohol problems were a more constant cause of higher scores. On the other hand, more chest pain and skin rash, general health problems, limitations in doing moderate activities, too much energy and situational anxiety strikingly contributed to relative *reductions* in total QIDS values during the whole follow-up period. 'Days since baseline' appears twice; this feature was collected twice for each participant - as part of the demographics questionnaire, and again as part of the cumulative illness rating questionnaire. As the two measures were unfortunately far from perfectly correlated ($\rho = 0.71$), we included both.

4.6 Relapse, conditioned on treatment period

Similar to online prediction, we can train the GPTP model on the combined observations during treatment and follow-up, and then employ matrix conditioning to predict trajectories during follow-up based on the observed trajectory during treatment. Doing so greatly improved results as summarized in table 8. A predictive RMSE of 0.695 indicated that $100 * (1 - 0.695^2) = 51.70$ percent of variance could be explained (without conditioning it was only 37.60 percent). The best balanced accuracy along the validation ROC curve shown in Figure 26 was 74.31 percent, with an AUC of 0.822.

A total of 205 out of 262 features were selected. Comparing to 142 features in the previous section, we notice that this approach is less parsimonious. 150 out of 205 features deviated by more than 0.05 from zero, 71 features by more than 0.10, and 26 features by more than 0.15. These are shown in Figure 27, and are quite similar to the corresponding weight trajectories from the previous section, with the main difference being that often trajectories are non-zero for more CV folds than before (compare for example features 'Change in Skin Rash', 'Receive unemployment benefits' or 'Vision').

Table 8: Prediction of relapse, conditioned on treatment phase. Results for the best model obtained by the prior shrinkage procedure, as selected by BIC. The scores marked with a star are the ones obtained by using the best possible classification threshold on the training data, according to the ROC curve.

	train	**test**	**validation**
rmse	0.713 ± 0.022	0.722 ± 0.034	0.695 ± 0.018
accuracy	58.74	57.40	57.45
accuracy*	71.33	71.98	74.31
sensitivity*	72.54	76.25	74.80
specificity*	70.12	68.71	73.83

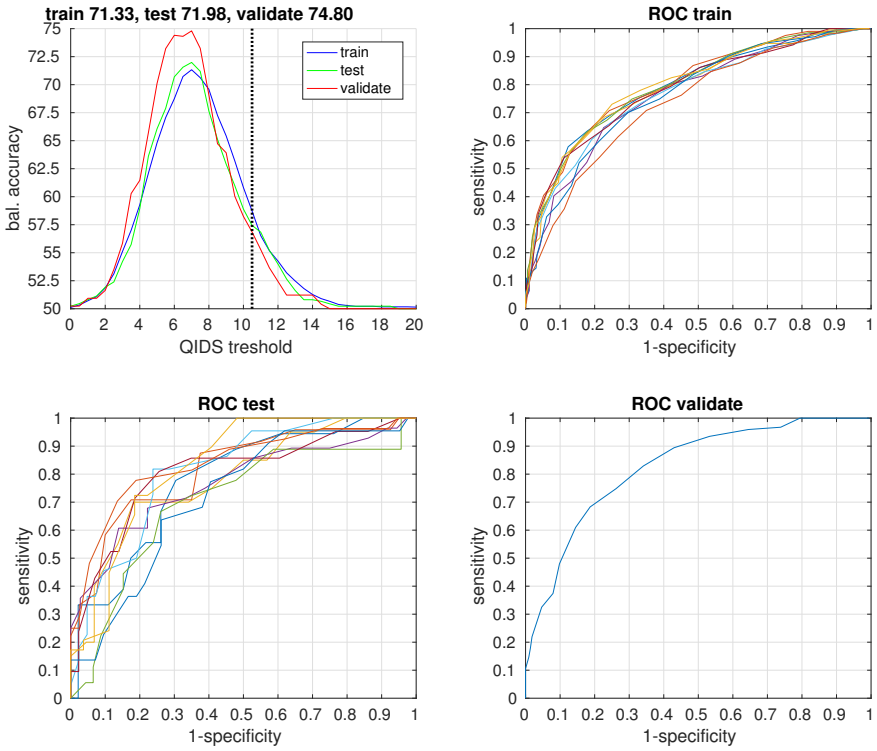


Figure 26: Top left: Relapse classification accuracy versus QIDS threshold ω_{pred}, when conditioning on data in the treatment phase. The vertical dotted black line marks the clinical criterion for realpse. The threshold optimizing training accuracy is a QIDS of 7; the same threshold also improves test and validation accuracy. The top right (lower left) plot shows ROC curves for the predictors obtained from the different CV folds on train (test) data. The lower right plot shows the validation ROC curve obtained through majority voting from the 10 different predictors. Area under the curve for validation data was 0.822.

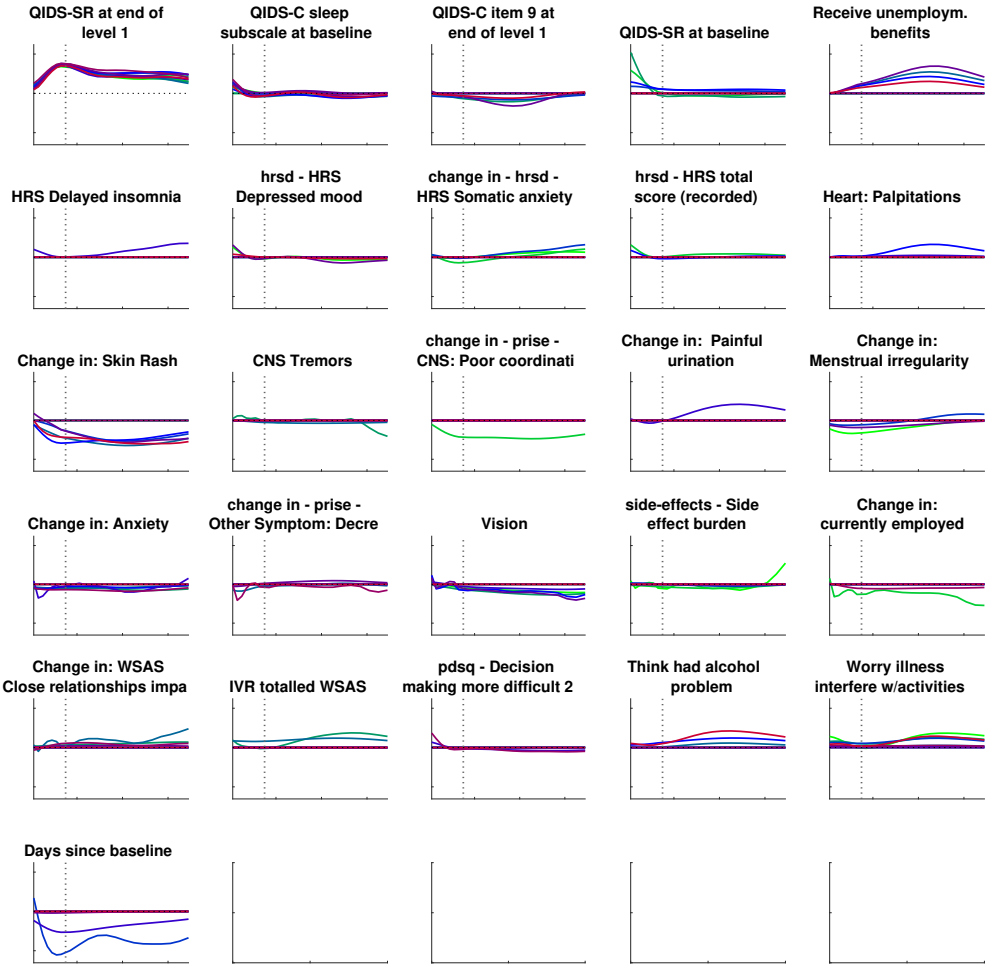


Figure 27: Posterior weight trajectories during treatment and follow-up for all features for which any($|w|$) > 0.15. The vertical dotted line marks end of the treatment period. Each line corresponds to one CV fold. Average number of selected features per fold was 51.8 ± 5.7.

4.7 Conclusion

We developed the method *Gaussian Process Trajectory Prediction* (GPTP) for prediction of longitudinal data, based on covariates that were collected at the beginning of the observation period, and applied it to a large data set of patients suffering from depression. GPTP captures the intuition that each covariate's influence changes with time, and that the observed overall trajectory results from a linear combination of these individual contributions. In Structural Equation Modeling terms, we can interpret our model as a structural model for covariate change (the GP prior), plus the most simple observation model (likelihood with Gaussian noise). GPTP deals in a principled way with missing data and censored data through marginalization rules for Gaussians.

This work represents a translation of a fully Bayesian version of Varying Coefficient Models from the geostatistical literature (Gelfand and Schliep, 2016; Bussas et al., 2015) for general applicability in machine learning. First, we extended the approach, which is usually not employed in the repeated measures setting, to longitudinal data, by assuming i.i.d. participants. Second, we have employed a much simpler cross-covariance kernel function than is usually the case, arguing that this would not adversely affect results. Indeed, simulations showed that even in cases where the kernel function is mis-specified, inference recovers true latent feature trajectories well. Third, our choice of kernel function enables the use of principled Bayesian feature selection. We generalized Automatic Relevance Determination to the multivariate setting with separable cross-covariances, which to our knowledge has not been done before. This, in turn, enables GPTP to be used for many more features than is usually the case in VC models. An additional advantage of the method is that it can be used online, such that predictions can be updated as new observations become available, without having to re-estimate the model.

One strength of the method is that regression weights can be directly interpreted as latent weight trajectories; this is in contrast to other ways in which covariates can be embedded in Gaussian Process Regression, whose application to longitudinal data has been covered recently very thoroughly by Karch (2016). However, these weight trajectories are constrained to be the same for all participants. This is quite a severe restriction. One possible extension could be to employ Dirichlet Process Mixtures to automatically find clusters of participants for which trajectories are the same (Hannah et al., 2011); however, any clustering that is not purely based on covariates precludes the possibility of prediction. Other ways of extending the model could be to (i) incorporate a non-zero mean function for the GP prior, which is a function of the covariates, (ii) let the kernel parameters become functions of the covariates or (iii) extend the likelihood to include additional terms for fixed or random effects.

Applying our method to a large dataset of patients suffering from major depressive disorder, we were able to explain 53.4 percent of the variance in observed QIDS scores during treatment, and 51.7 percent during follow-up (when conditioning on the treatment period). Thresholding the predictions, we were able to

predict remission at the end of the treatment period with 68.14 percent accuracy on validation data (with an AUC of 0.75), a strong improvement over the previously best approach by Chekroud et al. (2016) which achieved 64.50 percent accuracy. Prediction of relapse during follow-up has not been done on these data on the level of individual participants, to our knowledge. We were able to achieve an accuracy of 74.3 percent with an AUC of 0.82.

Although we aimed to find as sparse a model as possible, our results indicate that - at least with respect to our method - sparse models do not explain the data well, as 170 out of 211 features were selected during treatment and 205 out of 262 during follow-up. This speaks to the complexity of the disease that we investigated. However, it should be noted, that the available data consisted exclusively of questionnaires. Using brain imaging methods that record activity relevant to the disease, or mechanistic models that partly describe the nature of the disease, we might be able to extract relevant parameters that could be integrated as additional features into our approach, possibly increasing predictive power.

5 A computational approach to emotions

This section marks the beginning of the second part of this thesis. In the following, we present a computational approach to emotions in the context of decision-making. In order to understand why this is important with respect to computational psychiatry, we begin by describing depression as a disorder of emotion. Approaching emotions computationally means that we focus on the aspect of emotions as detectors of the relevance of an event. This requires introduction of the valuation of choices.

We also require the concept of metareasoning, i.e. how much one should "think about" a decision and its consequences before making the choice. Since metareasoning inadvertently happens within the processing limits of our brain, it considers optimal valuation in the face of resource constraints (Simon, 1956; Russell and Wefald, 1991). We assume the perspective that emotions result in fast approximations to the complex metareasoning problem, and develop detailed descriptions of this process with the goal to extract relevant parameters that could be used to increase the power of predicting disease outcomes or treatment success.

5.1 Introduction

We have already discussed the importance of improving our understanding of depression in section 4. One interesting way in which major depressive disorder can be characterized is as a disorder of emotion dysregulation and sustained negative affect (Angst et al., 2003; DeRubeis et al., 2008; Elliott et al., 2011). Before we go on to describe these two aspects and their interaction in detail, however, we need to define what we mean when we refer to the term 'emotion', as it is notoriously difficult to define, and no consensus exists on its scientific definition (Scherer, 2005).

From an evolutionary perspective, emotions have probably evolved from simple reflex mechanisms that enable animals to seek valuable resources and avoid harm (Scherer, 2005; LeDoux, 2012). Because the event that elicits emotion and its consequences must be somehow relevant to one's goals or concerns, emotions can be seen as relevance detectors (Frijda, 1987), which determine the relevance of an event by a complex, but nevertheless rapidly occurring evaluation process. This spreads over multiple levels of processing ranging from automatic and implicit to conscious, conceptual evaluations (van Reekum and Scherer, 1997). The subjective phenomena entering emotional awareness can be interpreted as introspective correlates of such valuation processes, which have direct, quantifiable consequences on behavior, providing a direct link between valuation, emotion and decision-making behavior (Rolls, 2005; Huys et al., 2015a). Here, we focus on this aspect of information processing, and the related valuation of choice options, which allows for a fresh perspective on the classic distinction between emotions and cognition(Damasio, 1997).

5.1.1 Depression as a disorder of emotion

As stated before, major depressive disorder can be characterized as a disorder of emotion dysregulation and sustained negative affect. Many contemporary models of depression explicitly formulate these two separate, but interacting cognitive components (Angst et al., 2003; DeRubeis et al., 2008; Roiser et al., 2012; Huys, 2007). Indeed, low mood, where all events are interpreted as indicating a negative immediate and future outcome, is one of the core symptoms of depression. Anhedonia is the other core symptom; it refers to a state in which things that were previously judged valuable are no longer found to be so. Because both symptoms refer to a dysfunction of valuation of outcomes, depression, at its heart, can be seen as a disorder of valuation. Consequently, it is useful to have a better understanding of how such valuations come about, in order to improve understanding of depression.

The first cognitive component is comprised of low-level affective processes (e.g. emotional reactions to affective stimuli). In MDD, these are abnormally sensitive to aversive events, and result in various behavioral changes such as increased sensitivity to punishments or attentional biases towards negatively valanced stimuli. For example, when presented with a happy and a sad face, depressed patients tend to focus more on the sad face as compared to healthy individuals (Gotlib and Joormann, 2010). More generally, aversive stimuli attract more attention than in healthy controls, which might reflect an expectation that such kind of information is more informative than it is (Barry et al., 2004). These processes are thought of as low-level, because they do not require processing of relevant stimuli in higher cognitive brain areas. As a result, low-level biases manifest themselves relatively quickly.

The second component consists of high-level cognitive emotion regulation processes. These are also known as appraisals and can be viewed as interpretations of the situation, depending on ones' current goals and past experiences (Smith et al., 1993), i.e they happen when something is considered to be somehow relevant to the organism. One example of an adaptive regulatory strategy is positive reappraisal - appraising the emotional situation in a new way - which is strongly associated with resilience in the face of stress (Southwick and Charney, 2012). It consists of changing the way a situation is construed in order to decrease its emotional impact. It acts relatively early in the process during which emotion is generated (this process can be roughly described as consisting of a phase of attentional deployment, followed by a phase of cognitive change - the appraisal - and then resulting in a response).

However, the lines can be slightly blurred, as the low-level biases in depression are relatively slow to emerge (Huys et al., 2015a), and might have high-level aspects. Patients who engage in maladaptive rumination, which is a pathological focus on how bad things are, strongly believe that it is important and helpful to engage in rumination, e.g. Nolen-Hoeksema et al. (2008); Treynor et al. (2003).

Another kind of (negative) appraisal, helplessness, comes later in the emotion-generative process, and is especially relevant in the context of depression. Help-

less appraisals can be of differing types. They can be stable in the sense that they persist over time, or unstable, that is, quickly changing in time. Global appraisals are likely to affect many areas of life, while local appraisals are more targeted at one specific kind of events. Finally, internal appraisals are ones where the event is thought to be caused by the appraising person themselves, whereas external appraisals are attributed to other factors. Stable, global and internal helpless appraisals predict future depressive episodes (Haeffel et al., 2008). One way in which helpless appraisals can manifest is through optimal inference under chronic stress, i.e. one correctly infers that one cannot change the situation for the better (Huys and Dayan, 2009).

Cognitive emotion control paradigms, in which subjects explicitly control their emotional response to affective stimuli, have conclusively shown that cognitive top-down control can modify low-level processes such as emotional consequences of affective stimulation, which in turn change the nature of the behavioral reaction (Gross, 2002). For example, instructing study participants to view a movie scene as a detached observer decreases the strength of their emotional reaction in comparison to participants that were instructed to vividly imagine the scene as if it were real. On the other hand, high-level processes can also be modified by low-level negative biases (Roiser et al., 2012). There is substantial literature on the interplay between these two processes, but explicit or mechanistic examinations of their interaction are rare; using standard paradigms, it is not possible to examine it in detail.

In this work, we focus on the valuation aspect of emotion as detailed above, and develop corresponding computational decision models. This might provide a lens through which the complex interplay between low-level biases and cognitive processes can be better understood, and quantitative statements of individual differences in such valuations may be obtained.

5.1.2 Valuation

From a theoretical perspective, valuation of action outcomes may or may not be based on a world model (i.e., an understanding of the structure of the world) that the agent has learned through experience. Model-based approaches assume such an explicit model of the world. Future consequences of actions, which may be inferred from the model, determine the value of stimuli. From these, optimal policies, which prescribe how to best act in each possible situation, can be inferred. This process of searching the model for an optimal action is flexible, but requires lots of processing power; it is intuitively close to what we understand as "thinking". Since model-based valuation takes place in the subjects' mind prior to making a decision which impacts the world outside the mind, the decision process corresponding to model-based valuation sits on top of the decision process which results in actions. We can say that the higher-level process is a metareasoning process (which we describe in section 5.1.3) resulting in the evaluation of actions which informs the choice mechanism itself. However, in any but the most simple situations, exact inference is computationally infeasible; instead, values

can for example be estimated by sampling possible future states.

In contrast, model-free approaches learn action (outcome) values directly by updating current value estimates through trial-and-error learning. In a given situation, the best action can directly be executed without having to execute a policy, because values for each action are available directly. These techniques are much faster than model-based methods; however, they require extensive experience in order for all action values to be learned accurately. These differing characteristics have been used to accumulate evidence that both systems are implemented in humans (Doya, 1999). The opposing strengths and weaknesses of model-based and model-free evaluation imply that interactions between the two systems might be advantageous, and they do in fact seem to exist (Huys et al., 2012; Pfeiffer and Foster, 2013). One interesting possibility is that the model-based system may generate experience through simulation, which is used to train the model-free system (Daw et al., 2011; Gershman et al., 2014). Since simulated actions only take place in the subjects' mind, we call the corresponding evaluations of action outcomes 'internal evaluations'.

Bayesian decision theory (BDT) couples the link between valuation and action quantitatively (Berger, 1985). It is possible to define heuristics or approximations within BDT, but it also provides a framework for organizing and providing a fresh perspective on the evidence for dysfunctional decision behavior in psychiatric disorders (Maia and Frank, 2011; Huys et al., 2015a). BDT specifies that actions should be chosen that are expected to lead to maximal long-run utility, where "utility" refers to the benefit of executing an action, and "long-run" means that not only immediate outcomes are considered, but also all future consequences. This is done from a Bayesian perspective, i.e., the *expected* long-run utility is computed, where the expectation averages over all uncertainties. For example, an agent might not have full knowledge about which objective state of the world it is in; the agent's subjective state represents everything that is known about the objective state. The uncertainty about the true objective state is evaluated by averaging over all possibilities according to their probabilities, which in turn combine current information with prior knowledge, for example acquired from previous experience. All possible choices and their outcomes can be summarized in a decision-tree; see Figure 28 for an example.

With regard to depression, aberrant valuations have a prominent model-based component. One example is helplessness, which can be applied to internal evaluations, and thus becomes part of the subjects' world model; this reduces the expected gains from engaging in model-based calculations (Huys et al., 2015a). From the perspective of BDT, helplessness can be understood as a form of uncontrollability (Huys and Dayan, 2009), which implies that subjects believe that there is little to no relationship between actions and their outcomes.

5.1.3 Metareasoning

Model-based values can not be computed exactly in all but the most simple circumstances, due to the limitations of computational power in the human brain.

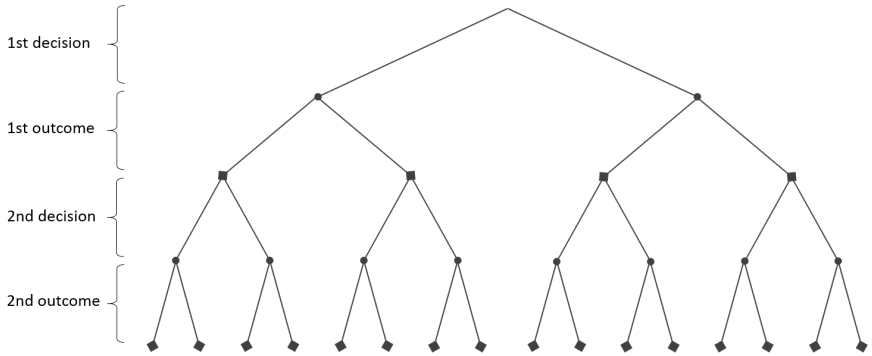


Figure 28: A decision-tree of depth 2. Each of the two possible actions in the first decision can lead to two different outcomes. These in turn enable 2 more actions each with another 2 possible outcomes. The number of leafs (nodes on the lowest level) grows exponentially with the length of the decision sequence.

The fact that the available resources constrain the amount of processing gives rise to the question of how to allocate the internal computational resources optimally in order to evaluate behavioral options. The metalevel decision problem is to choose what future action sequences to explore (Hay and Russell, 2011). This is a decision problem about which of the various options to evaluate internally. As such, it is a meta decision problem, because it sits on top of the original decision-problem. When an action and its outcome depend on previously selected actions, as for sequential decisions, metareasoning takes the form of a search in the underlying decision-tree (see Figure 29).

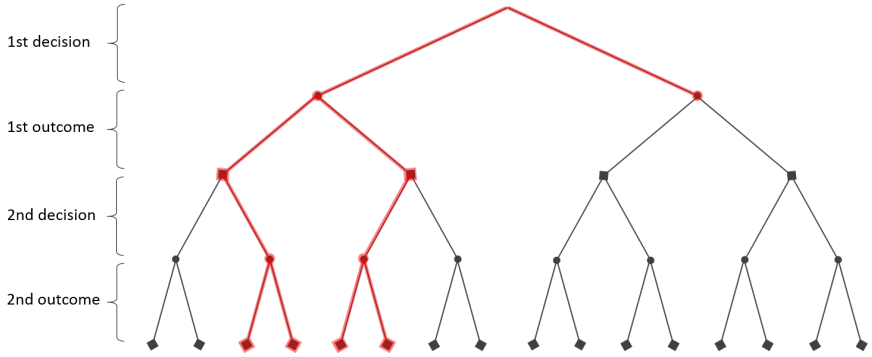


Figure 29: The red lines show a (meta-)search tree on top of the underlying decision-tree (black lines). The metareasoning problem is how to build up such a search tree. As such, it specifies a decision-problem on top of a decision-problem. The number of search trees grows with the number of leafs of the underlying decision-tree faster than the underlying decision-problem does, because any subtree is a valid solution candidate.

The difference between this and the original decision problem is that internally simulating poor actions is useful (Bubeck et al., 2009; Hay and Russell, 2011); the goal is not to avoid simulating suboptimal actions, but to maximize the expected improvement in decision quality. This induces a problem that is

even harder than the decision problem: The question is how to allocate available resources best or, equivalently, which choice to think about next. Even severely restricted versions of this problem have been shown to be $\mathcal{NP}$-hard (Conitzer and Sandholm, 2003).

The challenge in model-based reasoning is two-fold: the size of the problem, and the even harder task of finding solutions with limited resources. There are probably no relevant real-life situations where it can be solved exactly, so approximations are mandatory. One possibility is that emotions are effectively implementations of approximate metareasoning strategies, since they cause strong focus on particular behaviors, but also on internal evaluation of a narrow set of states (Huys and Renz, 2017). Thus, emotions function as approximate metareasoning strategies that prescribe how computational resources are allocated.

Given the importance of model-based valuation, and the compounding complexity of metareasoning, it is necessary to build models to examine how internal decisions about cognitive resource allocations are made. This may have beneficial effects for depression, but is a general problem of importance for cognitive and computational neuroscience.

Because it is so hard, approximations to the metareasoning problem need to be profound. That is, there likely exist some very simple metareasoning strategies. The problem can be divided into smaller subproblems, so subtasks with their own subgoals can be defined (Sutton et al., 1999). Efficient algorithms may be employed that quickly find good solutions, for example through sampling (Kearns et al., 1999). Another kind of approximate solution can be reached by never evaluating actions that yield bad immediate outcomes, independent of their future consequences (Knuth and Moore, 1975; Huys et al., 2012). This simplifies the problem considerably by effectively pruning away part of the search space. As pruning is our focus in this work, we describe it in detail in the next section.

5.1.4 Pruning

Some previous work has tried to start examining pruning (Huys et al., 2012, 2015b). In a sequential decision-making task, subjects were forced to approximate their internal evaluation of which actions to take, because the task was too complex to solve fully. Subjects were substantially impaired when the optimal action sequence involved a salient loss. In these cases, they frequently chose suboptimal sequences, even when that led to very substantial losses in comparison to the optimal sequence. This implies that they must have chosen to not evaluate the optimal action path during planning; they pruned away the corresponding part of the decision-tree. As this behavior did not seem to change with the size of the loss, the authors concluded that pruning is an inflexible, reflexive response that influences internal evaluations prior to decision-making.

Pruning might be relevant for the treatment of depression, as it was shown to correlate with sub-clinical scores of depression (Huys et al., 2012). The resulting undersampling of aversive outcomes biases the overall valuation positively (Dayan and Huys, 2008). In depression, the opposite may be the case.

However, the model proposed by Huys et al. (2012) is not a real process model, i.e. it does not actually describe how the pruning effect occurs, but only describes its strength on average. As such, it is a poor starting point to understand the underlying neurobiology. It is also poor in terms of measuring neural correlates because it is not very precise in terms of time. In section 6, we develop models that explicitly model the meta-reasoning process with the aim to better understand its components and effects, such as pruning.

5.2 Methods

5.2.1 Markov Decision Processes

Given a state s, an action a results in a transition to the next state s', yielding a certain reward. A sequence of n such state-action-reward-state episodes spans a decision-tree of depth n with the starting state as the tree root. Here we give a short introduction into Markov Decision Processes as a formal framework for valuation and decision-making. This provides the prerequisite to understanding metareasoning formally, and developing our own decision and reasoning models. As we have described above, the metareasoning problem arises as a decision-tree search in situations with several sequential decisions. This is why we have to investigate *sequential* decision-making as opposed to simple decision where only one choice has to be made. It can be formalized in terms of Markov Decision Processes (MDPs), which model time-discrete stochastic state-transition automata (Bellman, 1957). An MDP is defined by the 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \pi)$, where $\mathcal{S}$ is the set of states and $\mathcal{A}$ the set of actions. $\mathcal{R}$ is the reward function with $\mathcal{R}^a_{s,s'} = \mathbb{E}[R_{t+1}|S_{t+1} = s', S_t = s, A_t = a]$ being the expected reward received by changing from state s to state s' when executing action a at time t; capital R, S, A denote random variables. $\mathcal{T}$ is the state transition probability function with $\mathcal{T}^a_{s,s'} = P(S_{t+1} = s'|S_t = s, A_t = a)$.

Given a state s, an action a results in a transition to the next state s', yielding a certain reward. A sequence of n such state-action-reward-state episodes spans a decision-tree of depth n with the starting state as the tree root. In the case where the model is completely known, i.e., $\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}$ are known, the value $Q(a, s)$ of taking action a in state s can be written explicitly as

$$Q(a, s) = \sum_{s'} \mathcal{T}(s'|a, s) \left[\mathcal{R}(s, a, s') + V(s')\right]. \tag{72}$$

This formulation is known as the Bellmann equation (Bellman, 1961). Because the value of a state $V(s')$ is defined as

$$V(s) = \max_a Q(s, a). \tag{73}$$

This implies a recursive definition of the decision-tree.

The goal is to determine a policy $\pi(s)$ that maps each state to the action that maximizes total expected long-run utility (also referred to as 'return'),

$$a^* \leftarrow \operatorname*{argmax}_a Q(a, s).$$

The expected long-run utility itself is defined by the action-value function $Q(a, s)$, which may be rewritten as

$$Q(a, s) = \mathbb{E}\left[\sum_{t=0}^{\infty} r_t \Big| s, a\right], \tag{74}$$

where r_t is the reward obtained at time step t. Because the sum inside the expectation is not necessarily finite, it is often replaced by $\sum r_t \gamma^t$, with $0 < \gamma < 1$, so that γ can be interpreted as the discount factor for future rewards.

In an MDP, the current state of the environment is independent of the history given the previous state of the environment, $P(s_t|s_{t-1}) = P(s_t|s_{t-1}, ..., s_1)$, and thus $\mathcal{R}$ and $\mathcal{T}$ are time-invariant. This assumption is often summarized as "the future is independent of the past given the present".

An MDP as defined above describes a fully observable environment. Almost all sequential decision-making problems can be formalized in such a way, because problems where the agent only indirectly observes the environment (partially observable Markov decision processes), can be reduced to the MDP case, although this typically results in an explosion of the size of the state space (Sutton and Barto, 1998).

5.2.2 Meta Decision Processes

As stated earlier, the metalevel decision problem is to choose what future action sequences to explore (Hay and Russell, 2011). In this section, we define the problem formally as well as provide an overview of previous research.

Let $\mathcal{M} = (U_1, ..., U_K, \mathcal{E})$ be a metalevel probability model, where U_k is the utility of performing action k and $\mathcal{E} = \{E_1, ..., E_J\}$ is a set of internal evaluations. We can define a policy as $\pi : \mathcal{S} \to \mathcal{E} \cup \{\perp\}$, where $\perp$ indicates the operation that stops the metareasoning process (at which point an object-level decision is selected). Then the expected utility of a meta-policy π is

$$V^{\pi}_{\mathcal{M}}(s) = \mathbb{E}_{\pi}\left[-cN^{\pi}_{s} + \max_{k} \mathbb{E}\left[U_k|s\right] | S_0 = s\right], \tag{75}$$

where s is the state of current knowledge, c is the (constant) cost of each internal evaluation, and N^{π}_{s} is the number of internal evaluations performed if the policy is started in state s. We use the subscript in $V^{\pi}_{\mathcal{M}}$ to differentiate this metalevel value function from the choice-level value function V in the previous section. With this definition in place, we can state the metalevel decision problem formally:

Metalevel decision problem: Given $\mathcal{M}, s$, find $\text{argmax}_{\pi} V^{\pi}_{\mathcal{M}}(s)$.

The metalevel probability model can be shown to be equivalent to a metalevel MDP specified by the 5-tuple $(\mathcal{S} \cup \{\perp\}, \mathcal{E} \cup \{\perp\}, \mathcal{R}, \mathcal{T}, \pi)$ with $\mathcal{R}(s, E, s') = -c$ for $s, s' \in \mathcal{S}$ and $E \in \mathcal{E}$, and $\mathcal{R}(s, \perp, \perp) = \max_k \mathbb{E}\left[U_k|s\right]$. The expected utility of a policy for such an MDP is $V^{\pi}_{\mathcal{M}}(s) = \mathbb{E}_{\pi}\left[\sum_t r_t\right]$ and does not directly depend on c, because the cost of internal evaluation is factored into the reward. Both

formulations are valid specifications of metalevel decision processes (Hay and Russell, 2011).

Bandits (Berry and Fristedt, 1985) are one context in which metalevel decisions have been studied. In a bandit problem, there are K available actions. At each time step, one of these may be chosen, and the reward is drawn from a fixed distribution that only depends on the chosen action (so the state space contains only one state). For example, consider the question how K different treatments should be allocated to patients in order to maximize the number of successful treatments. The trade-off between exploring which treatments are best and choosing the best known available treatment for a particular patient at any given time is known as the exploration-exploitation dilemma (Sutton and Barto, 1998). A solution in bandit problems is the upper confidence bounds (UCB) algorithm (Auer et al., 2002) which selects the action that maximizes a sum of a term that favors actions that have performed well before, and a term that favors actions that have been executed fewer times. As a result, bandits are biased towards higher means and higher uncertainty (Yu, 2011).

In sequential decision problems, where an action and its outcome may depend on previously selected actions, metareasoning takes the form of a search in the corresponding underlying decision-tree. In a Monte Carlo simulator, the generative model of the MDP can be used to perform stochastic simulations of the consequences of actions. This corresponds to making a noisy measurement of the expected utilities of actions through sampling. Each generated simulation sequence leads from a leaf of the current search tree to a terminal state. Monte Carlo tree search (MCTS) provides a framework for implementing an objective that specifies how to use Monte Carlo simulations to decide which sequence to simulate next, and which action to take based on the resulting tree. This problem - how resources that are available for internal evaluation during planning must be allocated - has been investigated from a number of different perspectives (Anderson and Oates, 2007; Hay and Russell, 2011; Griffiths et al., 2015; Moya et al., 2016). Yet, how exactly the trade-off between costs of internal evaluations (the passage of time) and benefits in term of improvements in decision quality can be implemented in general remains an open question.

Usually, the UCB algorithm is employed in order to choose which simulation to make next, and the resulting MCTS algorithm is known as UCT (Upper Confidence Bound 1 applied to trees; see Kocsis and Szepesvári (2006)). This algorithm picks the simulation which maximizes a weighted sum of how well that simulation has done previously (corresponding to exploitation) and a term that scales inversely with the number of times this simulation has occured before (corresponding to exploration).

However, despite having a number of nice properties (such as that the probability of simulating a suboptimal action converges to zero polynomially), the UCT algorithm does not solve the metareasoning problem. First, simulations cost more for actions that seem worse, but in metareasoning the value of an action should not impact on its simulation cost. This has the effect that simulations are biased away from actions which currently have a low estimated long-run

utility. Instead, a metareasoning process really should just do exploration, for example by choosing the action with the highest uncertainty. Second, UCT does not implement a trade-off between expected gains and costs of internal evaluations, because it does not specify a rule for stopping simulations.

However, it is in general not easy to find alternatives to or improvements over UCT. One simple example are myopic policies, where each internal evaluation is selected based on the assumption that at most one more internal evaluation remains (this is also called the metagreediness assumption; see Russell and Wefald (1991)). Such a policy will stop computing and select an action if the expected improvement in utility of computing the next step falls below the cost of computation c:

$$\sum_{s'} \mathcal{T}^{E}_{s,s'} \max_j \mathbb{E}\left[U_j|s'\right] - \max_j \mathbb{E}\left[U_j|s\right] < c \quad \forall E. \tag{76}$$

Here, $\mathcal{T}^{E}_{s,s'}$ denotes the state transition probability function and U_j is the utility of performing evaluation E_j.

Some interesting properties can be proven, such as that if the optimal policy would stop in a certain state, the myopic policy is guaranteed to stop as well (Hay and Russell, 2011). However, the myopic restriction means that a corresponding policy will stop too early in cases when changing the currently preferred action to the optimal action takes more than one internal evaluation.

Fundamentally, a myopic strategy implements a greedy, depth-one search at the metalevel. Nonetheless, in solving the problem of optimal metareasoning, *sequences* must be considered because in some cases the value of a internal evaluation may not be apparent as an improvement in decision quality until further internal evaluations have been done. As an alternative to providing a fixed (depth-one) metalevel policy, one could learn one using the technique of Reinforcement Learning, as has been suggested in Russell (1997). When doing so, the credit assignment problem arises: rewards that were obtained through actions need to somehow be assigned to the (potentially very many) internal evaluations that were done preceding the action. It is currently not clear how to do this in general. Additional information about the internal evaluations would naturally greatly improve the ability to learn an explicit meta-reasoning model from data. Indeed, in section 6 we explore how to learn a model on the meta-level from choice data, as well as incorporate additional data that more directly informs such a model, with the hope to extract parameters that describe individual differences, for example with respect to pruning behavior.

5.2.3 Pruning mean-field models

Because we build on work by Huys et al. (2012), we describe here formally their approach, and introduce the basic concepts and models that are needed to understand it. Huys et al. (2012) compared several sequential choice models,

where the action-value function as defined in equation 72 takes the form

$$Q(\boldsymbol{a}|s,d) = \sum_{i=1}^{d} \gamma_{r_i}^{i-1} \beta_{r_i} r_i. \tag{77}$$

The action/choice sequence $\boldsymbol{a}$ is of length d, s is the starting state, $r_i = r(a_i)$ is the reward obtained by the i-th action in the sequence and β_{r_i} are the 4 reward sensitivity parameters corresponding to the 4 reward sizes that were available in the corresponding decision-making task (big loss, small loss, small win, big win). Finally, γ_{r_i} are the parameters which discount future rewards. The model contains two such parameters, one corresponding to the lowest possible reward (=big loss), and another parameter for all other reward sizes. The vector of actions contains numbers that code for the specific actions. The action-values are converted into probabilities by passing them through the softmax function

$$p(\boldsymbol{a}|s,d) = \frac{e^{Q(\boldsymbol{a}|s,d)}}{\sum_{\boldsymbol{a}' \in \mathcal{A}_{s,d}} e^{Q(\boldsymbol{a}'|s,d)}}, \tag{78}$$

where $\mathcal{A}_{s,d}$ is the set of possible action sequences of length d for starting state s. This assigns higher probabilities to sequences with higher Q-values, such that differences between Q-values are converted into much bigger differences in probability - a disproportionately big part of the probability mass is consequently assigned to the highest Q-value.

Huys et al. (2012) argue that the discount parameters can be interpreted as an approximation of parameters that control how likely an explicit forward decision-tree search is to stop at a certain node when encountering r_i. The discount parameter corresponding to the biggest losses can consequently be interpreted as indicating how much individuals prune. This trick enabled the authors to model an aspect of forward tree search, and hence the metareasoning process, without explicitly having to compute the search itself. Here, we extend their model by discarding the discount parameters, and instead taking the tree search explicitly into account. It is not immediately clear how to do this, as it requires modeling a separate meta-reasoning process somehow on top of the model defined in equation 77.

One way is to replace the discount parameters by weights $\pi(a_i)$ representing how likely on average each particular action a_i has been thought about given a particular tree-search strategy. In section 6.4 we formulate several specific hypotheses as tree-search models and evaluate their ability to explain the data.

5.2.4 Internal evaluations are decision-tree sequences

We formalize the concept of a sequence of internal evaluations ('thought sequence') as a sequence of decision-trees with each evaluation adding a branch to the tree. Such a sequence can be concisely expressed as one single search tree with edge labels that specify the ordering in which evaluations have happened. An example is shown in Figure 30, where the left part shows the layout of the

pruning paradigm that we introduce in section 6; the 6 states are shown on screen as rectangles, with the possibility to move between them. The subject might first consider the transition from state 1 to 4, then the transition from state 4 to 2 etc. Each additional evaluation of a transition effectively grows the corresponding search tree by adding a branch carrying a label denoting the rank of the transition (rank 1 refers to the first evaluation, rank 2 refers to the second evaluation and so on).

A valid internal evaluation is then nothing but an expansion of the search tree at any node which has at least one edge that has not been expanded so far. The metareasoning process now involves an internal decision between branches of the tree that can be added. More generally, the process defines a transition matrix between search trees (=partially evaluated decision-trees). We can formally define this process by writing the probability of the l-th evaluation / branch being added as the probability of moving from search tree t_{l-1} to t_l. A very simple approach, for instance, would state that individuals tend to consider branches with higher immediate reward according to the softmax,

$$p(t_l|t_{l-1}, \beta_{\mathcal{M}}) = \frac{e^{\beta_{\mathcal{M}} r(t_l|t_{l-1})}}{\sum_{t'_l \in \mathcal{T}_l} e^{\beta_{\mathcal{M}} r(t'_l|t_{l-1})}}, \tag{79}$$

where $\beta_{\mathcal{M}}$ is the reward-sensitivity parameter of the metareasoning model $\mathcal{M}$, $r(t_l|t_{l-1})$ the meta-reward obtained by the l-th internal evaluation and $\mathcal{T}_l$ the set of possible search trees. At first glance, this runs counter to our aims of minimizing uncertainty about outcomes during metareasoning. However, when the task does not include any stochasticity in the first place, as is the case in the paradigm that we discuss in section 6, choosing evaluations based on their immediate rewards seems to us like a sensible choice.

With respect to our main question - how exactly participants perform metareasoning, especially with respect to pruning - different models can be compared, in order to pick the best one. Differences between models can arise by the particular search strategy that specifies which nodes can potentially be considered for the next evaluations (e.g., depth-first vs width-first search); other possibilities include picking any particular stopping criterion, or changing how to pick the next choice among the possible candidates defined by the search strategy (i.e., changing equation 79). We discuss and compare the corresponding models in the next section.

6 Pruning paradigm

In order to test specific statements about the metareasoning process, we implemented a behavioral paradigm which is a variant of the pruning paradigm by Huys et al. (2012), which we already described in section 5.1.4. Pruning might be relevant for the treatment of depression, as it was shown to correlate with subclinical scores of depression (Huys et al., 2012). The resulting undersampling of aversive outcomes biases the overall valuation positively (Dayan and Huys, 2008).

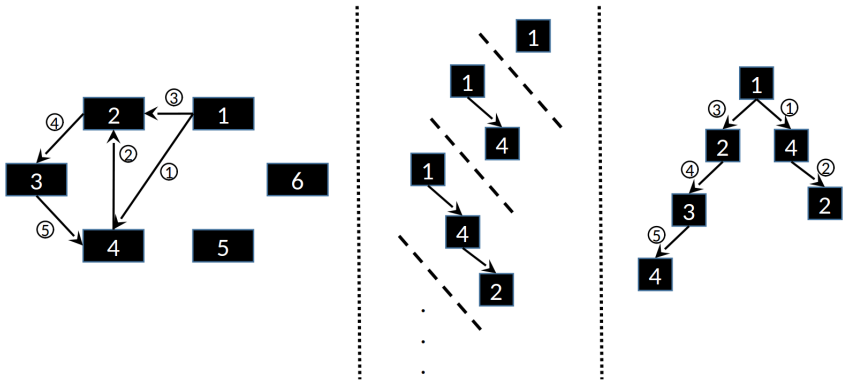


Figure 30: Modeling the internal evaluation process as a sequence of decision-trees. Left: Pruning game states are numbered from 1 to 6; circled numbers indicate the sequence of evaluations. The example shows one sequence of internal evaluations. Middle: The corresponding sequence of decision-trees, where the numbers in the nodes correspond to respective game states. Right: A more concise notation for the whole sequence as one edge-labeled decision-tree, which we call 'search tree'. We speak of the node at the top of being at the highest level. Thus, children nodes are always one level lower than their parents.

In depression, the opposite may be the case. However, the model proposed by Huys et al. (2012) is not a real process model, i.e. it does not actually describe how the pruning effect occurs, or when exactly, but only describes its strength on average. We aimed to build an explicit model of the metareasoning processm, which would capture the effect much more precisely, and hence could also be more sensitive to individual differences.

In the pruning paradigm, participants had to plan paths through mazes to maximize the total rewards and minimize the total losses earned along these paths. In the paradigm, subjects were forced to approximate their internal evaluation of which actions to take through pruning away bad parts of the decision-tree, because the task was too complex to solve fully. Because binary decisions provide only very limited information (2^n bits for a sequence of length n) about the metareasoning process, we collected additional data: Individuals' eye movements were recorded with the underlying assumption being that participants will tend to look at the areas of the computer screen that they are currently evaluating. That is, we assumed that gaze position should be approximately informative about the internal evaluations.

In the following sections, we first describe the paradigm in detail. Then, we develop a set of computational models that constrain the search space such that it becomes possible to infer the metareasoning process from behavioral data alone. We show that there is not enough information in the sequential decisions alone to infer the metareasoning process, so we develop a second set of models, which relate eyetracking data to sequences of internal evaluations, and thus allow for a more detailed inference procedure. These models directly address the

metareasoning problem of how computational cost is traded for improvement in decision quality. Inference with these models is tricky, due to their complexity - we attempt both across-trial inference, as well as trial-based inference, and finally compare the results.

6.1 Experimental description

The environment ("maze") consisted of 6 states distributed across the computer screen. In each state, two choices were available, each choice resulting in a transition to another state. However, the available choices / transitions and their corresponding rewards were not visible on screen; instead, they were learned in a training phase (see Figure 31).

After training, participants were presented with a number of trials, each of which consisted of 4 phases, with 1 second intervals of blank screen in between. In the preparation phase, participants saw a message about the length of the next choice sequence in the center of the screen for a duration of 3 seconds. Immediately afterwards, the planning phase began: the starting state was indicated (by coloring the corresponding game state on screen in green), and participants were given a maximum period of 9 seconds to plan their sequence of decisions, which they could abbreviate at any given moment by starting to input their decisions through pushing the corresponding buttons on the response box. A message box with the countdown was shown in the center of the screen. As soon as the 9 second period was over, or participants had pushed one button on the response box, the input phase began, which lasted 800 ms plus an additional 300 ms for each choice to be made. The input phase was designed to be just long enough so that participants could manage to input their planned choice sequence without being able to think more about it during input. If they did not input enough choices, they received a penalty of -200 points. Finally, participants were given visual feedback about their choices in form of arrows pointing along the corresponding game state transitions on screen; each arrow was displayed for 500 ms. See Figure 32 for a depiction of the trial design.

Each participant was presented with 144 trials, each of which consisted of a sequential decision-making task involving sequences of 3-6 binary decisions under time constraints. The trials were split into three blocks, with small breaks of maximum 5 minutes in between. The four different sequence lengths combined with 6 different possible starting states to 24 unique trials in total, so that participants encountered each of these trial types a total of 6 times. For input, response boxes with two active buttons were used.

Before being presented with the trials, participants were trained to ensure that they understood the paradigm sufficiently. The first half of the training phase consisted of learning the transitions between states. After one minute of being able to explore the maze freely, where the current state was indicated by color, tasks of differing difficulty were presented, asking participants to get from one state to another in exactly 1, 2 or 3 choices. This part of the training was repeated until participants' error rate on 1-choice tasks dropped below 10 percent.

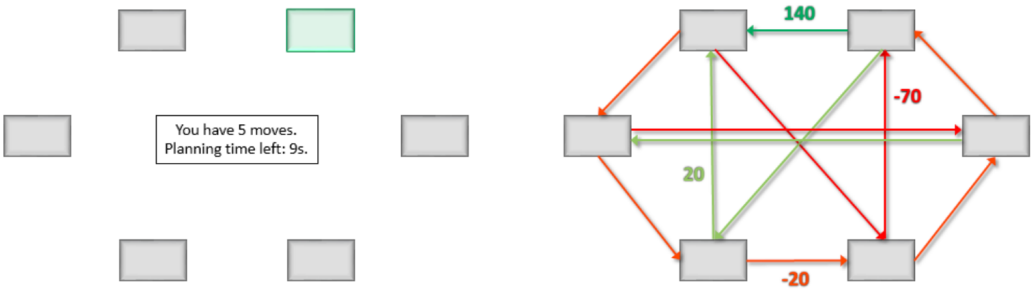


Figure 31: Left: In each trial, six boxes were displayed on the screen corresponding to the 6 game states. The color green marked the starting state. During the planning phase, the length of the choice sequence as well as the planning time left was displayed in the center of the screen. Right: The underlying transition structure was never shown, but participants were trained until they had learned it (see text). The transition from the top right to the top left state yielded 140 points, all light green transitions 20 points, while the light red transitions (which run counter-clockwise around the circle) incurred a loss of 20 points and the dark red transitions incurred a loss of 70 points.

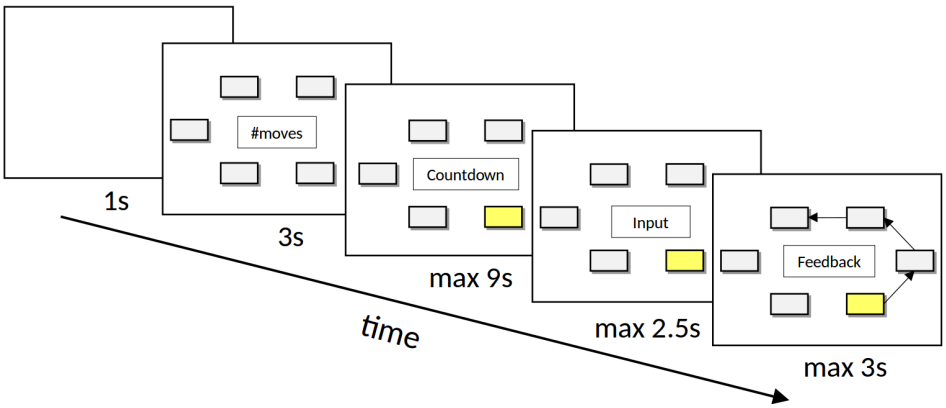


Figure 32: Pruning trial design: Each trial started with a blank screen that was shown for 1 second, after which the number of choices for the current trial was shown for 3 seconds. Third, the starting state was marked and participants had a maximum of 9 seconds to plan their choices. Finally, after the input phase, a visual feedback was given to the participants with arrows indicating the choices they made; the rewards were displayed next to the corresponding arrows.

The second half of the training consisted of learning the rewards. Participants could again explore the maze freely for one minute, where this time the reward was displayed after each choice. After exploration, 5 tasks were presented that required the participants to maximize the reward obtained from a sequence of 2 or 3 choices, given a certain starting state. Next, 5 more tasks were presented that required the participants to make their choices under the same time constraints and in exactly the same way as was required in the trials post-training. In the end of this training phase, the participants were asked to name the reward associated with each of the transitions. This phase, too, was repeated until error rate dropped below 10 percent.

6.2 Recruitment

We recruited a total of 22 participants through the "University Registration Center for Study Participants" in Zürich. Exclusion criteria were age < 18 or age > 50, any history of psychiatric illness (including drug addiction), any history of neurological or endocrinological illness, any history of brain injury, drug consumption within the last 6 months, and alcohol consumption within 24 hours before the experiment. One of the recruited participants turned out to have a neurological implant; another participant was unable to finish the training successfully. Three further participants scored worse than chance throughout the whole task. We concluded that they could not have understood the rules sufficiently, such that all analyses reported here refer to the remaining 17 participants.

6.3 Preprocessing

Eyetracking data was acquired with an EyeLink II from SR Research, Ottawa, Canada. We extracted fixation data as provided by the manufacturer's preprocessing software. Despite careful calibration, a histogram over all fixation data revealed that the data was not always properly centered. We show one example in Figure 33. Thus, we aggregated fixation data from the preparation phase (where only a message box was displayed in the center of the screen) independently for each of the three blocks of trials for each of the participants, and corrected all fixation data from that block by the difference between the center of the screen and mean fixations during preparation. Visual inspection of the histograms suggested that this led to good calibration.

Below, we are interested in choices that are indicative of internal evaluation processes. At times, subjects made frank and obvious errors that resulted from translating an intended path to a sequence of left-right button choices. Examples are going back and forth from between the leftmost and rightmost states; taking a large loss transition at the very end of a sequence; or failing to take the large reward when possible. As we reasoned that these error trials would not be indicative of the underlying valuation process we removed them from consideration.

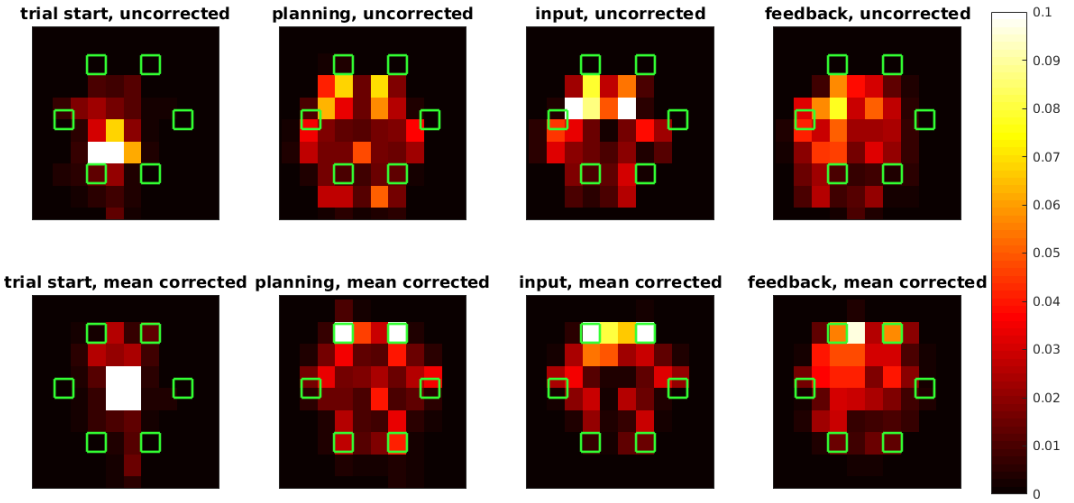


Figure 33: Top row: the fixation histogram across all trials for one participant. The green rectangles represent the game states. Brighter colors indicate more frequently gazed at screen positions. The data seemed to be shifted towards the bottom left corner in each of the four trial phases. The bottom row shows the same data after mean correction. As expected, the most frequently gazed at screen positions are the two game states at the top, the transition between which yielded the big reward. It is worth pointing out that indeed participants focused on the states; alternatively, it is not implausible that they could have focused on the invisible transitions between states. So any model that incorporates the eye data should do so by using it as information about states rather than actions. During feedback arrows were displayed representing the chosen actions; gazes were much more diffuse in general, but focused more on actions rather than states.

6.4 Behavioral analysis

We first investigated how much information could be extracted from behavioral choice data alone. We did this by asking to what extent parametrically defined evaluation strategies formulated as generative models could be distinguished. Specifically, we formulated a variety of models that specify how individuals might choose to evaluate parts of the decision-tree they are faced with, and asked whether these models led to more or less parsimonious fits of the behavioral data. As the fits are computationally challenging, we started by examining fixed-effects approaches in the sense that they collapsed across individual trials. As such, here we only make claims about metareasoning processes for individuals overall, but not for individual trials.

6.4.1 Computational models

As discussed previously, the standard (mean-field) model by Huys et al. (2012) that we presented in section 5.2.3 does not include an explicit model of the metareasoning process, because of the immense complexity. Here, instead, we attempt to build behavioral models that implement a full model $\mathcal{M}$ of the sequential order of evaluations during the planning process. Several specific hypotheses about the metareasoning process that are formulated as such process models can be compared and evaluated by their ability to explain the data.

A first approach to testing such explicit models of internal evaluation strategies is to replace the parameter γ in equation 77 by a weight $\pi(a_i)$ representing the probability with which each particular choice a_i would have been considered under a particular tree-search strategy, such as greedy depth-first search. The motivation for precomputing the weights as averages given a parametrized description of the nature of the metareasoning process (instead of modeling one explicit process per trial) is that this decouples modeling of the metareasoning process from each individual trial. However, we note here that the number of sequences of internal evaluations that have to be evaluated to compute $\pi(a_i)$ is usually very substantial because *all* possible sequences have to be considered. Therefore some processes, such as any kind of width-first search, had to be excluded for computational reasons. We do not consider this as a strong limitation, as a width-first strategy also seems inherently unlikely as it would require extensive 'jumping around' between distant nodes in the search tree during the process of evaluation, and this appears a priori biologically very unrealistic.

A complete specification of the metareasoning process must include a stopping criterion that specifies the probability to stop thinking and start acting after each thought. This is because the key question from a metareasoning perspective is whether costs of further computation are outweighed by the anticipated gains. For reasons of tractability, we started by making the stopping criterion only a function of time, so it is independent of which actual internal evaluations were carried out. As such, it depends on the cost of computation, but neglects the contribution of anticipated gains. We should note that because the paradigm does not include stochasticity, minimizing uncertainty of outcomes, which is the

main goal of the metareasoning process as stated in section 5.2.2, is not our aim here.

6.4.1.1 Formal definition Let the value Q of an action sequence depend on the metareasoning model $\mathcal{M}$ as follows:

$$Q(\boldsymbol{a}|s,d) = \sum_{i=1}^{d} \pi_{\mathcal{M}}(a_i)\beta_{r_i} r_i, \tag{80}$$

where the metareasoning model-derived 'expected probability' of having thought about a certain action, $\pi_{\mathcal{M}}(a_i)$, substitutes for $\gamma_{r_i}^{i-1}$ in the mean-field model (equation 77). The expected probability π is computed with respect to which of the search trees contain action a_i, where the trees are weighed by the probability of encountering them according to the model. Let

$$\pi_{\mathcal{M}}(a_i) = \sum_{l=1}^{L} \sum_{t_l \in \mathcal{T}_l} p_{\mathcal{M}}(t_l) I(\boldsymbol{a}_{1:i} \in t_l), \tag{81}$$

where $\mathcal{T}_l$ are all internal evaluation sequences of length l, $p_{\mathcal{M}}(t_l)$ is the probability of t_l under model $\mathcal{M}$, L is the maximum number of evaluations that are allowed, and $I(\boldsymbol{a}_{1:i} \in t_l)$ is equal to 1 if the sequence of choices $[a_1, ..., a_i]$ is wholly contained in t_l, and 0 otherwise.

Next, we specify the probability of t_l under the metareasoning model $\mathcal{M}$. As mentioned above, we work with a stopping process that just captures some cognitive resource independent of the evaluation results. Let

$$\begin{aligned} p_{\mathcal{M}}(t_l) &= q(l) p^*_{\mathcal{M}}(t_l) \\ p^*_{\mathcal{M}}(t_l) &= (1 - q(l-1)) \sum_{t_{l-1} \in \mathcal{T}_{l-1}} p^*_{\mathcal{M}}(t_{l-1}) p_{\mathcal{M}}(t_l | t_{l-1}, \beta_{\mathcal{M}}), \end{aligned} \tag{82}$$

where the term $q(l)$ specifies the probability of stopping to evaluate after l evaluations, and $p^*_{\mathcal{M}}(t_l)$ denotes the probability of having thought about a sequence starting with the subsequence given by t_l. Conversely, $(1 - q(l - 1))$ describes the probability of continuing to evaluate beyond length l, and $p_{\mathcal{M}}(t_l|t_{l-1}, \beta_{\mathcal{M}})$ is the transition matrix that captures how a particular metareasoning strategy determines the transition between partially evaluated subtrees. It is a softmax with parameter $\beta_{\mathcal{M}}$ across the values of all possible next evaluations from tree t_{l-1} according to the current search strategy and state of the tree. Finally, we set $p_{\mathcal{M}}(t_0) = 1$, where t_0 is the evaluation sequence containing just the starting state, i.e., we assume that participants always understood the task and start their evaluation at the starting state.

The sequence in which choices / tree nodes are considered, as well as the probability of stopping $q(l)$ are defined as part of the individual metareasoning strategies that we describe below.

6.4.1.2 Metareasoning strategies We considered several different mathematical forms that we describe in the following, each corresponding to a metareasoning strategy, i.e. some recipe that prescribes in which sequence nodes of a search tree should be evaluated. These were implemented as computational models such that each individual model was fitted to the whole dataset of all individuals; as a consequence, claims about which model describes the data best can only be made at the level of the whole set of individuals. We compared two node selection strategies (depth-first and beam search) and three stopping criteria. Further, we investigated if models explained the data better when we added some kind of Pavlovian attraction bonus, that scales with the distance to the big reward, or if we add some effect of learning either on the choice level or on the meta level. We tested all combinations of the ingredients above, and so we tested a total of $2 \cdot 3 \cdot 2 \cdot 2 = 24$ metareasoning models, and compared these to a model that did not model metareasoning at all ($\pi_{\mathcal{M}}(a_i) = 1 \; \forall a_i$). The details are described below.

First, we compared two different ways that prescribe the set of tree nodes that can potentially be evaluated at each metareasoning step: Depth-first search always seeks to expand the edge with the highest value on the lowest level. Beam search, on the contrary, expands the edge with the highest value at each step, independent on which level it is on. If the current tree contains more than K leafs, all but the best K are discarded; this implements essentially a memory-limited best-first search. $\mathcal{T}_l$ increases very rapidly with l for high values of k, so that we chose a relatively low $k = 3$. However, we do not see this as a major restriction as it seems intuitively implausible that many participants would have been able to retain more than 3 intermediate results in their memory while computing the next step. We do not show results for simpler beam search models ($k = 1$ and $k = 2$) here, as these performed much worse.

We next investigated adding a Pavlovian effect at the meta level. Simple Pavlovian effects capture conditioned behavior that is generally independent of the action being taken (however, they *do* depend on the current state). In the context of metareasoning, they can be used to approximately capture general attraction towards rewarding states, or repulsion from punishing states. Here, our Pavlovian component captures the 'pull towards the big reward' by including a term that considers distance to big reward (the transition from the top right to the top left) for each state, where distance is the minimum amount of choices that lead to the big reward. This models an immediate attraction towards states that are closer to the big reward, and are thus experienced as being more rewarding themselves; it can be viewed as a crude estimate of the long-run utility of these states, based on the assumption that the action leading quickest to the big reward is taken. This component is similar to a Pavlovian component that was introduced by Huys et al. (2012) in their winning model, where the amount of attraction towards the big win transition was estimated as a state-value for each game state independently. Here, we fix the ratios of the Pavlovian values between game states, and simply estimate their magnitude, such that Pavlovian values scale with distance to the big reward in the same manner for each game

state. Let

$$p_{\mathcal{M}}(t_l) = (1 - q(l-1)) \sum_{t_{l-1} \in \mathcal{T}_{l-1}} \left[p^*_{\mathcal{M}}(t_{l-1}) p_{\mathcal{M}}(t_l | t_{l-1}, \beta_{\mathcal{M}}) + \zeta d(t_{l-1}) \right], \tag{83}$$

which is identical to equation 82, except that we add a weighted version of the distance $d(t_{l-1})$ to the big reward from the current state. The weight ζ is an additional parameter to be estimated, and is unconstrained.

Additional confounding factors include the stopping probability, but also the possibility that participants learn during the paradigm. For the probability of stopping $q(l)$ the tree search at the l-th evaluation, the most simple variant is to choose a constant probability $q(l) = \lambda$, which results in an exponential distribution over the lengths of internal evaluation sequences, $p(l) = \lambda \exp(-\lambda l)$. As a more flexible alternative, we considered a negative binomial distribution with one free parameter λ, where we fixed the parameter r to five:

$$q(l) = \binom{l + r - 1}{l} \lambda^l (1 - \lambda)^r. \tag{84}$$

This makes it possible to model stop probabilities that increase or decrease with time. We also explored the possibility to make the stopping probability depend on the reward size by specifying $q(l, r_i)$ as the negative binomial above for all rewards but the big losses, and as a constant $q(l, r_i) = \lambda$ for big losses. As a result, the probability distributions over lengths of internal evaluation sequences are allowed to differ based on the particular sequence being evaluated, so the stopping criterion becomes at least partly a function of the anticipated gains.

Although participants were trained extensively, they might still have exhibited some kind of learning during the task, evidence for which was also found by Huys et al. (2015b). This could affect model fit profoundly, and so we chose to explicitly model this. With respect to our model formulation, which includes a decision process as well as a meta decision process, it could happen both on the choice level or on the meta level. Assuming choice level learning modifies equation 80, such that a learning weight $\omega_i(m)$is multiplied with each term in the sum,

$$Q(\boldsymbol{a}|s, d, m) = \sum_{i=1}^{d} \omega(s_i, a_i, k) \pi_{\mathcal{M}}(a_i) \beta_{r_i} r_i. \tag{85}$$

Here, m indicates the m-th trial, and the weights $\pi_{\mathcal{M}}$ are computed as described in the previous section. Alternatively, learning could happen on the meta level, such that

$$Q(\boldsymbol{a}|s, d, m) = \sum_{i=1}^{d} \pi_{\mathcal{M}}(a_i, m) \beta_{r_i} r_i, \tag{86}$$

where the calculation of the weights $\pi_{\mathcal{M}}$ changes such that in equation 82 the rewards used to compute the softmax $p_{\mathcal{M}}(t_l | t_{l-1}, \beta_{\mathcal{M}})$ are multiplied by their

respective learning weights. The simple learning behavior we assume here is that the learning weights scale with the frequency with which we have encountered the corresponding game transitions so far.

Switching notation slightly, we can write $\omega_k(m)$ for each learning weight, where k indicates the transition that results from taking action a_i in state s_i. Then, we can easily justify our choice by noting that it is essentially equivalent to assuming a binomial distribution for each transition k with parameters θ_k and updating the corresponding parameters $\omega_k(m)$ at trial k according to

$$\begin{aligned}\omega_k(m) &= \mathbb{E}[p(\theta_k|s'_k, s_k)] \\ &\propto p(s'_k|\theta_k, s_k)p(\theta_k) \\ &= \theta_k^{q_m}(1-\theta_k)^{r_m-q_m}\theta_k^{\alpha}(1-\theta_k)^{1-\alpha}, \end{aligned} \tag{87}$$

where s_k and s'_k are starting and end state of transition k, q_m denotes how often s_k has been observed at time m and r_m how often the transition from s_k and s'_k has been observed. We set the prior parameter $\alpha = 0.5$, effectively initializing all observations with the value one. For both choice-level and meta-level learning, we learned transition values separately only for trials with different starting states (We compared to learning transitions separately for each (s,d)-combination, but found this to be inferior).

6.4.1.3 Inference We introduce a prior distribution on the parameters which mainly serves to regularize the inference. This prevents parameters from taking on extreme values, if they are not well-constrained. Let $\boldsymbol{\xi}_n$ denote the vector of parameters for participant n. We choose Gaussian priors on all parameters, and call the (hyper-)parameters denoting means and covariances $\boldsymbol{\zeta} = [\boldsymbol{\mu}_\xi, \boldsymbol{\Sigma}_\xi]^\top$. Inference on both $\boldsymbol{\xi}_n$ and $\boldsymbol{\zeta}$ proceeds through ML estimation via the Expectation Maximization algorithm (Bishop, 2006), which finds a local maximum of the log-likelihood through gradient descent. Let

$$\begin{aligned}\hat{\boldsymbol{\zeta}} &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \log p(\tilde{\boldsymbol{a}}|\boldsymbol{\zeta}) \\ &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \log \prod_{n=1}^{N} p(\tilde{\boldsymbol{a}}_n|\boldsymbol{\zeta}) \\ &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \log \prod_{n=1}^{N} \int_{\boldsymbol{\xi}_n} p(\tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n)p(\boldsymbol{\xi}_n|\boldsymbol{\zeta}) \\ &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \sum_{n=1}^{N} \log \int_{\boldsymbol{\xi}_n} \prod_{k=1}^{K} p(\boldsymbol{a}_{n,k}|\boldsymbol{\xi}_n)p(\boldsymbol{\xi}_n|\boldsymbol{\zeta}), \end{aligned} \tag{88}$$

where N is the number of participants and K is the number of trials per participant, and $\tilde{\boldsymbol{a}} = [\tilde{\boldsymbol{a}}_1^\top, ..., \tilde{\boldsymbol{a}}_N^\top]^\top$ with $\tilde{\boldsymbol{a}}_n = [\boldsymbol{a}_{n,1}^\top, ..., \boldsymbol{a}_{n,K}^\top]^\top$. The vector $\boldsymbol{a}_{n,k}$ denotes the sequence of actions that participant n took in trial k; each entry corresponds to one of the 12 possible transitions in the game as shown in Figure 44 in the

lower right. The notation is complicated by the fact that while before we only discussed sequences of actions within one trial of one individual, we now take into consideration all trials, and all individuals. Thus we add the indices n and k to each action sequence. Vectors $\tilde{a}_n$ simply contain all action sequences of individual n stacked on top of each other, and vector $\tilde{a}$ contains all action sequences of all individuals stacked on top of each other. By factorizing across trials and participants, we assume each trial is independent from any other after conditioning on the parameters.

The expression in equation 88 is difficult to solve because of the integral over $\boldsymbol{\xi}_n$, and both parameters $\boldsymbol{\xi}_n$ and hyper-parameters $\boldsymbol{\zeta}$ are unknown. The EM algorithm estimates both parameters and hyperparameters iteratively; it alternates between performing an E(xpectation) step, which computes an expectation of the log-likelihood while holding parameters $\boldsymbol{\xi}_n$ fixed, and a (M)aximization step, during which the expectation is maximized with respect to the parameters.

Letting $\boldsymbol{\xi} = [\boldsymbol{\xi}_1^\top, ..., \boldsymbol{\xi}_N^\top]^\top$, we can reformulate

$$\begin{aligned}
\log p(\tilde{\boldsymbol{a}}|\boldsymbol{\zeta}) &= \int_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \log p(\tilde{\boldsymbol{a}}|\boldsymbol{\zeta}) \\
&= \int_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \log \frac{p(\tilde{\boldsymbol{a}}, \boldsymbol{\xi}|\boldsymbol{\zeta})}{p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta})} \\
&= \int_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \log \frac{q(\boldsymbol{\xi}) p(\tilde{\boldsymbol{a}}, \boldsymbol{\xi}|\boldsymbol{\zeta})}{p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}) q(\boldsymbol{\xi})} \\
&= \int_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \log \frac{p(\tilde{\boldsymbol{a}}, \boldsymbol{\xi}|\boldsymbol{\zeta})}{q(\boldsymbol{\xi})} - \int_{\boldsymbol{\xi}} q(\boldsymbol{\xi}) \log \frac{p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta})}{q(\boldsymbol{\xi})} \\
&= \mathcal{L}(q(\boldsymbol{\xi}), \boldsymbol{\zeta}) + \mathrm{KL}(q(\boldsymbol{\xi}) || p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta})),
\end{aligned}$$

where KL denotes the Kullback-Leibler (KL) divergence. It is non-negative, and minimized for $q(\boldsymbol{\xi}) = p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta})$, so $\mathcal{L}$ is a lower bound on $\log p(\tilde{\boldsymbol{a}}_n|\boldsymbol{\zeta})$.

In the EM algorithm, the i-th E(xpectation)-step consists of maximizing $\mathcal{L}(q(\boldsymbol{\xi}), \boldsymbol{\zeta}^{(i)})$ with respect to $q(\boldsymbol{\xi})$ while holding $\boldsymbol{\zeta}^{(i)}$ fixed. Since this means that the log-likelihood is constant, maximization of $\mathcal{L}$ is equivalent to minimization of the KL, and so we set $q(\boldsymbol{t}) = p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)})$. In the subsequent M(aximization)-step, $q(\boldsymbol{\xi})$ is held fixed and we compute $\boldsymbol{\zeta}^{(i+1)} = \mathrm{argmax}_{\boldsymbol{\zeta}} \mathcal{L}$, where $\mathcal{L}$ takes the form

$$\begin{aligned}
&\mathcal{L}(q(\boldsymbol{\xi}), \boldsymbol{\zeta}) \\
&= \int_{\boldsymbol{\xi}} p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)}) \log p(\tilde{\boldsymbol{a}}, \boldsymbol{\xi}|\boldsymbol{\zeta}) && - \sum_{\boldsymbol{t}} p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)}) \log p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)}) \\
&= \int_{\boldsymbol{\xi}} p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)}) \log p(\tilde{\boldsymbol{a}}, \boldsymbol{\xi}|\boldsymbol{\zeta}) && - C \\
&= \mathbb{E}_{p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)})} [\log p(\tilde{\boldsymbol{a}}, \boldsymbol{\xi}|\boldsymbol{\zeta})] && - C, \qquad (89)
\end{aligned}$$

where the second term is constant with respect to $\boldsymbol{\zeta}$. This will increase both $\mathcal{L}$ and KL, and as a result of that the log-likelihood. The two steps are then iterated until convergence.

In our application, following (Huys et al., 2012), we approximate $p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)})$ with a Gaussian around its mode for the E step; this is known as the Laplacian approximation. We do this individually for each participant, such that $p(\boldsymbol{\xi}_n|\tilde{\boldsymbol{a}}, \boldsymbol{\zeta}^{(i)}) = \mathcal{N}(\hat{\boldsymbol{\xi}}_n^{(i)}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}_n}^{(i)})$. Since the posterior is proportional to the joint, we can simply find the mode of the joint

$$\begin{aligned}\hat{\boldsymbol{\xi}}_n^{(i)} &= \operatorname*{argmax}_{\boldsymbol{\xi}_n} p(\tilde{\boldsymbol{a}}_n, \boldsymbol{\xi}_n|\hat{\boldsymbol{\zeta}}^{(i-1)}) \\ &= \operatorname*{argmax}_{\boldsymbol{\xi}_n} p(\tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n) p(\boldsymbol{\xi}_n|\hat{\boldsymbol{\zeta}}^{(i-1)}).\end{aligned} \tag{90}$$

The variances are approximated by the second moments $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}_n}^{(i)}$ around $\hat{\boldsymbol{\xi}}_n^{(i)}$. In the i-th M step, the priors $p(\boldsymbol{\xi}_n|\boldsymbol{\zeta})$ are updated. Maximization of equation 89 yields

$$\begin{aligned}\hat{\boldsymbol{\mu}}_\xi^{(i)} =& N^{-1} \sum_n \hat{\boldsymbol{\xi}}_n^{(i)} \\ \hat{\boldsymbol{\Sigma}}_\xi^{(i)} =& N^{-1} \sum_n \left(\hat{\boldsymbol{\xi}}_n^{(i)} (\hat{\boldsymbol{\xi}}_n^{(i)})^\top + \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}_n} \right) - \boldsymbol{\mu}_\xi^{(i)} (\boldsymbol{\mu}_\xi^{(i)})^\top,\end{aligned} \tag{91}$$

where we remind ourselves that we defined $p(\boldsymbol{\xi}_n|\boldsymbol{\zeta}) = \mathcal{N}(\boldsymbol{\mu}_\xi, \boldsymbol{\Sigma}_\xi)$. We repeat the procedure several times with different initial starting values, as the EM algorithm only finds local optima. As a result, we obtain ML estimates $\hat{\boldsymbol{\zeta}}$ as averages of $\hat{\boldsymbol{\xi}}$ and MAP estimates of $\hat{\boldsymbol{\xi}}$ where $\hat{\boldsymbol{\zeta}}$ acts as a prior.

6.4.1.4 Model comparison For Bayesian model comparison, we integrate out all individual-level parameters, and penalize more complex models at the group level. This can be achieved by approximating the Bayesian Information Criterion (BIC; see Schwarz (1978)) using importance sampling. The BIC itself approximates the log marginal likelihood for model $\mathcal{M}$,

$$\log p(\tilde{\boldsymbol{a}}|\mathcal{M}) = \int_{\boldsymbol{\zeta}} p(\tilde{\boldsymbol{a}}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}|\mathcal{M}) \tag{92}$$

In the context of behavioral modeling, the approximation using importance sampling was proposed by Huys et al. (2011). Following the authors, we call the resulting approximation integrated BIC (iBIC),

$$\begin{aligned}\text{BIC} &= -2 \log p(\tilde{\boldsymbol{a}}|\hat{\boldsymbol{\zeta}}) + \|\boldsymbol{\zeta}\|_0 \log(\|\tilde{\boldsymbol{a}}\|_0) \\ &= -2 \log \prod_n \int_{\boldsymbol{\xi}} p(\tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}) p(\boldsymbol{\xi}|\hat{\boldsymbol{\zeta}}) + \|\boldsymbol{\zeta}\|_0 \log(\|\tilde{\boldsymbol{a}}\|_0) \\ &\approx -2 \sum_n \log M^{-1} \sum_m p(\tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}^{(m)}) + \|\boldsymbol{\zeta}\|_0 \log(\|\tilde{\boldsymbol{a}}\|_0) \qquad \boldsymbol{\xi}^{(m)} \sim p(\boldsymbol{\xi}|\hat{\boldsymbol{\zeta}}) \\ &= \text{iBIC},\end{aligned} \tag{93}$$

where we approximate the integral over subject-specific parameters $\boldsymbol{\xi}$ by sampling them from the prior $p(\boldsymbol{\xi}|\hat{\boldsymbol{\zeta}})$ M=10'000 times for each model.

6.4.2 Results

The results suggest that participants are most likely to use a depth-first search combined with a non-constant stop probability: Across all meta process models, beam search scored worse than depth-first search ('metaDF'), and a constant stop probability was always worse than either using the negative binomial ('stopNegbin'), or the combination with a separate constant stop probability for big losses ('stopCombo'); Figure 34 does not show models that included these 'inferior' ingredients which when present always caused the corresponding models to become worse.

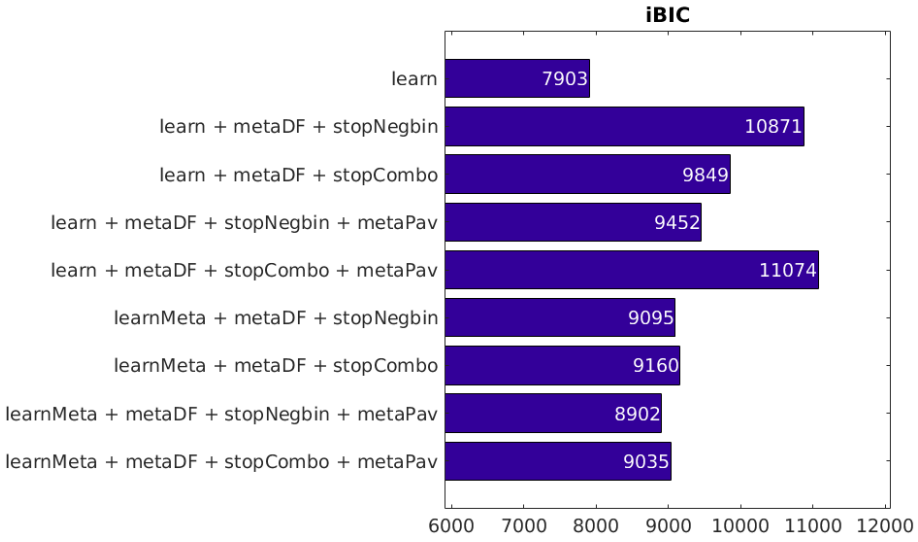


Figure 34: Model comparison results of the best performing behavioral models. The iBIC is an approximation of the negative model log likelihood, so smaller values are better.

The importance and influence of non-stationarity was quite notable. In terms of the stopping probability, the conclusions depended on how learning was modelled. When modelling learning at the metareasoning level, the 'stopNegbin' model outperformed 'stopCombo', with further improvements when adding Pavlovian attraction. On the other hand, when modelling learning at the choice level, a model that included a termination factor when encountering a large loss ('learn + metaDF + stopCombo') outperformed a model without such a factor ('learn + metaDF + stopNegbin'). However, this pattern reversed when adding Pavlovian attraction to big reward ('metaPav'). Thus, the best model that included the meta-decision process weights $\pi_{\mathcal{M}}$ was 'learnMeta + metaDF + stopNegbin + metaPav'. That is, amongst the metareasoning models we tested, the data pointed at a combination of depth-first search, negative binomial stop probability, instrumental learning on the meta-level and Pavlovian attraction towards the big reward on the meta level.

Figure 35 shows the standard BICs for each individual participant; it shows

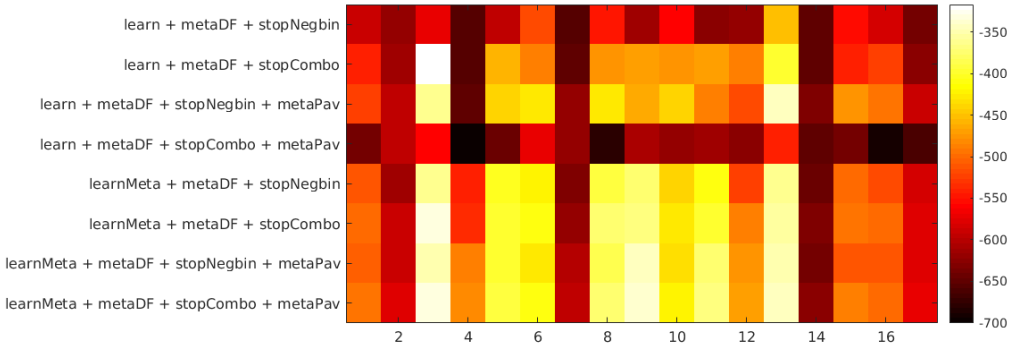


Figure 35: BIC's for individual participants for each of the metareasoning models. The winning model is the second from the bottom. Brighter values indicate better model fit. About half the individuals (2, 3, 6, 10, 14, 15, 16, 17) are better described by other models.

that the winning model only fits 8 out of the 17 participants best. Model 'learn + metaDF + stopCombo' explains participant 3 much better, model 'learnMeta + metaDF + stopCombo' explains participant 6 much better, and 'learn + metaDF + stopNegbin + metaPav' is a better model for participant 16, for example.

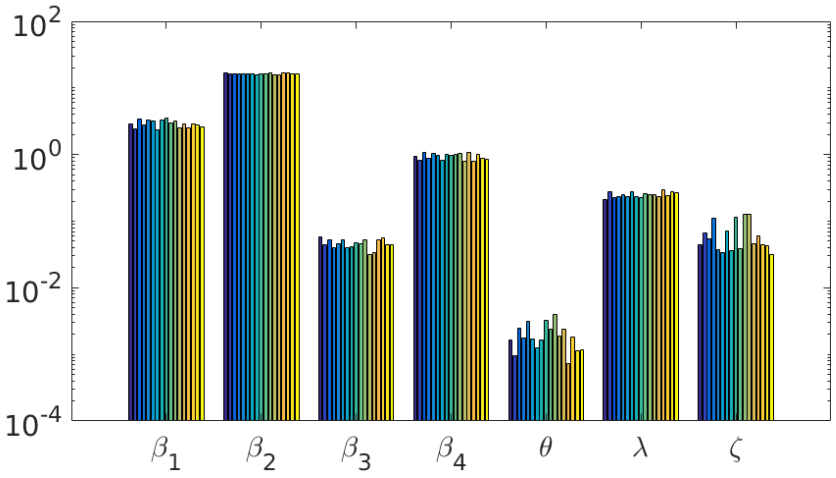


Figure 36: Winning model's parameter estimates. Each bar represents one participant. Individual variation is mainly visible in the meta-reward sensitivity parameter θ and the Pavlovian parameter ζ.

The winning model's parameter estimates are shown in Figure 36. It is notable that most of the individual variability is captured by two of the three metareasoning parameters: While the stopping parameter λ is very much the same for all individuals, substantial variability can be seen in the meta-reward sensitivity parameter θ and the Pavlovian attraction parameter ζ. Figures 37 to 39 show the model fits independently for each combination of starting state and choice length, for lengths 3 to 5; the fits are plotted as red lines on top of

the histograms that show choice data across all participants. More detailed plots are shown in Appendix A. While many patterns can be captured very well, e.g. Figure 37 top left and bottom right, or Figure 39 top left, clearly the model also severely over- or underestimates some choice sequence probabilities, e.g. Figure 37 top right, or Figure 39 middle left.

However, with an iBIC of 8902 this model was still worse than the simplest model 'learn' (iBIC of 7903) that did not explicitly model the metareasoning process , i.e. $\pi_{\mathcal{M}}(a_i) = 1 \; \forall a_i$. This model has one reward sensitivity parameter for each reward size ($|\boldsymbol{\xi}| = |\boldsymbol{\beta}| = 4$) and no other parameters because the choice-level learning did not induce any free parameters. The original mean field model without learning as defined in equation 77 scored an iBIC of 8888.

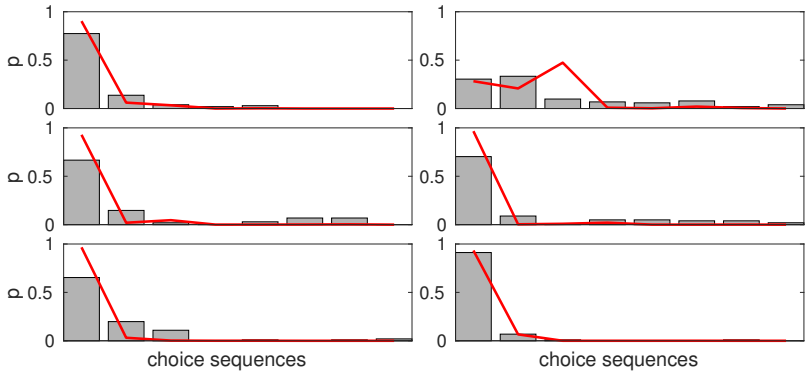


Figure 37: Winning model's fit to choice data for choice sequences of length 3. Each plot shows a histogram (grey bars) over choice data from all participants for one particular game starting state, with state 1 in the top left and state 6 in the bottom right. Red lines indicate model fit. Thus, each bar represents the frequency of one particular choice sequence; we sorted the sequences in descending order of total reward earned, so a perfect player would always choose the first sequence. Appendix A shows the same data in a slightly different format, such that the labels according to the actual choice sequences are displayed, too. Apart from the third best choice sequence in the top right plot (starting state 2), the model fits data for choice sequences of length 3 well.

6.4.3 Discussion

Unfortunately, none of the 24 metareasoning models was able to outperform the basic model that did not include a metareasoning process, suggesting that none of the simple tree-search strategies that we tested can explain the internal evaluation process across individuals well.

One of the likely reasons is that any kind of greedy tree search chooses the best among the set of possible next evalutation at each step; as a consequence, any evaluation sequence that chooses a smaller reward starting at a particular state of the search tree will always yield a smaller metareasoning weight, as compared to sequences that choose a bigger reward. Instead, participants might

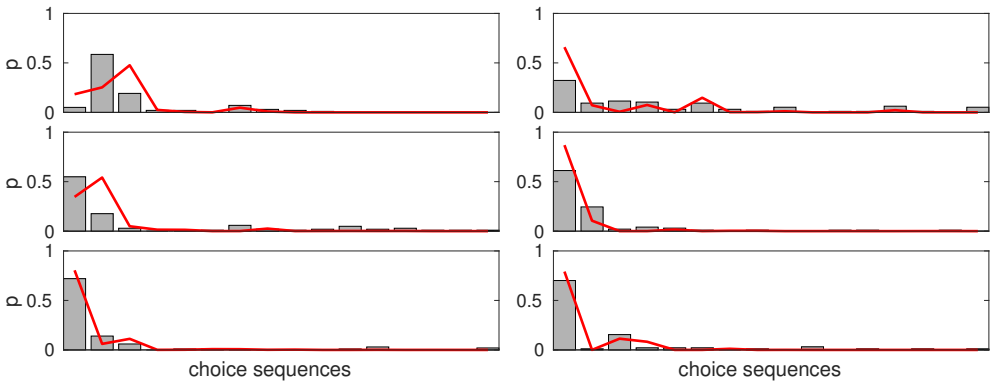


Figure 38: Winning model's fit to choice data for choice sequences of length 4. Some details are missed by the model - for example, in the top right plot (starting state 1), the probabilities of the best and third-best choice sequences (with corresponding rewards 140,-70,-70,140 and 140,-20,-20,-20) are overestimated while the probability of the second-best choice sequence (rewards 140,-20,-20,20) is under-estimated. See Appendix A for more detailed histograms.

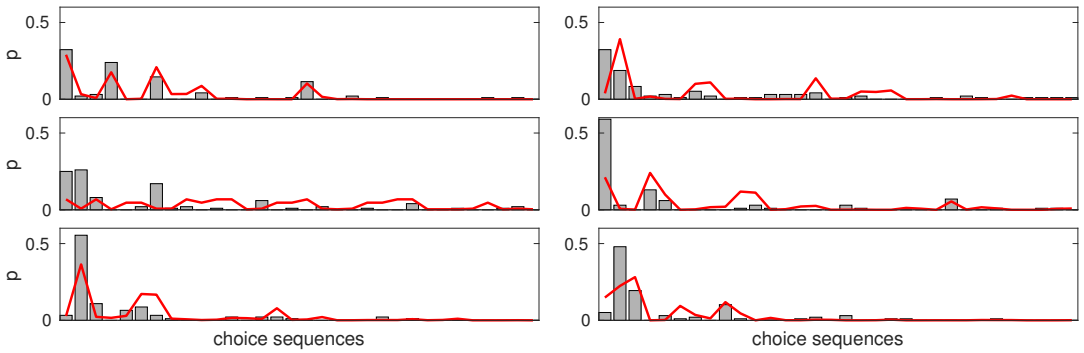


Figure 39: Winning model's fit to choice data for choice sequences of length 5. The pattern in the middle left plot (starting state 3) is completed missed by the model.

not do greedy search at all, or use more complex stopping criteria. It is also possible that they adopt a different node selection strategy; for example they might do a variant of depth-first search, where they expand a transition at the deepest node in the search tree only if the total accumulated reward is above a certain threshold.

Another reason why the model fits are below our expectations might be that tree search strategies differ greatly between individuals (see Figure 35), such that no single search strategy can explain the data well on the group level. From this we conclude that participants' metareasoning in the pruning task is much more complicated than we had initially assumed, such that we moved on to collect more data (eye gazes) and built models to incorporate these additional data.

6.5 Inferring metareasoning strategies from choices and gaze patterns

Because the results of the previous section suggest that behavioral choices alone might be insufficient to constrain the metareasoning process, we move to incorporate gaze position in the hope that this might be more directly indicative of the sequence of internal computations individuals perform. That is, we consider the possibility that individuals may look at the states they are currently considering transitioning to, and that the sequence of eye movements may reflect the sequence of internal evaluations they have gone through. If this is true, then this sequence should be informative about the metareasoning process, i.e. the process by which they decide which move to evaluate. Below, we first examine this proposition in the data and then describe a likelihood model for how to relate eye gaze position to internal evaluations. As these models consider individual trials, we also have to adapt the likelihood of the choices. Finally, we examine and fit joint generative models of both choice probability and eye gaze position.

6.5.1 Visualizations

Before modeling the gaze positions, we first characterize the acquired data. Figures 40, 41 and 42 show trials from 3 representative subjects. Visual inspection of the data reveals several between-subject issues. The first issue is that some participants move their eyes more than others. A participant who scored well and moved his or her eyes a lot is shown in Figure 40. This appears to be ideal for modelling, as it appears to be rich in information. Another participant who did not move his or her eyes much in the planning phase is shown in Figure 41. As this subject also performed well and hence must have performed evaluations, it might be that much of these occurred through covert shifts of attention without overt gaze shifts. As this is, however, not known, this individual's data is less directly informative about the metareasoning process. Finally, some participants' gaze shifts did not appear to have any obvious relationship to evaluations. Figure 42 shows such an example. What is clear from these examples, is that individual variation in how informative gaze position is about internal evaluations might

well be substantial and should be taken into account.

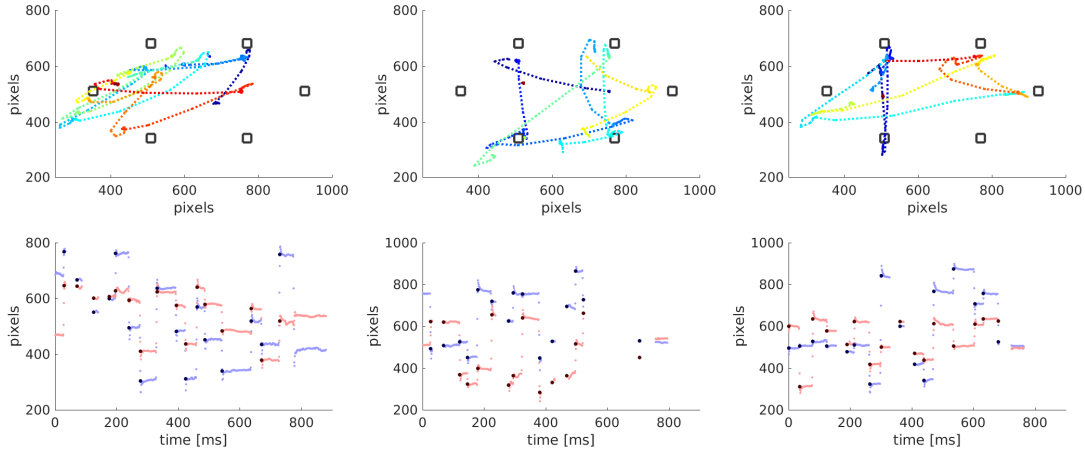


Figure 40: Eye-tracking data from the planning phase of a participant who was performing very well in terms of total score, and explicitly looked at many states. Each plot in the top row shows data from one example trial on a schematic representation of the screen, where game states are indicated by black rectangles, and gaze data is represented as small circles connected by dotted lines. Time is encoded by color (dark blue indicates the start of the planning phase and red the end). Plots in the bottom row show the same data in a different way: one trace each marks horizontal (blue) and vertical (red) gaze position on the screen. The small dots show the raw data while bigger and more strongly colored dots indicate the fixation data that was automatically extracted from the eyetracking device and which we used for the model of gazes.

To further explore the gaze data and get a sense for how informative the data are overall, we attempted to directly map eye movements onto explorations of decision-trees, i.e. onto moves along the transitions in the maze. To do so, we converted gazes into transitions between game states: we defined a"thought" about a game state as at least 50ms of eye-tracking data that was closer to that particular state than to any other state. A 'causal transition' was registered whenever thoughts about two different game states followed a game transition forwards, meaning that such a transition was allowed by the game rules. An 'acausal transition' was registered whenever thoughts about two different game states followed a game transition *backwards*. Figure 43 compares causal eye-tracking transitions with participants' choice behavior individually for all starting states and choice sequence lengths. Clearly, some patterns of the choice data appear in the causal eye transitions. For example, participants' gaze focuses much more on transitions 1 and 3 than on transition 2, and this is reflected in the frequent choice of this transition. When starting in state 1 (rows 1, 7, 13, and 19 in the histograms), participants focus more on transition 1 than when starting in any other state. Figure 44 summarizes causal vs acausal vs choice behavior, aggregated across choice sequence lengths and starting states. 63.8% of valid

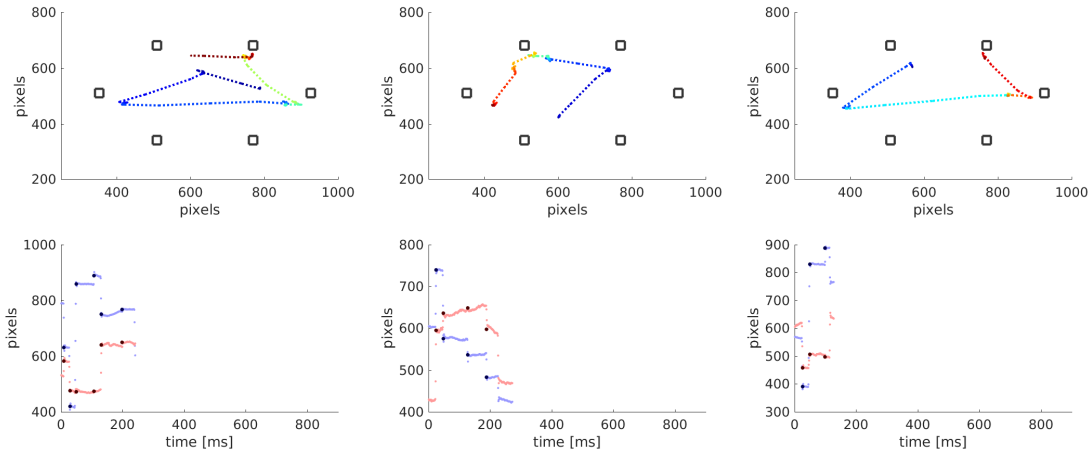


Figure 41: Eye-tracking data from the planning phase of a participant who was performing very well, but did not look at many states before deciding relatively quickly on a sequence of choices.

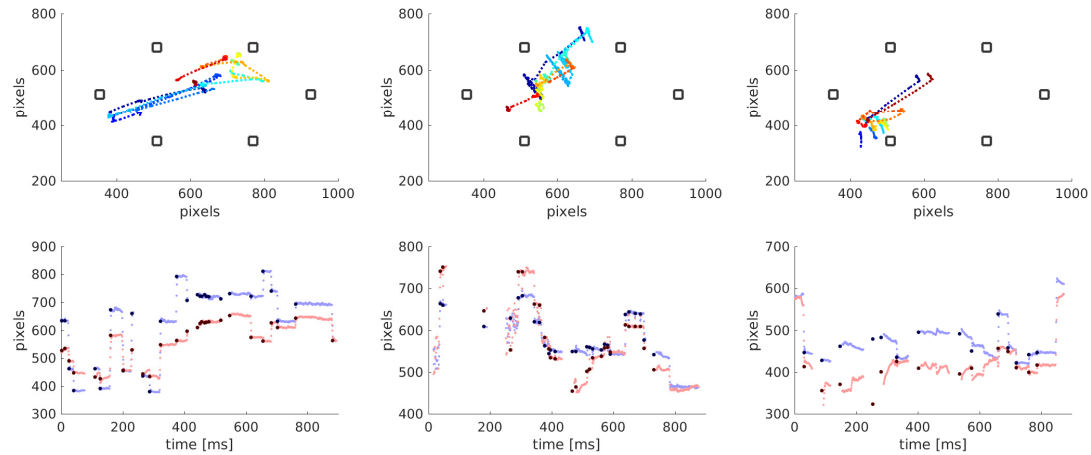


Figure 42: Eye-tracking data from the planning phase of a participant whose gaze positions did not reveal an obvious relationship to internal evaluations. Of note, though, this participant also performed poorly in terms of choices.

transitions were causal while the remaining 36.2% were acausal. It is important to consider the implications of this, when combined with plots of individual trial data, such as Figures 40, 41 and 42. For instance, participants often looked back and forth between two states, suggesting that they traversed the corresponding transition in their minds several times back and forth. Overall, however, the data do appear to reflect choices and hence might carry some information about the metareasoning process.

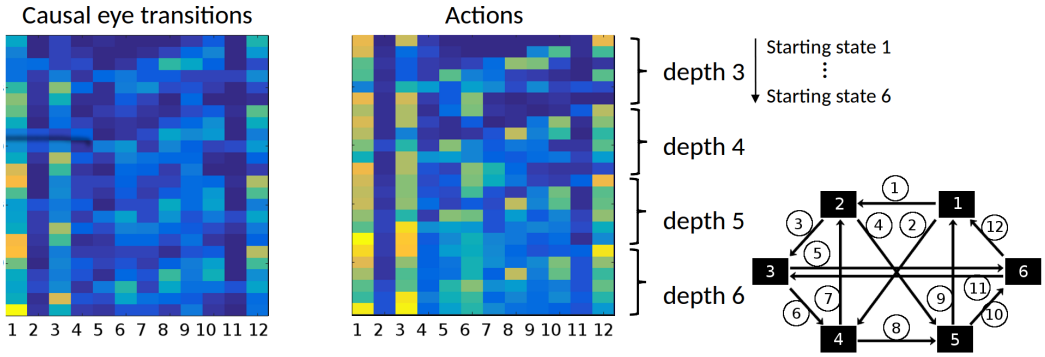
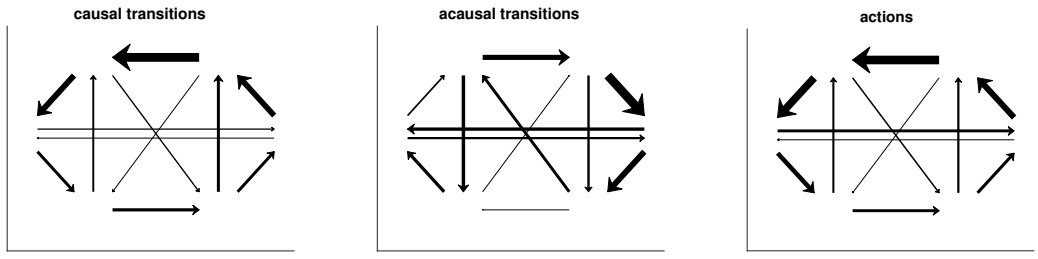


Figure 43: Direct comparison of the frequency of causal gaze transitions and action, averaged across participants. The schema in the lower right shows the numbers we assigned to each of the 12 game transitions, as well as the numbers of the 6 game states themselves. In the 2-d histograms on the left, each transition corresponds to a column, with the transition's number appearing below. The first six rows show the transition probabilities for depth 3 problems starting in state 1 to 6. The next six rows show the probabilities for all depth 4 problems, and so on. Some of the pattern in the action data clearly seems to be captured by the eye data.



(a) (b) (c)

Figure 44: Comparison of transitions extracted from gaze positions and actions, averaged over trials of all depths and starting states. The thickness of the arrows indicates the frequency of the corresponding transitions.

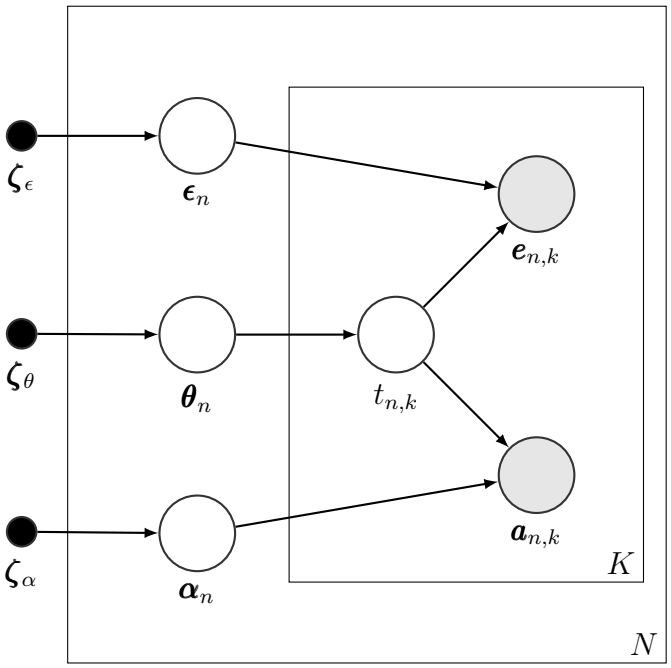


Figure 45: Graphical model for the full generative model including batch action and eye data; these are assumed to be conditionally independent given the (unobserved) evaluation processes.

6.5.2 Computational model

We specify a full model that implements a greedy metareasoning process. We believe this to be a relatively weak constraint, as it seems intuitive that participants evaluate state transitions in the order that they appear in the game. As such, this model should provide a relatively unconstrained examination of the metareasoning process. The model factorizes over study participants as well as individual trials for each participant; it decomposes into three parts for each trial. Let $p(t_{n,k}|\boldsymbol{\theta}_n)$ be the probability of the search sequence $t_{n,k}$ (as defined in section 6.5.2.1) in trial k for participant n, parametrized by the vector of parameters $\boldsymbol{\theta}_n$. Next, define $p(\boldsymbol{a}_{n,k}|t_{n,k}, \boldsymbol{\alpha}_n)$ and $p(\boldsymbol{e}_{n,k}|t_{n,k}, \boldsymbol{\epsilon}_n)$ as the probability of action $\boldsymbol{a}_{n,k}$ and eye data $\boldsymbol{e}_{n,k}$, given search tree sequence $t_{n,k}$. Vectors $\boldsymbol{\epsilon}_n$ and $\boldsymbol{\alpha}_n$ specify the parameters for the n-th participant.

The vector $\boldsymbol{a}_{n,k}$ denotes the sequence of actions that participant n took in trial k; each entry corresponds to one of the 12 possible transitions in the game as shown in Figure 44. The vector $\boldsymbol{e}_{n,k}$ denotes the corresponding fixations from the eye-tracking data during the planning phase of the trial (prior to choice input); each entry corresponds to a position on the screen.

Given these ingredients, we can decompose the full joint distribution over all

data,

$$p(\tilde{\boldsymbol{e}}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{t}}, \boldsymbol{\epsilon}, \boldsymbol{\alpha}, \boldsymbol{\theta} | \boldsymbol{\zeta}_\epsilon, \boldsymbol{\zeta}_\alpha, \boldsymbol{\zeta}_\theta) = \prod_{n=1}^{N} p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{t}}_n | \boldsymbol{\epsilon}_n, \boldsymbol{\alpha}_n, \boldsymbol{\theta}_n) p(\boldsymbol{\theta}_n | \boldsymbol{\zeta}_\theta) p(\boldsymbol{\epsilon_n} | \boldsymbol{\zeta}_\epsilon) p(\boldsymbol{\alpha}_n | \boldsymbol{\zeta}_\alpha) \tag{94}$$

$$p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{t}}_n | \boldsymbol{\epsilon}_n, \boldsymbol{\alpha}_n, \boldsymbol{\theta}_n) = \prod_{k=1}^{K} p(\boldsymbol{a}_{n,k} | t_{n,k}, \boldsymbol{\alpha}_n) p(\boldsymbol{e}_{n,k} | t_{n,k}, \boldsymbol{\epsilon}_n) p(t_{n,k} | \boldsymbol{\theta}_n), \tag{95}$$

where N is the number of participants and K is the number of trials per participant, $\tilde{\boldsymbol{a}} = [\tilde{\boldsymbol{a}}_1^\top, ..., \tilde{\boldsymbol{a}}_N^\top]^\top$ with $\tilde{\boldsymbol{a}}_n = [\boldsymbol{a}_{n,1}^\top, ..., \boldsymbol{a}_{n,K}^\top]^\top$, and, similarly, $\tilde{\boldsymbol{e}} = [\tilde{\boldsymbol{e}}_1^\top, ..., \tilde{\boldsymbol{e}}_N^\top]^\top$ with $\tilde{\boldsymbol{e}}_n = [\boldsymbol{e}_{n,1}^\top, ..., \boldsymbol{e}_{n,K}^\top]^\top$ and $\tilde{\boldsymbol{t}} = [\tilde{\boldsymbol{t}}_1^\top, ..., \tilde{\boldsymbol{t}}_N^\top]^\top$ with $\tilde{\boldsymbol{t}}_n = [t_{n,1}, ..., t_{n,K}]^\top$. Vectors $\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{e}}$ and $\tilde{\boldsymbol{t}}$ contain all data from all participants stacked on top of each other, while vectors $\tilde{\boldsymbol{a}}_n$, $\tilde{\boldsymbol{e}}_n$ and $\tilde{\boldsymbol{t}}_n$ contain data from all trials of participant n stacked on top of each other.

We assume that parameters are characterized by Gaussian prior distributions $p(\boldsymbol{\epsilon}_n | \boldsymbol{\zeta}_\epsilon)$, $p(\boldsymbol{\theta}_n | \boldsymbol{\zeta}_\theta)$ and $p(\boldsymbol{\alpha}_n | \boldsymbol{\zeta}_\alpha)$, where the (hyper-)parameters $\boldsymbol{\zeta}_\alpha$, $\boldsymbol{\zeta}_\epsilon$, $\boldsymbol{\zeta}_\theta$ contain the respective Gaussian means and covariances. The graphical model corresponding to this formulation is depicted in Figure 45. The three basic model ingredients are the probability distribution $p(t_{n,k} | \boldsymbol{\theta}_n)$ over search trees (sequences of internal evaluations), the distribution $p(\boldsymbol{a}_{n,k} | t_{n,k}, \boldsymbol{\alpha}_n)$ over actions given search trees and the distribution $p(\boldsymbol{e}_{n,k} | t_{n,k}, \boldsymbol{\epsilon}_n)$ over gazes given search trees. Their detailed explanation follows below.

6.5.2.1 Metareasoning process Our initial approach here differs from the one in section 6.4.1.1. Rather than explicitly defining several metareasoning models $\mathcal{M}$ and comparing them, we wanted to examine to what extent the eyetracking data might provide direct evidence about the sequences of internal evaluations and hence the metareasoning process in an unconstrained fashion. That is, we wanted to infer the most likely sequences of internal evaluations from the data without prescribing a particular ordering a priori.

Although we do not prescribe any particular node selection strategy (i.e., depth-first or beam search) here, we rest the generative model on the assumption that individuals search the tree in a greedy fashion - in other words, we assume that during the process of internal evaluation, individuals cannot skip evaluations of nodes in the decision tree.

We define the probability distribution $p(l)$ over the length l of a sequence of internal evaluations, which embodies our general expectations of how many transitions each participant might think about during the planning phase of each trial. Let for trial k and subject n

$$p(l, t_{n,k} | \theta_n) = p(l) \prod_{i=1}^{l} p((t_{n,k})_i | \theta_n, (t_{n,k})_{i-1}), \tag{96}$$

where $(t_{n,k})_i$ denotes the state of the search tree $t_{n,k}$ after the i-th internal evaluation. The last term in equation 96, $p((t_{n,k})_i|\theta_n, (t_{n,k})_{i-1})$, implements a softmax with parameter θ_n across the values of all possible next evaluations according to the current state of the tree (see Figure 46 for an example illustration).

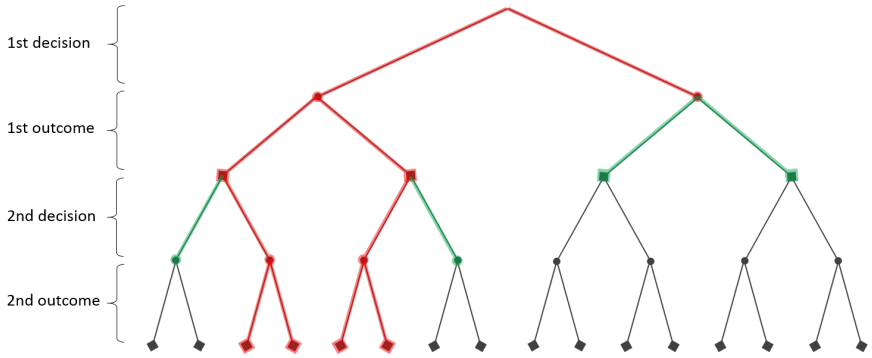


Figure 46: Green lines indicate possible next evaluations according to the current state of the search tree, which is shown in red. Next evaluations are determined in a greedy fashion, meaning that all possibilities to expand the search tree at any node that is not fully expanded yet are considered. This is a much less strict constraint than e.g. depth-first search.

We chose an exponential distribution over sequence lengths,

$$p(l) = p(l|\mu) = \mu^{-1} \exp(-l/\mu), \tag{97}$$

where we estimated parameter μ by fitting the function to the mean number of 'state transitions' across participants from the eye tracking data, prior to inference about all other model parameters. Figure 47 shows the number of transitions and their durations, as well as the fitted functional form. Note that in this relatively simple formulation, this part of the model contains only one single free parameter θ_n.

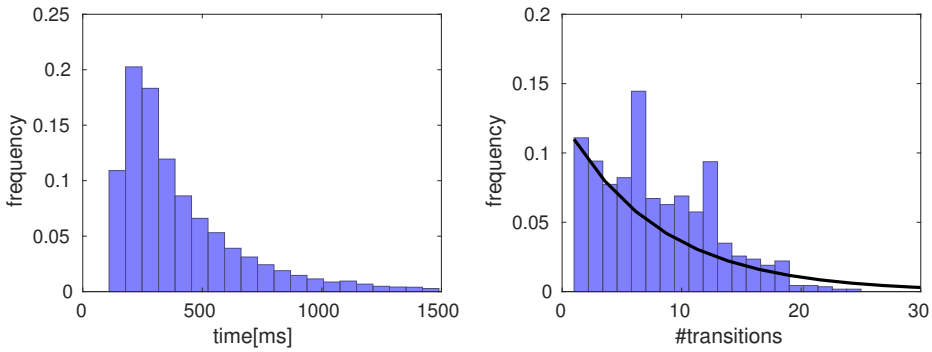


Figure 47: Left: Average duration of game state transitions in each trial, as derived from the gaze data (see text). Right: Average number of transitions per trial. The black line indicates a fitted exponential distribution with $\mu = 8.0389$.

6.5.2.2 Action likelihood We model the probability of actions conditioned on internal evaluation processes in a similar manner as we have done for the behavioral analysis in equation 80, with the difference that we drop the mean tree weights from the action-values. Instead, the probability of an action sequence is now conditioned on a particular evaluation sequence in such a way that we substitute the true reward with a constant value ρ_n if a particular action has not been evaluated. Let

$$Q(\boldsymbol{a}_{n,k}|t_{n,k},\boldsymbol{\alpha}_n) = \sum_i r(a_{n,k,i}|t_{n,k})$$
$$r(a_{n,k,i}|t_{n,k}) = \begin{cases} \beta_{r_{n,k,i}} r_{n,k,i} & \forall a_{n,k,i} \in t_{n,k} \\ \rho_n & \forall a_{n,k,i} \notin t_{n,k} \end{cases} \tag{98}$$

where $r_{n,k,i} = r(a_{n,k,i})$ and $\boldsymbol{\alpha}_n = \{\beta_n, \rho_n\}$. The assumption here is that participants do not have a preference for actions they did not evaluate internally. Furthermore, by letting ρ_n be a model parameter, we can explain how likely participants are to take actions that they did not 'think about'. For instance, $\rho \to -\infty$ indicates participants that never choose actions they have not thought about. Intuitively, we can imagine that a participant might have only evaluated sequences that (s)he feels are worse than random, so that (s)he prefers to explore other possibilities instead.

The probability distribution over action sequences results from computing the softmax over action values,

$$p(\boldsymbol{a}_{n,k}|t_{n,k},\boldsymbol{\alpha}_n) = \frac{e^{Q(\boldsymbol{a}_{n,k}|t_{n,k},\boldsymbol{\alpha}_n)}}{\sum_{\boldsymbol{a}'_{n,k}} e^{Q(\boldsymbol{a}'_{n,k}|t_{n,k},\boldsymbol{\alpha}_n)}}. \tag{99}$$

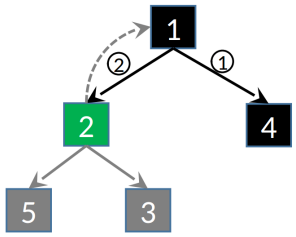


Figure 48: In this search tree, edge labels correspond to the order of internal evaluations in the evaluation process. The search tree consists of the black and green nodes. The green node is the current one, with grey lines indicating possible next evaluations. Three valid evaluations are available: Either of the children of the green node can be evaluated about next; alternatively, it is also possible to 'check' by traversing back to the tree root.

6.5.2.3 Gaze model The final ingredient is the mapping from a search tree $t_{n,k}$ to a sequence of gaze positions, which we extracted from the raw eyetracking

data as explained in section 6.3. The main challenges are that dwell-times (number of consecutive gaze positions that are classified to belong to the same game state) vary and need to be allowed for; that gaze shifts might not correspond to or fully reflect a legal transition (and might not even always correspond to which game states participants were thinking about); that observed gaze positions might be noisy; and that there is a one-to-many mapping between search trees and internal evaluation sequences, as we explain below.

Any given search tree prescribes a sequence of game states, but the corresponding sequence of states that individuals *look at* does not necessarily need to be the same. To see that, consider the search tree shown in Figure 48, where first the right child (game state 4) and then the left child (game state 2) of the root node (game state 1) is expanded. Thus, the corresponding sequence of game states is 1-4-2. However, we expect the sequence of states that individuals look at to be different, because they are first evaluating the transition 1-4 and then the transition 1-2, which should result in the complete looked-at state sequence 1-4-1-2. We call this phenomenon 'backtracking', as it results in extra insertion of the parent node when its second child is expanded (it's a 'backtrack' to the parent). Thus, we extend our concept of a search tree so that any node in the tree may contain a list of checking evaluations, a 'virtual trace'.

Additionally, the data strongly suggest that participants at times 're-think' some of their thoughts, a behavior we call 'checking'; this is likely why so many of the transitions in the eye data are acausal (see section 6.5.1). Checking is equivalent to traversing edges of the search tree that have already been expanded before. An example is shown by the dotted gray line in Figure 48. We can think of any sequence of checking moves as attached to a node in the search tree. The sequence may contain any edge that has already been expanded in the current search tree, and each edge may be traversed forwards (down the tree) or backwards (up the tree). An example of such a 'virtual trace' is shown in Figure 49 on the right. In the rest of this section, we'll refer to any state sequence resulting from applying the principles of checking and backtracking simply as the state sequence $x_1, ..., x_M$, where M is the length of the sequence.

To map any such state sequence to a gaze sequence, we need to (i) map gaze positions to game states and (ii) be able to 'jump' game states, because individuals might not look at every single state that they think about. Here, we opted for a fully generative probabilistic approach using a Hidden Markov Model (HMM) and the forward-backward algorithm. A first-order HMM assumes a first-order Markov process on M unobserved (hidden) states x, such that the probability of being in a certain state x at time t depends only on the state at time $t-1$; it is independent on the previous history of the system. States emit observations (= gazes) with certain probabilities and so the HMM is completely characterized by a transition matrix T and an emission matrix E.

The transition matrix probabilities $p(X_t = x_m | X_{t-1} = x_{m'})$[1] specify the probability of transitioning from state x_m at time t to state $x_{m'}$ at time $t+1$. The most simple way to specify it is to allow non-zero entries only for $p(X_{t+1} =$

[1] capital letters denote random variables

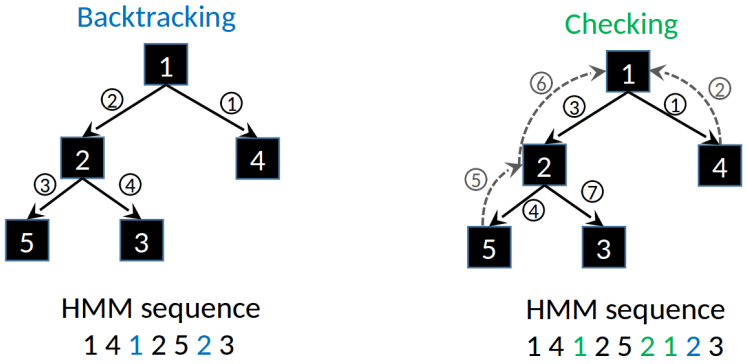


Figure 49: Mapping from search trees to hidden markov model sequences. Left: Example of backtracking, where a node is always inserted into the state sequence prior to any child that is expanded, specifically also if the second child is expanded. Right: Example of checking behavior, where transitions between nodes that were expanded before can be traversed again, and the corresponding nodes are inserted into the state sequence.

$x_m|X_t = x_m)$ and $p(X_{t+1} = x_{m+1}|X_t = x_m)$. This means that the model can only either stay at the current state, or progress to the immediate next one. The probability to switch to the next state can naively be obtained by dividing the length of the state sequence through the length of the gaze sequence. The sum of switching and staying must equal one, and the probability to stay is the same for each state.

We are able to restrict our transition matrices in this way, because we effectively extended the concept of the search tree with checking moves such that the one-to-many mapping between search trees and possible internal evaluation sequences does not need to be resolved by using the HMM, but has already been dealt with in constructing the state sequence $x_1, ..., x_M$. Being able to stay in one state for many time points ($p(X_{t+1} = x_m|X_t = x_m) > 0$) deals with the issue of differing dwell-time lengths.

For the noise model, which results in the emission matrix, we considered the histogram of the eye data during the planning phase across all trials and all participants (Figure 50). Spherical shapes are visible around each of the game states, so we chose a mixture of Gaussians to describe the probability of an gaze position on screen, where the mean of each Gaussian is at the center of one of the game states. We estimated the covariances $\mathbf{\Sigma}_e$ from the histogram, by first clustering gazes according to predefined screen regions shown in Figure 50 on the right, and then estimating the covariances for each cluster separately. The emission matrix contains the probabilities $p(E_t = e_t|X_t = x_m, \mathbf{\Sigma}_e)$ that an gaze e_t is generated by the state x_m and is computed according to these Gaussians.

The forward-backward algorithm for HMM computes the smoothing distribution, i.e. the posterior marginal $p(X_1, ..., X_T|E_1, ..., E_T)$ of all hidden states given the observations (Rabiner and Juang, 1986). From this, we can compute

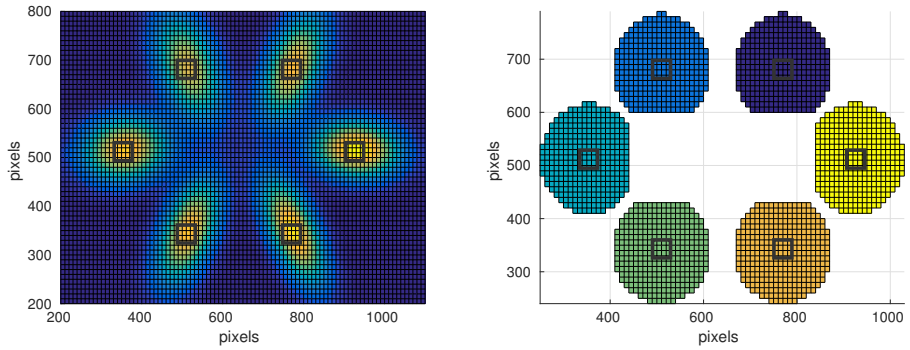


Figure 50: Left: Histogram of gazes during planning across all participants. From this we estimated covariances for each of the 6 states shown as black boxes here. We clustered gazes that fell into one of the colored regions on the right plot and then computed the variances for each region independently.

$p(E_1, ..., E_T|X_1 = x_1, X_T = x_M, \mathbf{\Sigma}_e)$, where it is important that we condition on the fact that the game state sequence must start in x_1 and end in x_M. If we did not do so, all state sequences of which $[x_1, ..., x_M]$ is a sub-sequence would return the same probability, in case the eye data only supports the states in the sub-sequence.

So we modify the forward pass of the standard forward-backward algorithm, requiring that the HMM model starts in state x_1 and ends in state x_M. Let

$$p(E_1, ..., E_T|X_1 = x_1, X_T = x_M, \mathbf{\Sigma}_e) = \frac{p(E_1, ..., E_T, X_T = x_M|X_1 = x_1, \mathbf{\Sigma}_e)}{p(X_T = x_M|X_1 = x_1)}, \tag{100}$$

where $p(X_T = x_M|X_1 = x_1)$ is the probability of ending in x_M when starting in x_1 according to the transition probabilities only, and we dropped trial and subject indices for clarity of exposition. This allows us to calculate

$$\begin{aligned} &p(E_1, ..., E_T, X_T = x_M|X_1 = x_1, \mathbf{\Sigma}_e) \\ &= p(E_1|X_1 = x_1, \mathbf{\Sigma}_e)p(X_T = x_M|X_{T-1})p(E_T|X_T = x_M, \mathbf{\Sigma}_e)p(X_1, ..., X_T|E_1, ..., E_T), \end{aligned} \tag{101}$$

where the last term, $p(X_1, ..., X_T|E_1, ..., E_T)$, is computed according to the forward-backward algorithm,

$$p(X_1, ..., X_T|E_1, ..., E_T) = \prod_{t=2}^{T-1} p(X_t|X_{t-1})p(E_t|X_t, \mathbf{\Sigma}_e). \tag{102}$$

6.5.3 Inference

The aim is to infer posterior distributions over evaluation trees such as to characterise the process. To do so, we first have to infer the prior parameters

$\boldsymbol{\zeta} = \{\boldsymbol{\zeta}_\theta, \boldsymbol{\zeta}_\alpha\}$. We do this through ML estimation by maximizing the log-likelihood of $\boldsymbol{\zeta}$,

$$\begin{aligned}\hat{\boldsymbol{\zeta}} &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \log p(\tilde{\boldsymbol{e}}, \tilde{\boldsymbol{a}}|\boldsymbol{\zeta}) \\ &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \log \prod_{n=1}^{N} p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\zeta}) \\ &= \underset{\boldsymbol{\zeta}}{\operatorname{argmax}} \log \prod_{n=1}^{N} \int_{\boldsymbol{\xi}_n} p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n) p(\boldsymbol{\xi}_n|\boldsymbol{\zeta}), \end{aligned} \tag{103}$$

where $\boldsymbol{\xi}_n = \{\boldsymbol{\alpha}_n, \boldsymbol{\theta}_n\}$. Here, we have assumed that participants are independent, and thus the corresponding probabilities factorize. In order to get the ML estimate of $\boldsymbol{\zeta}$, we need to integrate out $\boldsymbol{\xi}_n$. The prior $p(\boldsymbol{\xi}_n|\boldsymbol{\zeta})$ over $\boldsymbol{\xi}_n$ is assumed to be Gaussian and mainly serves to regularize the inference. It leads to increased convergence behavior of the EM algorithm, because it decreases the danger to get trapped in local optima far away from the group mean.

Similarly to the behavioral models, the EM algorithm can be used to compute the solution. In the i-th E step, we use a Laplacian approximation for the conditional distribution over subject-specific parameters given the data, $p(\boldsymbol{\xi}|\tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{e}}_n) = \mathcal{N}(\hat{\boldsymbol{\xi}}_n^{(i)}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}_n}^{(i)})$, i.e., we assume that they are normally distributed with modes

$$\hat{\boldsymbol{\xi}}_n^{(i)} = \underset{\boldsymbol{\xi}_n}{\operatorname{argmax}}\, p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n) p(\boldsymbol{\xi}_n|\hat{\boldsymbol{\zeta}}^{(i-1)}). \tag{104}$$

The variances are approximated by the second moments $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}_n}^{(i)}$ around $\hat{\boldsymbol{\xi}}_n^{(i)}$. In the i-th M step, the priors $p(\boldsymbol{\xi}_n|\boldsymbol{\zeta})$ are updated. Assuming $p(\boldsymbol{\xi}_n|\boldsymbol{\zeta}) = \mathcal{N}(\boldsymbol{\mu}_\xi, \boldsymbol{\Sigma}_\xi)$ the update becomes

$$\hat{\boldsymbol{\mu}}_\xi^{(i)} = N^{-1} \sum_n \hat{\boldsymbol{\xi}}_n^{(i)} \tag{105}$$

$$\hat{\boldsymbol{\Sigma}}_\xi^{(i)} = N^{-1} \sum_n \left(\hat{\boldsymbol{\xi}}_n^{(i)} (\hat{\boldsymbol{\xi}}_n^{(i)})^\top + \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}_n} \right) - \boldsymbol{\mu}_\xi^{(i)} (\boldsymbol{\mu}_\xi^{(i)})^\top. \tag{106}$$

The main obstacle for implementation is that in order to evaluate $p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n)$ in equation 104 we have to sum over all latent evaluation processes (search trees) $t_{n,k} \in \mathcal{T}_{n,k}$:

$$\log p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n) = \log \sum_{k=1}^{K} \sum_{t_{n,k} \in \mathcal{T}_{n,k}} p(\boldsymbol{e}_{n,k}|t_{n,k}, \boldsymbol{\epsilon}_n) p(\boldsymbol{a}_{n,k}|t_{n,k}, \boldsymbol{\alpha}_n) p(t_{n,k}|\boldsymbol{\theta}_n). \tag{107}$$

where K indicates the number of trials. We note that the only free parameters that our formulation of $p(\boldsymbol{e}_{n,k}|t_{n,k}, \boldsymbol{\epsilon}_n)$ contains are the covariances of the Gaussians in the noise model.

Since the number of possible search trees per trial $|\mathcal{T}_{n,k}|$ is huge (see Figure 51), it is infeasible to calculate this expression exactly. We investigate two different approaches to approximate $\log p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n)$ in each E-step. In this section, we

describe an approximate EM sampling scheme (inside the outer EM algorithm that computes subject-specific parameters) that calculates the expectation of the log-likelihood across trials. In section 6.6, we develop a trial-based inference approach based on Markov Chain Monte Carlo, with a custom proposal distribution $q(t'_{n,k}|t_{n,k})$ over search trees.

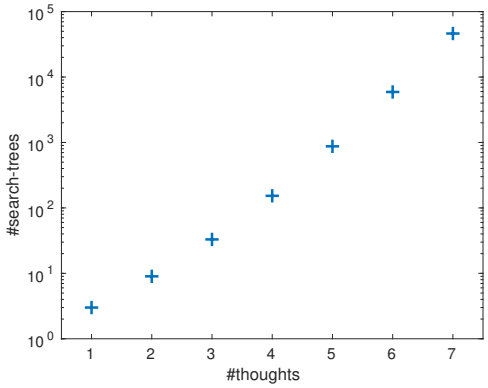


Figure 51: Number of possible search trees vs. maximum length of evaluation sequence for the case where decision-sequences (so the output of the evaluation process) are of length 6. Note the logarithmic scaling. The number of possible evaluation sequences / search trees increases over-exponentially with their maximum length. It was infeasible to generate all search-tree for evaluation sequence lengths of 8 or higher, however, as Figure 47 suggests, we need to set the maximum number of evaluations to 15 or higher. Furthermore, in this simulation we did not allow checking moves, otherwise the number of search trees would have been even much higher.

6.5.4 EM importance sampling

As calculation of the log-likelihood $\log p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n)$ via equation 107 is infeasible, we develop here an approximate EM procedure with the hope that this approximation will allow us to infer the parameters precisely enough so that we can compute the posteriors.

We already described the exact EM algorithm in section 6.4.1.3. Here, we wish to approximate $\log p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n|\boldsymbol{\xi}_n)$. Thus, the i-th E-step consists of maximizing $\mathcal{L}$ with respect to $q(\tilde{\boldsymbol{t}}_n)$ while holding $\boldsymbol{\xi}_n^{(i)}$ fixed. In the subsequent M-step, $q(\tilde{\boldsymbol{t}}_n)$ is held fixed and we compute $\boldsymbol{\xi}_n^{(i+1)} = \text{argmax}_{\boldsymbol{\xi}_n} \mathcal{L}$, where $\mathcal{L}$ takes the form

$$\mathcal{L}(q(\tilde{\boldsymbol{t}}_n), \boldsymbol{\xi}_n) = \mathbb{E}_{p(\tilde{\boldsymbol{t}}_n|\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n, \boldsymbol{\xi}_n^{(i)})} \left[\log p(\tilde{\boldsymbol{e}}_n, \tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{t}}_n|\boldsymbol{\xi}_n)\right] - C, \tag{108}$$

where the second term is constant with respect to $\boldsymbol{\xi}_n$. We cannot analytically solve this expression exactly, as the number of trees is too big, so we approximate the expectation through sampling from the prior $p(\tilde{\boldsymbol{t}}_n|\boldsymbol{\xi}_n^{(i)})$ instead of the posterior

(which is unavailable to us). This results in an approximative EM scheme

$$
\begin{aligned}
&\mathbb{E}_{p(\tilde{\boldsymbol{t}}_n|\tilde{\boldsymbol{e}}_n,\tilde{\boldsymbol{a}}_n,\boldsymbol{\xi}_n^{(i)})}\left[\log p(\tilde{\boldsymbol{e}}_n,\tilde{\boldsymbol{a}}_n,\tilde{\boldsymbol{t}}_n|\boldsymbol{\xi}_n)\right] \\
&= \sum_k \sum_{t_{n,k}\in\mathcal{T}_{n,k}} p(t_{n,k}|\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k},\boldsymbol{\xi}_n^{(i)}) \log p(\boldsymbol{e}_{n,k},\boldsymbol{a}_{n,k},t_{n,k}|\boldsymbol{\xi}_n) \\
&\approx \sum_k J^{-1} \sum_j \log p(\boldsymbol{e}_{n,k},\boldsymbol{a}_{n,k},t_{n,k}^{(j)}|\boldsymbol{\xi}_n), \qquad\qquad t_{n,k}^{(j)} \sim p(t_{n,k}|\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k},\boldsymbol{\xi}_n^{(i)}) \\
&\approx \sum_k J^{-1} \sum_j \frac{p(t_{n,k}^{(j)}|\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k},\boldsymbol{\xi}_n^{(i)})}{p(t_{n,k}^{(j)}|\boldsymbol{\xi}_n^{(i)})} \log p(\boldsymbol{e}_{n,k},\boldsymbol{a}_{n,k},t_{n,k}^{(j)}|\boldsymbol{\xi}_n), \quad t_{n,k}^{(j)} \sim p(t_{n,k}|\boldsymbol{\xi}_n^{(i)}) \\
&= \sum_k J^{-1} \sum_j \frac{p(\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k}|t_{n,k}^{(j)},\boldsymbol{\xi}_n^{(i)})}{p(\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k}|\boldsymbol{\xi}_n^{(i)})} \log p(\boldsymbol{e}_{n,k},\boldsymbol{a}_{n,k},t_{n,k}^{(j)}|\boldsymbol{\xi}_n), \quad t_{n,k}^{(j)} \sim p(t_{n,k}|\boldsymbol{\xi}_n^{(i)}) \\
&\approx \sum_k \sum_j \frac{p(\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k}|t_{n,k}^{(j)},\boldsymbol{\xi}_n^{(i)})}{\sum_l p(\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k}|t_{n,k}^{(l)},\boldsymbol{\xi}_n^{(i)})} \log p(\boldsymbol{e}_{n,k},\boldsymbol{a}_{n,k},t_{n,k}^{(j)}|\boldsymbol{\xi}_n), \quad t_{n,k}^{(j)} \sim p(t_{n,k}|\boldsymbol{\xi}_n^{(i)}).
\end{aligned}
\tag{109}
$$

In the last line we used the approximation $p(\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k}|\boldsymbol{\xi}_n^{(i)}) = \sum_l p(\boldsymbol{a}_{n,k},\boldsymbol{e}_{n,k}|t_{n,k}^{(l)})$, where j,l enumerate all samples drawn from the prior in the current EM iteration.

6.5.5 Results

6.5.5.1 Mixture model As a first step, we tested the EM importance sampling procedure as defined in equation 109 on a Gaussian mixture model (Bishop, 2006). We were particularly interested in examining how it might scale. Let

$$
p(\boldsymbol{z_n}) = \prod_k \pi_k^{z_{n,k}}, \qquad p(x_n|\boldsymbol{z_n}) = \prod_k \mathcal{N}(x_n|\mu_k,\sigma_k)^{z_{n,k}}, \tag{110}
$$

where K is the number of mixture components N the number of samples, and $\boldsymbol{\theta} = [\pi_1,\mu_1,\sigma_1,...,\pi_K,\mu_K,\sigma_K]$ the parameters. z_n are the hidden variables denoting the mixture component in a 1-in-K representation (a vector in which exactly one value is one, and all other values are zero), such that $\mathcal{N}(x_n|\mu_k,\sigma_k)^{z_{n,k}}$ and $\pi_{n,k}^{z_{n,k}}$ are equal to 1 for all $z_{n,k}$ but one. Following Bishop (2006), The objective function includes an additional term with a Lagrange multiplier to ensure that the component weights π_k of the GMM sum up to one,

$$
\begin{aligned}
\mathcal{L}(\theta) &= \operatorname*{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{z}|x,\boldsymbol{\theta})}[\log p(x,\boldsymbol{z}|\boldsymbol{\theta})] \qquad \text{s.t.} \qquad \sum_k \pi_k = 1 \\
&\approx \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_n \sum_m \log p(x_n,\boldsymbol{z}_n^{(m)}|\boldsymbol{\theta}) w_n^{(m)} + \lambda\left(\sum_k \pi_k - 1\right), \quad \boldsymbol{z}_n^{(m)} \sim p(\boldsymbol{z}|x,\boldsymbol{\theta}),
\end{aligned}
$$

where one $\boldsymbol{z}_n^{(m)}$ is sampled for each component from the posterior $p(\boldsymbol{z}|x, \boldsymbol{\theta})$ and the importance weights $w_n^{(m)}$ are defined as

$$w_n^{(m)} = \frac{p(\boldsymbol{x}_n|z_n^{(m)}, \boldsymbol{\theta})}{\sum_l p(\boldsymbol{x}_n|z_n^{(l)}, \boldsymbol{\theta})}.$$

This can be solved analytically, yielding the update equations for the M-step,

$$\begin{aligned}
\mu_k &= \frac{\sum_{n,m} z_{n,k}^{(m)} w_n^{(m)} x_n}{\sum_{n,m} z_{n,k}^{(m)} w_n^{(m)}} \\
\sigma_k^2 &= \frac{\sum_{n,m} z_{n,k}^{(m)} w_n^{(m)} (x_n - \mu_k)^2}{\sum_{n,m} z_{n,k}^{(m)} w_n^{(m)}} \\
\pi_k &\propto \sum_{n,m} z_{n,k}^{(m)} w_n^{(m)}.
\end{aligned}$$

We generated $N = 100$ samples from mixtures with 2, 3 and 4 components, computed the exact EM solution and then compared to our approximate solution using $J \in \{10, 20, 40\}$ importance samples. We then repeated all simulations with $N = 1000$ samples; the parameters were set as follows: for the 2 component mixture we chose $\boldsymbol{\mu} = [-5, 5], \boldsymbol{\sigma} = [2, 1], \boldsymbol{\pi} = [0.5, 0.5]$, for the 3 component mixture we chose $\boldsymbol{\mu} = [-5, 0, 5], \boldsymbol{\sigma} = [1.2, 1.5, 1.2], \boldsymbol{\pi} = [0.4, 0.3, .3]$, and for the 4 component mixture we chose $\boldsymbol{\mu} = [-3, -1, 1, 3], \boldsymbol{\sigma} = [0.6, 0.9, 0.6, 0.9], \boldsymbol{\pi} = [0.2, 0.3, 0.3, 0.2]$.

The results are shown in Figure 52. For relatively easy problems (left column, $K = 2$), where the two mixture components separate almost fully, approximate EM sampling works just as well as the exact EM algorithm, even for relatively low number of importance samples ($J = 10$). In more interesting problems (middle column, end right columns), where the mixture components are not as easily identifiable, importance sampling performance scales with the number of importance samples, and for $J = 40$ we achieve performance similar to exact EM, providing a working proof-of-concept of our approximation.

6.5.5.2 Surrogate data Next, we examined whether parameters could be recovered from data that was sampled from the generative model defined in section 6.5.2, i.e. when the truth is known. In Appendix B we show some examples of which kind of search trees are generated by the model.

We performed 100 simulations each for $K \in \{10, 20, 40, 80, 160, 320\}$ number of trials and constrained the problem to decision-sequences of length 3 and a maximum number of internal evaluations equal to 6. Further, we generated 20 gazes per trial. For the E-step approximations, we used $J = 500$ importance samples. Parameter values were fixed to $\log(\theta) = -4.0$, $\log(\beta) = -4.0$ and $\rho = -10$. Since we expected the evaluation sequence lengths to exceed 6 often in the real data, this was a much simplified inference; however, this setup allowed us to compute the exact EM solutions for comparison. The results are shown

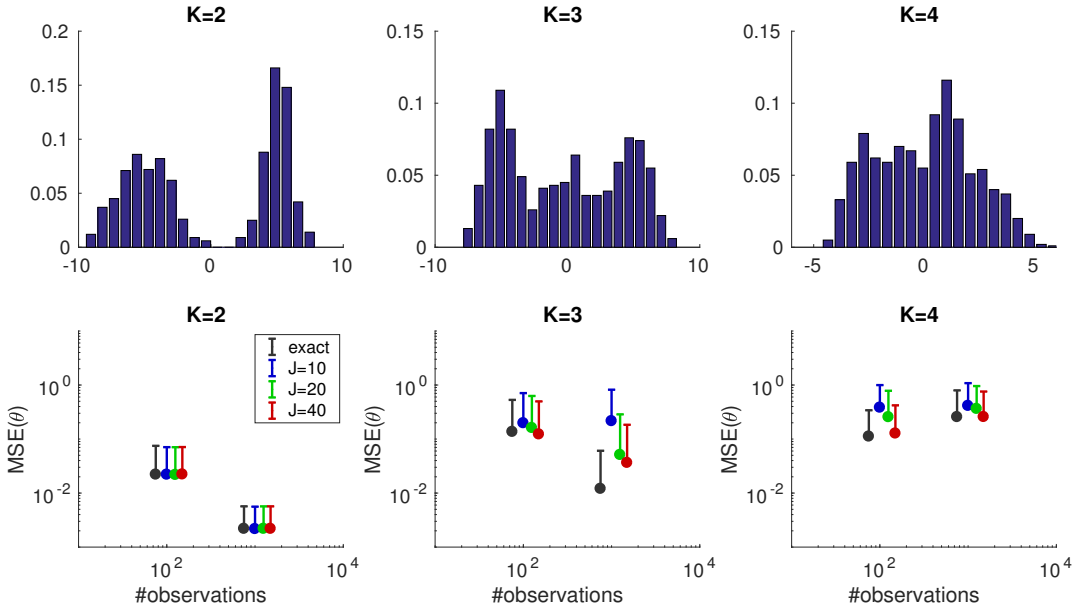


Figure 52: Results of approximate EM sampling for Gaussian Mixture Models. The top row shows examples of generated data for $N = 1000$, and the bottom row shows the results. We chose mixture models with 2 (left plot), 3 (middle plot) and 4 (right plot) components. The exact EM solution results are shown in black, approximate solutions for $J = \{10, 20, 40\}$ importance samples are shown in blue, green and red, respectively.

in Figure 53. The EM importance solutions have higher variance, and very occasionally yield solutions that differ very much from the correct values. The high mean error in the left plot for $K = 160$ trials is mainly caused by 3 of the simulations which were not able to recover the correct parameter even remotely. In general, however, the approximate approach seems to work well if the number of trials is large enough.

However, for the purpose of inferring individual differences the RMSE values of even the exact EM solution are unacceptable. For example, we know from previous experiments (see Huys et al. (2012)), that most participants have β-values of between 0.03 and 0.05. But Figure 53 clearly shows that even when using many more trials than we could use in our experimental setup, RMSE(β) ≈ 0.01. This is too big for individual differences to be inferred successfully.

Next, we designed a set of simulations in order to investigate why errors even for the exact EM solution were higher than we had anticipated. We fixed parameters β and ρ to their true values and only estimated θ, simplifying the inference task even further. This allowed us to compute the log-likelihood values $\log(p(\tilde{\boldsymbol{a}}_n))$ (that are being approximated by EM) on an evenly spaced grid of values of θ for 50 randomly generated data sets for each $K \in \{100, 200, 300, 400\}$, and computed the maximum to obtain parameter estimates. As there was only one free parameter, not only was it feasible to perform grid search, but we could also rule out the possibility that gradient descent would get stuck in local minima. For comparison, we also calculated the maximum of $\log(p(\tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{t}}_n))$, which is sometimes called the complete data log-likelihood. The result is the best solution that could have been found if the internal evaluation sequences had been observed. As before, we opted for an action sequence length of 3 and a maximum evaluation sequence length of 6 (but also compared to length 3).

As the results in Figure 54 indicate, the RMSE of θ obtained from maximizing $\log(p(\tilde{\boldsymbol{a}}_n, \tilde{\boldsymbol{t}}_n))$ was small (0.0040 for evaluation sequence length 3, 0.0028 for length 6) even for moderately small number of trials $K = 100$ - this was close to the lower bound achievable due to the spacing of the grid points. However, when utilizing the log-likelihood $\log(p(\tilde{\boldsymbol{a}}_n))$ instead, a comparably small RMSE was only achievable for $K = 400$.

6.5.6 Discussion

Unfortunately, the simulations carried out in this section indicate that we cannot estimate parameters accurately enough under the approximate EM scheme, even for the much simplified setting of the simulation in comparison with the real data. For realistic trial sequence lengths ($K < 200$), both approximate and exact EM solutions show high errors. This is because the log-likelihood does not carry enough information in order to infer the parameters more accurately, even if we could optimize it directly instead of utilizing EM.

Indeed, we showed how tricky the inference problem really is here: Even if there is just one free parameter and we optimize the complete data log-likelihood $\log(p(a, t))$ directly, more than 200 trials are necessary to accurately and robustly

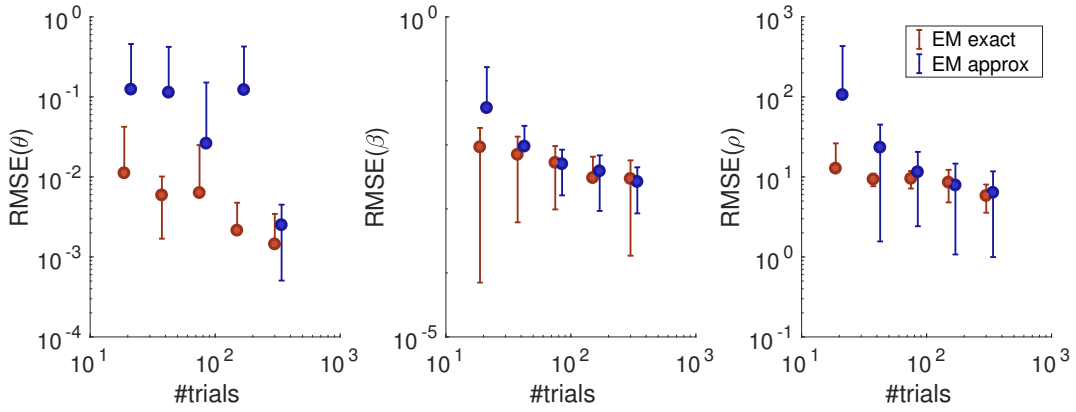


Figure 53: Results shown indicate mean and standard deviation of 100 simulations, in each of which parameters were inferred from a randomly generated data set. Exact EM results are shown in blue, EM importance results in brown. Only depth 3 trials, maximum number of evaluations 6, 20 eye fixations per trial, 500 importance samples in each E-Step. The plots from left to right show root mean square errors for the internal evaluation sensitivity parameter θ, the action sensitivity parameter β and the reward replacement parameter ρ (which substitutes the reward for actions that were not thought about).

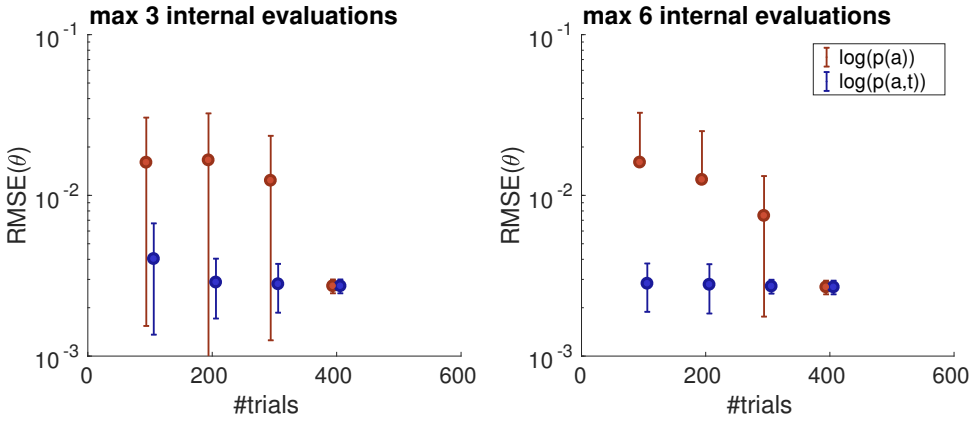


Figure 54: Results from a simplified set of simulations with only one free parameter, the internal evaluation sensitivity parameter θ. RMSE obtained from optimization of $\log(p(a,t))$ are shown in blue, log-likelihood results in brown. The left plots shows results for a maximum of 3 internal evaluations per trial; the right plot shows results for a maximum of 6 internal evaluations.

estimate the parameter. As we could not collect this amount of trials from participants due to time constraints, and do not have access to the internal evaluation sequences directly, the proposed approach must be discarded. We have to conclude that for our computational model no strategy that directly infers parameters by approximating the log-likelihood across trials can be accurate with the given amount of data.

6.6 Trial-based inference with Markov Chain Monte Carlo

Clearly, the approximate sampling scheme that we developed in the previous section is not accurate enough to infer model parameters with sufficient precision for the given amount of trials per participant. Even worse, since even directly optimizing the log-likelihood seems to not be accurate enough, we will have to resort to a completely different inference scheme. In this section we approximate the sum over all search trees in equation 107 *individually for each trial k* through Markov Chain Monte Carlo sampling.

6.6.1 Markov Chain Monte Carlo

The idea of Monte Carlo simulation is to obtain a set of samples $x^{(n)}$, where the samples are distributed according to some $p(x)$. This allows the expectation $\mathbb{E}_{p(x)}[f(x)]$ to be approximated,

$$\begin{aligned}\mathbb{E}_{q(x)}[f(x)] &= \int_x f(x)p(x)dx \\ &\approx I^{-1}\sum_{i=1}^{I} f(x^{(i)}), \quad x^{(i)} \sim p(x),\end{aligned} \tag{111}$$

where $x^{(i)}$ denotes the i-th sample and I is the total number of samples.

Many different Monte Carlo sampling schemes exist. One example is Markov Chain Monte Carlo (MCMC), which proceeds by constructing a first-order Markov Chain $\{x_1, ..., x_I\}$ whose invariant distribution is p. In somewhat loose terms, p is an invariant distribution if once p is reached it persists forever (Barber, 2011). A first-order Markov Chain, which we have already encountered as the sequence of states in an HMM in section 6.5.2.3, is a series of random variables with the property that each variable $x^{(i)}$ is independent of all the others if conditioned on the preceding variable $x^{(i-1)}$. Thus, it is completely characterized by the transition probabilities $q(x'|x^{(i)})$. In MCMC, at each iteration a new state x' is drawn from $q(x'|x^{(i)})$ and accepted as the chain's state for the next iteration with a certain probability called the proposal acceptance probability.

The simplest approach to designing a Markov chain with invariant distribution p is to make sure that the transition probabilities satisfy the *detailed balance* property,

$$p(x^{(i)})q(x^{(i)}|x') = p(x')q(x'|x^{(i)}), \tag{112}$$

where $x^{(i)}$ is the chains's state at iteration i and x' a proposal for the next iteration. In words, the probability of the current state of the chain times the probability of changing from the proposed state to the current state must be equal to the probability of the proposed state times the probability of changing to the proposed state from the current state.

If the chain is *homogenous*, i..e the transition probabilities are the same for all i, it can be shown that under weak conditions the invariant distribution is indeed p. The most popular MCMC sampling algorithm is arguably the Metropolis-Hastings (MH) algorithm (Hastings, 1970), where the acceptance probability is a direct result of the maintenance of detailed balance: Suppose that $q(x'|x^{(i)})p(x^{(i)}) > q(x^{(i)}|x')p(x')$, then there is an $a(x^{(i)}, x')$, for which $a(x^{(i)}, x')q(x'|x^{(i)})p(x^{(i)}) = q(x^{(i)}|x')p(x')$. Solving for a yields the proposal acceptance probability

$$a(x^{(i)}, x') = \min\left(1, \frac{p(x')q(x^{(i)}|x')}{p(x_i)q(x'|x^{(i)})}\right) = \min(1, \alpha). \tag{113}$$

If $\alpha \geq 1$, the proposed state x' is automatically accepted; otherwise the state is accepted with probability α. If the proposed state is rejected, then $x^{(i+1)} = x^{(i)}$.

6.6.2 Constructing the transition distribution

With respect to equation 107, we wish to draw samples from $p(t_{n,k}|\boldsymbol{\theta}_n)$ in order to evaluate $p(\boldsymbol{e}_{n,k}, \boldsymbol{a}_{n,k}|\boldsymbol{\xi}_n)$. Thus, the MH acceptance ratio becomes

$$\alpha_{(t_{n,k}^{(i)}) \to (t'_{n,k})} = \min\left(1, \frac{p(t'_{n,k}|\boldsymbol{\theta}_n)}{p(t_{n,k}^{(i)}|\boldsymbol{\theta}_n)} \cdot \frac{q(t_{n,k}^{(i)}|t'_{n,k})}{q(t'_{n,k}|t_{n,k}^{(i)})}\right). \tag{114}$$

The result is a set of samples $\{t_{n,k}^{(1)}, ..., t_{n,k}^{(I)}\}$, which can be used to approximate the likelihood,

$$\begin{aligned} p(\boldsymbol{e}_{n,k}, \boldsymbol{a}_{n,k}|\boldsymbol{\xi}_n) &= \sum_{t_{n,k} \in \mathcal{T}_{n,k}} p(\boldsymbol{e}_{n,k}|t_{n,k}, \boldsymbol{\epsilon}_n) p(\boldsymbol{a}_{n,k}|t_{n,k}, \boldsymbol{\alpha}_n) p(t_{n,k}|\boldsymbol{\theta}_n) \\ &\approx I^{-1} \sum_{i=1}^{I} p(\boldsymbol{e}_{n,k}|t_{n,k}^{(i)}, \boldsymbol{\epsilon}_n) p(\boldsymbol{a}_{n,k}|t_{n,k}^{(i)}, \boldsymbol{\alpha}_n), \qquad t_{n,k}^{(i)} \sim p(t_{n,k}|\boldsymbol{\theta}_n). \end{aligned} \tag{115}$$

The remaining question is how to specify the transition distribution $q(t'|t)$, such that detailed balance holds. A relatively simple solution is to propose two opposing types of modifications of a search-tree, such that $q(t|t') > 0$ for all t' for which $q(t'|t)$ - in words, we need to ensure that there is a positive probability to transition back from t' to t for any t' that we can reach from t. Here, we assume that all transitions are equiprobable, i.e. $q(t'|t) = c, \sum_{t'} q(t'|t) = 1$.

Checking moves complicate valid modifications of search trees. We can think of any checking sequence of backwards traversals starting at the current node

of the search tree; from here on, we will refer to it as the 'virtual trace' of that node. Further, we also need to consider the sequence of evaluations, where the corresponding sequence of nodes is given by the labels of the edges. Keeping that in mind, we introduce the 'birth' move, and its opposing move, the 'death' move, as ingredients of $q(t|t')$.

A 'birth' move is any move that appends a child to any node that is not already fully expanded. The new child node can be inserted into the evaluation sequence at any position after its parent; it is initialized with an empty virtual trace sequence. To maintain detailed balance, the 'death' must be defined as the exact reverse of the birth move. It can remove any leaf with an empty trace, but only if there are no traces of other nodes that contain references to this leaf node.

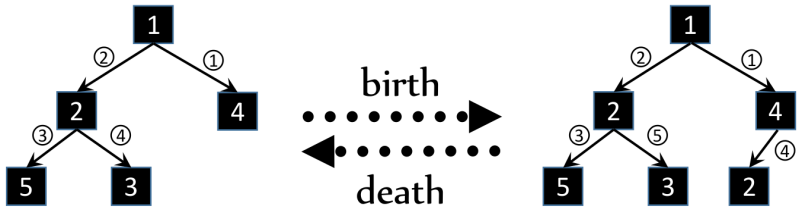


Figure 55: Birth and death move example. The virtual traces of all nodes are empty, meaning that none of the transitions was traversed backward.

There are some obvious limitations to simply using birth and death moves: first of all, no search trees will ever be generated that contain virtual traces. Second, such tree modifications are very local. If all trees that are reachable via either birth or death have a lower probability than the current tree, than sampling is likely to at least temporarily get stuck in a local minimum. As a result, exploration of the search space might be slow. The first limitation can be dealt with by introducing 'checking moves' which add or remove a 'checking evaluation' to a node's virtual trace, thus enabling traversal of transitions multiple times. An example of such a move is shown in Figure 56.

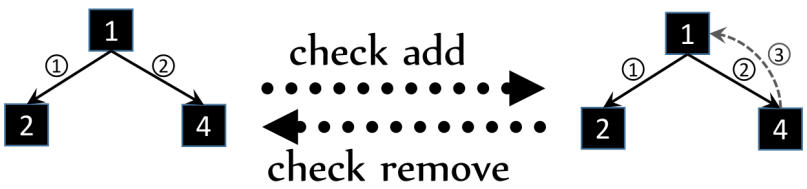


Figure 56: Example of a check / uncheck move pair.

However, the main problem really is the complex neighborhood structure of the search space. Figure 57 shows the differences in log-probability of search

trees vs. their distance in terms of number of birth/death moves. These are shown for all 617 search trees with starting state 4, action sequence length 3, a maximum evaluation sequence length of 5, and $\theta = \exp(-3)$. We did not allow virtual traces / checking moves in order to be able to evaluate all trees. Clearly, the distance between trees correlates only weakly with their log-probability difference. Just adding or removing one single node can change the tree probability by up to a factor of exp(10). This difference is so big that naive MH sampling would practically never accept a move that proposes a transition to a tree that is so much less likely. In order to be able to better explore the whole space, we implemented two modifications to naive MH sampling as we have described it above: First, we substituted MH sampling by Stochastic Approximation sampling, which penalizes trees that have already been visited and as such facilitates exploration of new regions in the search space. At the same time, it samples in log-probability space, such that big differences in probability can be handled easily. Second, we made the transition distribution more flexible, by adding moves that implement bigger changes in the search trees.

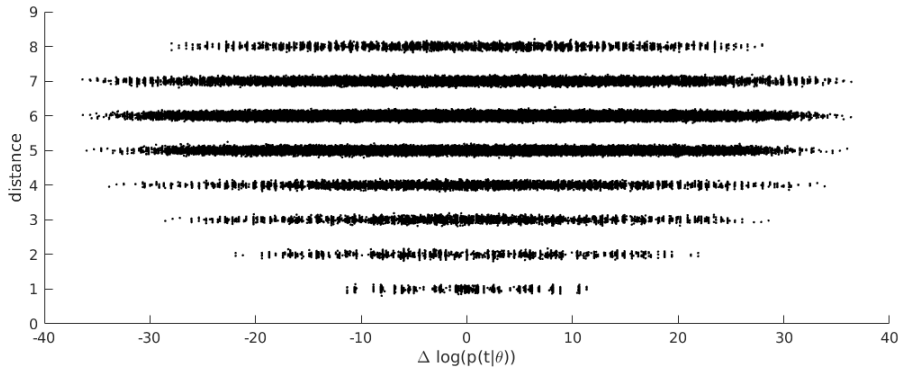


Figure 57: Difference in log-probability of search trees vs. distance between trees in number of birth/death moves for all 617 search trees with starting state 4, action sequence length 3, a maximum evaluation sequence length of 5, $\theta = \exp(-3)$, and virtual traces / checking moves disabled. We added a small amount of noise in the visualization so that results occupying the exact same point become visible. Even for such an 'easy' problem, we can observe that simply adding or removing one node from the tree (distance 1) can change the probability by a factor of exp(10). On the other hand, while making more changes to a tree on average results in higher differences in log-probability, this is individually often not the case. The Pearson correlation coefficient between the distance and $|\Delta \log p(t|\theta)|$ is 0.09.

6.6.3 Stochastic Approximation Monte Carlo

Stochastic Approximation Monte Carlo (SAMC; see (Liang et al., 2007)) encourages exploration of under-sampled parts of the probability space while at the same time guaranteeing convergence. It was developed to estimate the spectral

density $g(u) = \#\{x : U(x) = u\}$, i.e. the number of states x with equal energy for a discrete, finite set of energy values $\{u_1, ..., u_J\}$, where $U(x) = -\log p(x)$ is the energy function. To adapt SAMC to our setting, we define exactly one energy value for each search tree, $u_j = t_j \; \forall j \in [1, |\mathcal{T}|]$. The modified acceptance criterion can be written as

$$\alpha_{(t_{n,k}^{(i)}) \to (t'_{n,k})} = \min \left(1, \frac{e^{\kappa_i(t_{n,k}^{(i)})}}{e^{\kappa_i(t'_{n,k})}} \frac{p(t'_{n,k}|\boldsymbol{\theta}_n)}{p(t_{n,k}^{(i)}|\boldsymbol{\theta}_n)} \cdot \frac{q(t_{n,k}^{(i)}|t'_{n,k})}{q(t'_{n,k}|t_{n,k}^{(i)})} \right), \tag{116}$$

where $\kappa_i(t) = \log(\hat{g}_i(t))$ denotes the logarithm of the sampling frequency estimate of t at iteration i. The adjustment $e^{\kappa_i(t_{n,k}^{(i)})}/e^{\kappa_i(t'_{n,k})}$ always works in the reverse direction of the discrepancy between the realized sampling frequency and the desired one, which prevents the system from getting trapped in local minima. The frequency estimates are updated using the rule $\kappa_{i+1}(t^{(i+1)}) = \kappa_i(t^{(i+1)}) + \gamma_{i+1}$, where the sequence of gain factors $\{\gamma_i\}$ can be any positive decreasing sequence satisfying $\sum_{i=1}^{\infty} \gamma_i = \infty$ and $\sum_{i=1}^{\infty} \gamma_i^a < \infty$ for some $a \in (1, 2)$. Any such gain factor guarantees convergence under mild conditions. Liang et al. (2007) choose $\gamma_i = \frac{i_0}{\max(i_0, i)}$ for some $i_0 > 1$, which we use here, too. Since the estimates of the decision-tree probabilities are updated on a logarithmic scale, the algorithm works well in the setting when probability differences are big, as is the case for our search trees.

6.6.4 Additional transition moves

While in principle any search sequence is reachable via MCMC using birth, death, check and uncheck moves, we implemented two more proposal types to allow for bigger changes in the search trees: the swap move and the shuffle move.

6.6.4.1 Swapping subtrees We implement the swap move as its own reverse move; it takes a subtree, i.e. the part of the tree that starts at a given node and includes all its descendants, and changes its parent node (see Figure 58 for an example). It is not at all clear how the subtree to be swapped should be correctly inserted if the new parent already has a child (or subtree), because we are essentially free to choose the ordering of the nodes: While the ordering between nodes of the subtree is given, as is the ordering of all nodes outside of the subtree, we are free to interleave them in any way we choose after the swap. Thus, to make this move accessible, we require all nodes of a subtree to immediately follow each other, in order to be able to swap the subtree; we then essentially swap them as if they were 'glued together', inserting all nodes in the subtree immediately following the root of the subtree. We further require that no virtual traces may traverse into or out of the subtree, i.e. all traces of all nodes in the subtree must be wholly contained within that subtree, and all traces of all nodes that are not in the subtree may not reference any node contained in the subtree.

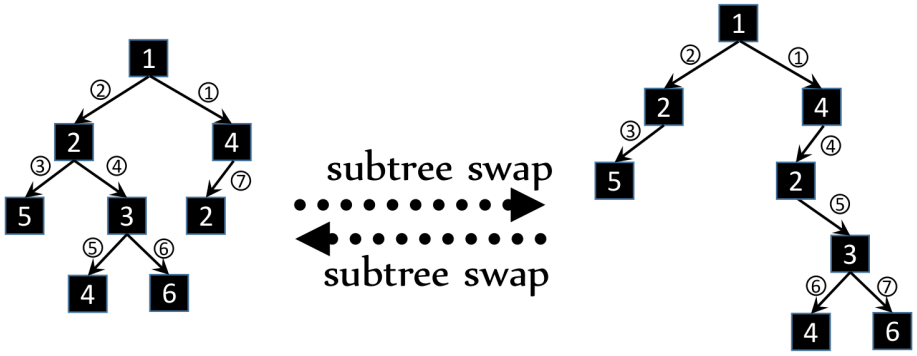


Figure 58: Subtree swap example. We only allow the subtree consisting of the nodes 3, 4 and 6 to be swapped if these nodes follow each other directly in the evaluation sequence (positions 4, 5 and 6).

6.6.4.2 Shuffle proposals The shuffle move allows repositioning of a node in the evaluation sequence anywhere between its parent and its lowest referencing point. This reference can either be the first node in the current node's subtree or any reference contained in any of the traces. An example is shown in Figure 59. The shuffle move is it's own reverse move.

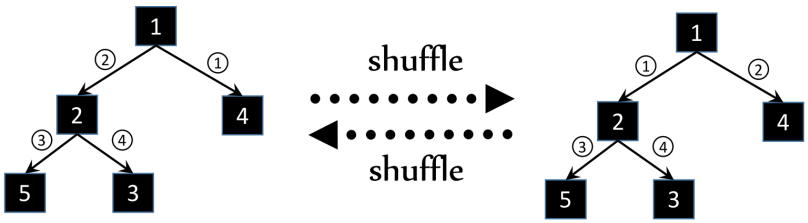


Figure 59: Example of a shuffle move pair: This shuffle move from left to right changes the beginning of the evaluation sequence by exchanging evaluations one and two. If the root had been present in the trace of the node labeled with a 2, for example, the shuffle move would not have been possible.

6.6.5 Results

Each set of simulations consists of 20 trials which we generated with the following parameters: $\rho = -10, \beta = \exp(-3), \theta = \exp(-3)$. We compared four different sampling approaches based on the Monte Carlo techniques that we developed in the previous section: In the first approach ('basic MCMC') we used regular MH sampling with only add/remove proposals moves (and check/uncheck proposals in the simulations that allowed for checking moves). The second approach ('extended MCMC') allowed all proposal moves that we defined in the previous section. The third and fourth approaches ('basic SAMC' and 'extended SAMC') used only add/remove proposal moves, or all proposal moves together

with SAMC sampling. The aim of this comparison was to assess the benefits of non-local search tree changes and using SAMC vs MCMC independently of each other. For each strategy, we ran 20 Markov Chains each of length $10^2, 10^3, 10^4$ and 10^5. Longer chains would have been infeasible in the context of the inference scheme, which utilizes MCMC iteratively within the EM algorithm. For SAMC, we set the speed of convergence parameter t_0 to one tenth of the size of the number of samples in the MCMC chain.

The first set of simulations was designed to be as simple as possible in order to test if our approach works as expected: the action sequence length was set to 3 and evaluation sequences were restricted to a maximum length of 3. We did not generate eyetracking data ($p(e|t) = 1\ \forall t$), and did not allow for checking moves to be generated in the search tree (and consequently did not allow checking proposals during Monte Carlo sampling). The results are shown in Figure 60. All four sampling strategies (MCMC basic, MCMC extended, SAMC basic, SAMC extended) were able to approximate the log-likelihood well for Markov Chains of lengths 10^3 or longer. The best performing method was MCMC extended with 10^5 samples per Markov Chain with a mean absolute error of 0.011 ± 0.014. However, there were only small differences in performance of the four methods.

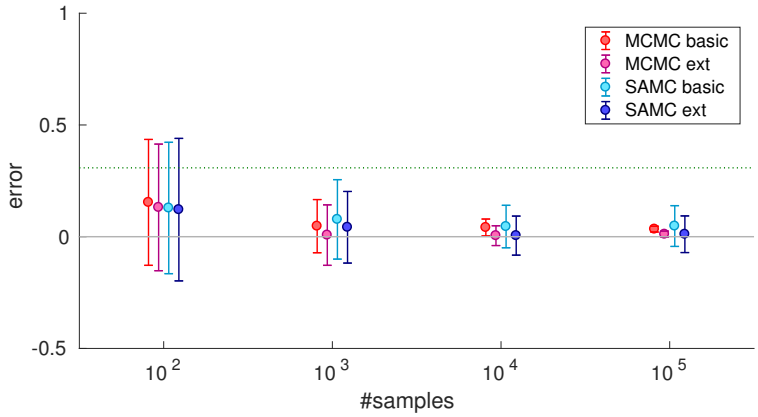


Figure 60: First set of simulations: No eye data, no checking evaluations, action sequence length 3, evaluation sequence length maximum 3. Shown are means and standard deviations of the absolute error in log-likelihood across a total of 400 simulations (20 simulated trials and 20 chains each). The green dotted horizontal line indicates the standard deviation of the 20 correct solutions (computed by summing over all possible search trees), and thus represents a good measure against which the errors can be compared. The method MCMC extended with 10^5 samples performed best (mean absolute error of 0.011 ± 0.014).

The second set of simulations was similar to the first one only that we allowed for longer action / evaluation sequences: the action sequence length was set to 4 and evaluation sequences were restricted to a maximum length of 6. Again, we did not generate eyetracking data ($p(e|t) = 1\ \forall t$), and did not allow for checking moves to be generated in the search tree (and consequently did not allow checking proposals during Monte Carlo sampling). The results are shown in Figure 61.

Again, performance was similar across all four sampling strategies (MCMC basic, MCMC extended, SAMC basic, SAMC extended). The best performing method was 'SAMC basic' with 10^5 samples per Markov Chain with a mean absolute error of 0.116, but a comparatively large standard deviation of 0.200. However, it is interesting to note that 'MCMC basic' achieved a similarly small error of 0.131 ± 0.2750 with only 10^3 MC samples. In comparison to the first set of simulations, the standard deviation of the correct solutions of the 20 generated trials increased from 0.308 to 0.617 by a factor of 2, whereas estimation error increased from 0.011 to 0.116 by roughly a factor of 10. As a consequence, the magnitude of the approximation error is about one fifth of the correct solutions' standard deviation (across trials) - even though the problem is still very much simplified with respect to the real data.

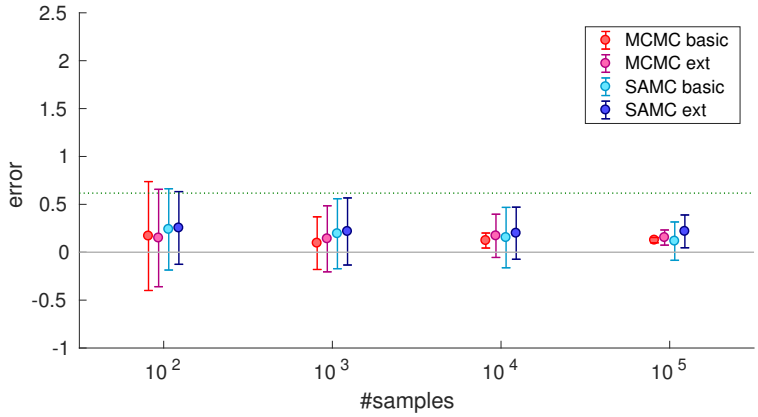


Figure 61: Second set of simulations: No eye data, no checking evaluations, action sequence length 4, evaluation sequence length maximum 6. SAMC basic with 10^5 samples performed best with a mean absolute error of 0.116 ± 0.200. The green dotted horizontal line indicates the standard deviation of the 20 correct solutions.

In the third set of simulations, we added gaze data, but still did not allow for checking moves / proposals. In each trial, we generated 10 gazes from the HMM sequence defined in equation 100. As before, the action sequence length was set to 4 and evaluation sequences were restricted to a maximum length of 6. Here, MCMC basic performed best at 10^4 samples per MCMC chain with an error of 0.287 ± 0.973 (see 62). Increasing the number of samples per chain did not yield any further improvements. However, the ratio of the magnitude of the best approximation error to the correct solutions' standard deviation is about 11 (3.237 / 0.287).

Finally, the fourth set of simulations included checking behavior in the gaze data and correspondingly allowed for checking proposals in the MC proposal distribution. As in the third set of simulations, we generated 10 gazes in each simulated trial, set the action sequence length to 4 and the maximum evaluation sequence length to 6. The results shown in Figure 63 indicate that due to the

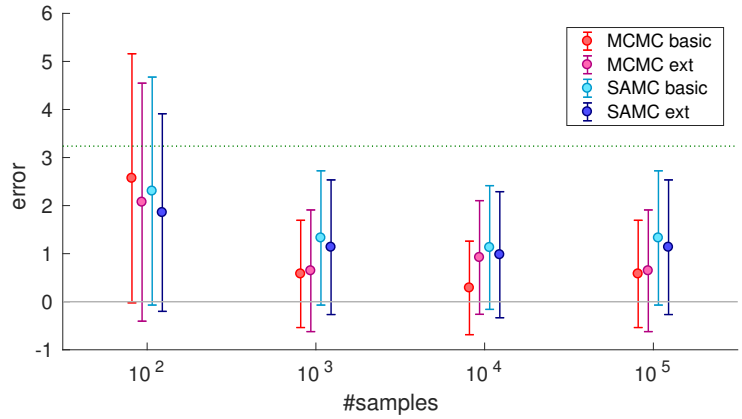


Figure 62: Third set of simulations: Including eye data, but no checking evaluation, action sequence length 4, evaluation sequence length maximum 6. MCMC basic with 10^4 samples performed best with a mean absolute error of 0.287 ± 0.973. The green dotted horizontal line indicates the standard deviation of the 20 correct solutions.

immense increase in complexity of the problem, none of the MC methods can approximate the solutions well.

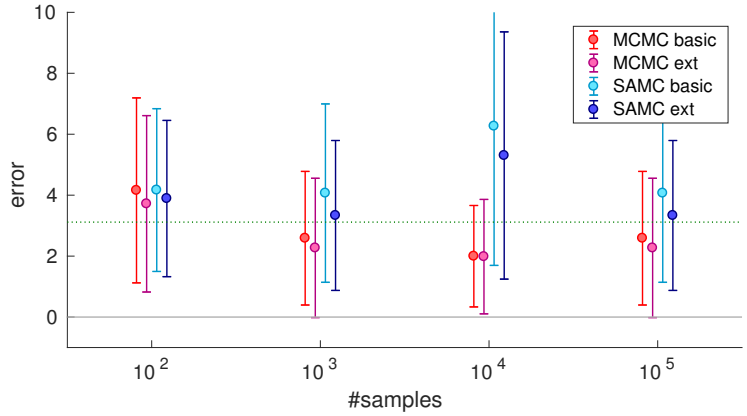


Figure 63: Fourth set of simulations: Including eye data and checking evaluations, action sequence length 4, evaluation sequence length maximum 6. None of the methods is able to approximate the solution well. The green dotted horizontal line indicates the standard deviation of the 20 correct solutions.

6.6.6 Discussion

The first set of simulations showed that all four methods (MCMC basic, MCMC extended, SAMC basic and SAMC extended) work well with a Markov Chain of at least length 1000. In such a simplistic setting, SAMC does not show any advantages over MCMC, as can be expected. Instead its error variance is higher,

because the chains need a longer time to converge. However, the extended versions have a clear advantage over the basic versions (thus, adding subtree swap and shuffle moves causes chains to converge much quicker).

The ratio of the magnitude of the best approximation error to the correct solutions' standard deviation is approximately 5 in the second set of simulations and 11 in the third set, implying that additional (eye gaze) data simplifies the inference problem considerably. The fourth set of simulations showed that as the inference problem gets more difficult (but still easy compared to the real data), the chains converge so slow that even after 10^5 samples, the chains did not converge to the correct solution accurately enough. It would not have been feasible to run even longer chains, due to computer hardware limitations.

We were not able to compute exact solutions for action sequences of length 4 or more (or longer evaluation sequences), but since such problems would be vastly more difficult than the ones we have tested in the previous section, it is clear that approximation through Monte Carlo is infeasible. As a consequence, we were unfortunately not able to apply the method to the real data set.

We believe that the reason for how badly accuracy of inference scales with the problem size is the immensity of the search space; it seems likely that the search space has to be extensively covered by the Markov chain for good convergence (as is the case in the first simulation). Randomly sampling a very small number of trees does not seem to represent the overall search space well. This is especially true when incorporating checking behavior, as this vastly increases the number of possible search trees.

6.7 Conclusion

This work represents - at least to our best knowledge - the first attempt to explicitly infer the metareasoning process in a sequential decision-making task that is too complex to solve fully under the given time constraints. We developed computational models for behavioral data with the aim to better understand the metareasoning process, and sophisticated computational analyses to fit the models to data which we had collected. For representing sequences of internal evaluations we developed the edge-labeled search tree, which includes an ordering of the evaluations given by the edge labels as well as the possibility to include 'checking evaluations' (multiple traversals of edges) as 'virtual traces', which are arrays of node references attached to a node in the search tree. We further developed, implemented and tested two types of computational models as well as three inference methods which employ search trees to explicitly model the metareasoning process.

The first type of model relies purely on behavioral data; it is based on the reinforcement learning approach introduced by Huys et al. (2012), but adds metareasoning weights that scale with the mean probability of having evaluated an action according to one of several tree search strategies. Unfortunately, none of these metareasoning models was able to outperform the basic model that did not include a metareasoning process, suggesting that none of the simple tree-

search strategies that we tested can explain the internal evaluation process across individuals well. From this we conclude that participants' metareasoning in the pruning task is much more complicated than we had initially assumed. One of the likely reasons is that any kind of greedy tree search chooses the best among the set of possible next evaluations at each step; as a consequence, no (sub-)sequences starting with a smaller reward can ever get a higher metareasoning weight than sequences starting with a bigger reward.

Our second type of model incorporates eyetracking data to more directly inform the metareasoning process. We developed a new approximate Expectation Maximization scheme using importance sampling from the search tree prior in order to implement across-trial inference, and demonstrated in simulations that it works well on simple examples. Unfortunately, even exact EM inference does not converge to the correct solution when the difficulty of problem sets approaches the difficulty of the tasks that participants actually performed, because the across-trial log-likelihood simply does not carry enough information to estimate parameters sufficiently well. From this we conclude that for the model that we defined in section 6.5.2 no strategy that directly infers parameters by approximating the across-trial log-likelihood can be accurate.

Finally, we developed an inference approach that works on a trial-by-trial basis, such that for each trial a distribution over search trees is inferred through Monte Carlo sampling in order to approximate the log-likelihood. Our proposal distribution implements both small, local as well as bigger changes to search trees. This was done to cope with local minima in the search space. In addition, we translated the Stochastic Approximation Monte Carlo approach, which samples in log-space, from the field of spectral density estimation.

Unfortunately, neither the naive Metropolis Hasting sampling scheme nor any of our improvements converged to the correct solutions for Markov Chains of reasonable length when running moderately complex simulations. As the real data had an even higher complexity, we could not have successfully achieved accurate parameter estimates. The combination of a rapid increase of search space with the maximum number of possible evaluations and a difficult topology of the search space (where changing a search tree by just one evaluation can dramatically change its probability) results in a very difficult inference problem indeed. More complicated proposal moves that implement even bigger changes in the search trees together with even more advanced MC sampling techniques and much more computational resources might eventually make inference possible. However, it seems unreasonable to hope that any one simple improvement will result in accurate inference.

Metareasoning in the pruning task seems to be a much more complex problem than we had assumed initially. While there is evidence that certain techniques are being employed during metareasoninng, such as pruning and some kind of Pavlovian learning, we were not able to successfully model the internal evaluation process explicitly. Contrary to our expectations, it is unlikely that our test subjects do a simple greedy search in the search tree. A possibility is that the evaluation process is very different to what we anticipated, in the sense that

participants change their strategies throughout the task, or use their memory extensively to mix and match whole sub-sequences of evaluations.

It is probably necessary to design a whole set of experiments in order to gradually gain a finer understanding of the metareasoning process in sequential decision-making. As a first step, an experiment could consist of a much simplified task, that may only contain sequences of 2 or 3 decisions in mazes that change in each trial, such that participants are forced to evaluate state transitions by looking at them. Whatever could be learned from such a task might then be used to inform inference in more complicated tasks.

7 Overall conclusion

In summary, there are many different ways, both conceptually and practically, in which computational perspectives and inferences can be implemented in the context of psychiatry. In the first part of this thesis we presented a data-driven approach, which we called Gaussian Process Trajectory Prediction. It can be used for prediction of longitudinal data based on covariates that were collected at the beginning of an observation period. GPTP captures the intuition that each covariate's influence changes with time, and that the observed overall trajectory results from a linear combination of these individual contributions; explicitly modeling the covariates' influences over time makes the method readily interpretable, in contrast to most modern machine learning approaches. The method implements automatic feature selection through ARD, which works well in the case that the total number of features is sufficiently small (that is, much smaller than the number of observations).

We showed that GPTP handles missing data, censored data and even non-Gaussian data well, and is usable online, such that predictions can be updated as new observations become available, without having to re-estimate the model. We applied GPTP to a large data set of patients suffering from depression. It can explain a large amount of variance, and improve on previously reported results for prediction of remission with an AUC of 0.75. When predicting relapse, which has not been done before on these data to our knowledge, we were able to achieve an even higher AUC of 0.82. No sparse model solely relying on questionnaire data could be found, however; additional regressors from other data sources, such as brain imaging, or behavioral computer tasks, might help in providing more sparse solutions.

In the second part we presented a theory-driven approach based on a mechanistic perspective. Computational models based on this approach and their corresponding quantitative characterization could potentially enrich traditional psychiatric classification. Here, we developed models of the internal evaluations during sequential decision-making, aimed to improve our working hypothesis of how the metareasoning process unfolds. This is important in the present context, because pruning, a metareasoning strategy, was shown to correlate with sub-clinical scores of depression. As part of a study that - among others - employs a metareasoning task to extract parameters of individual subjects, this could be helpful for improving prediction performance of GPTP.

For this purpose, we developed a suite of models, each of which exactly specified a way in which the metareasoning process may unfold. However, the way individuals solve the metareasoning problem turned out to be much more complex than we had anticipated: we showed that none of the tested greedy forward tree-search strategies is able to explain the data well. Due to the immense number of possibilities, it can not be complete ruled out that participants do use a form of greedy tree-search. For example, they could employ depth-first search that stops at the current branch (and stashes the intermediate result for possible later evaluation), if the total score goes below some expectation, and continues

somewhere else in the tree.

As a next step, we formulated models that did not constrain the evaluation strategy as much: although these were still greedy models, they were not constrained to a certain node-selection strategy, such as depth-first search. Also, they used additional eye gaze data to further inform the inference, and as such additional difficulties arose, such as having to model explicitly the possibility of back-traversing the search tree (checking behavior). We showed that the corresponding inference problem is very tricky indeed: We could not accurately estimate parameters with either of two different inference schemes (approximative EM across trials and MC on a per-trial level).

The difficulties that we faced in this work speak to the complex nature of psychiatric diseases, and cognitive processes associated with them. While we successfully developed a model for prediction of future development of major depressive disorder that was simply based on questionnaire data, we only succeeded in a better understanding of the metareasoning process insofar that we know that participants do not employ any kind of simple tree search, which is what we would have expected. The realization of how little we understand it provides fertile ground for future research on this topic.

Appendix A Behavioral model fits

This section shows detailed choice histograms and model fits from the winning model in section 6.4.2. Each histogram shows the data aggregated across all participants and trials, but separately for each game starting state and choice sequence length. Each bar corresponds to one possible choice sequence, with the corresponding choice rewards plotted as labels next to it. The histograms are sorted in order of descending return. For example, the very top bar in the left plot in Figure 64 refers to the choice sequence yielding the rewards 140 for the first choice, -20 for the second, and again -20 for the third choice. The model fits are plotted as red lines on top of the histograms.

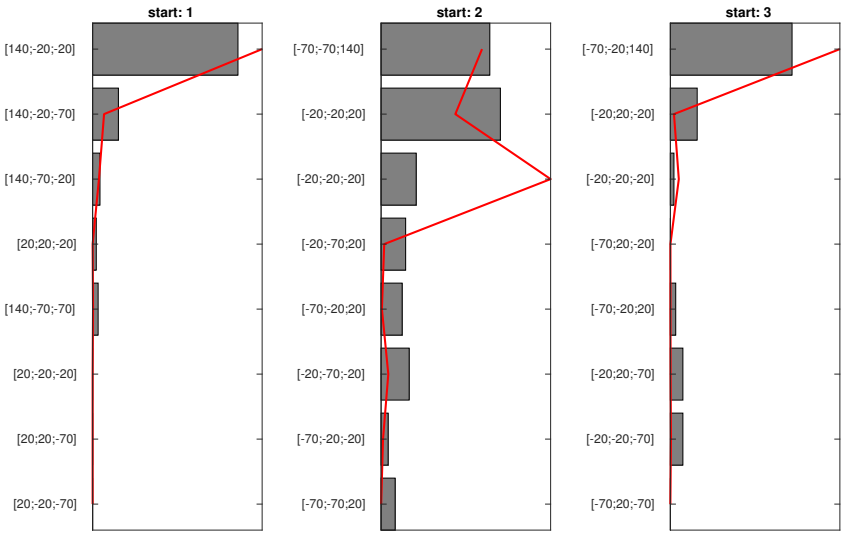


Figure 64: Choice sequence length 3, starting states 1-3.

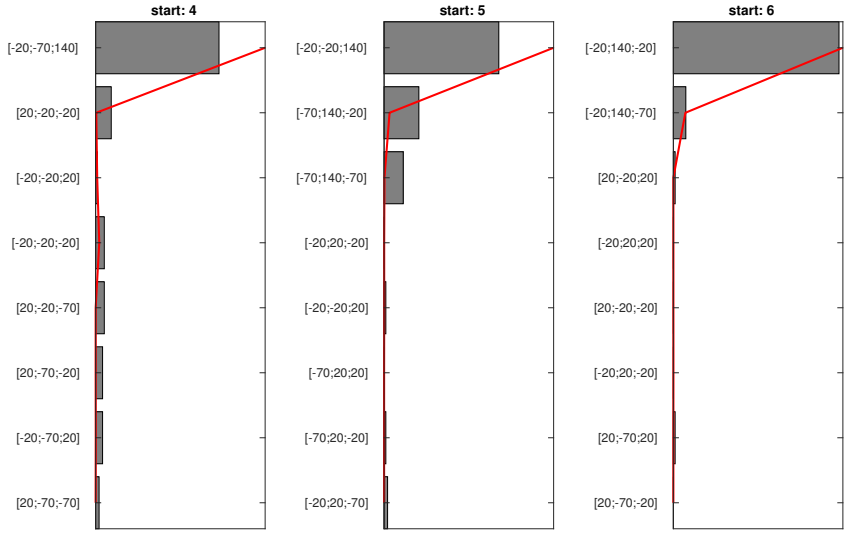


Figure 65: Choice sequence length 3, starting states 4-6.

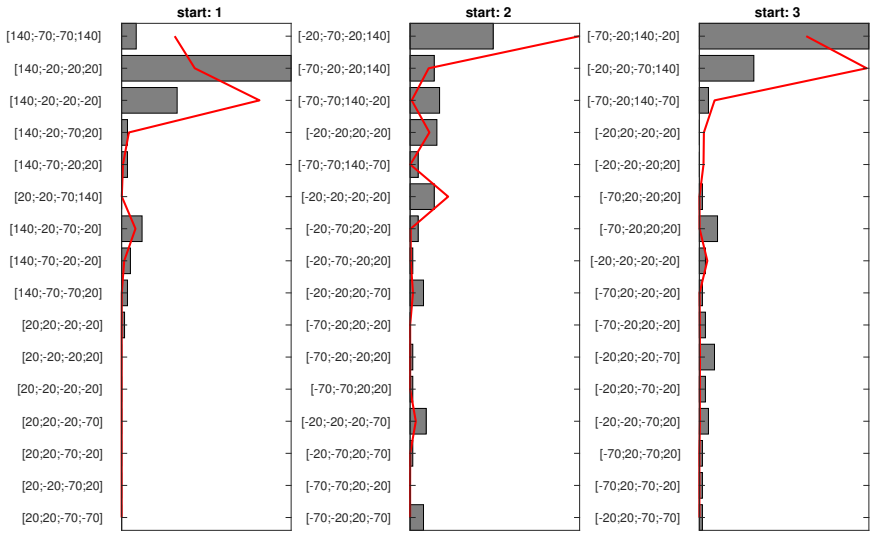


Figure 66: Choice sequence length 4, starting states 1-3.

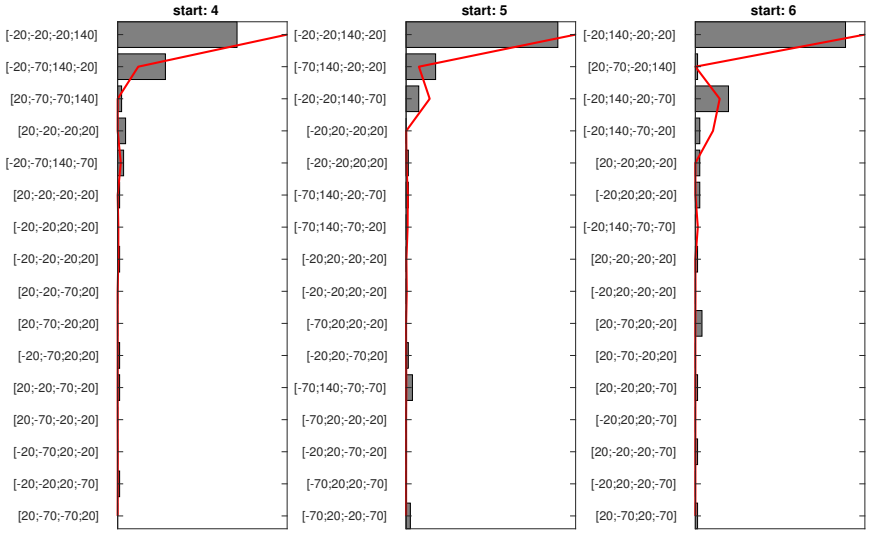


Figure 67: Choice sequence length 4, starting states 4-6.

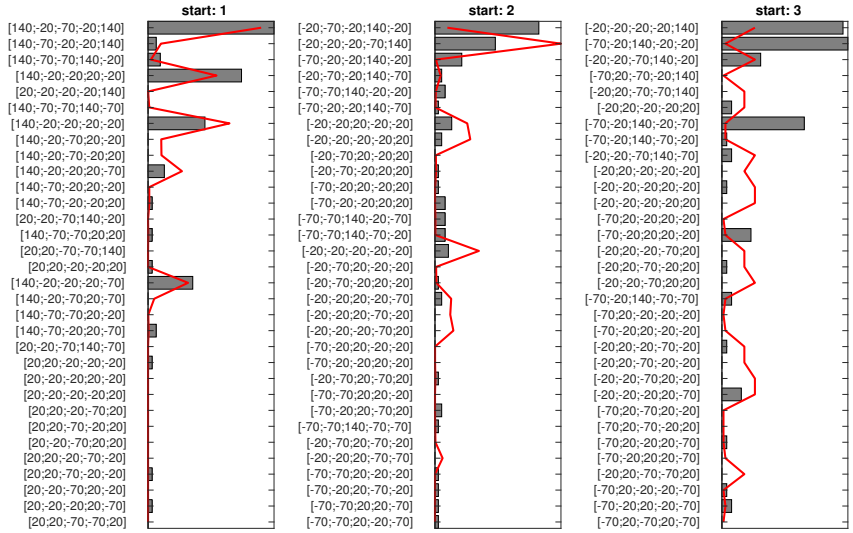


Figure 68: Choice sequence length 5, starting states 1-3.

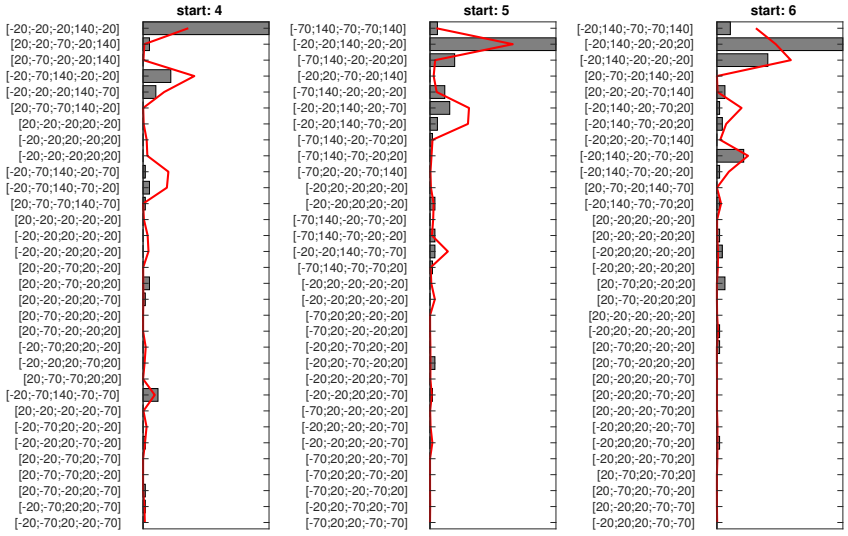


Figure 69: Choice sequence length 5, starting states 4-6.

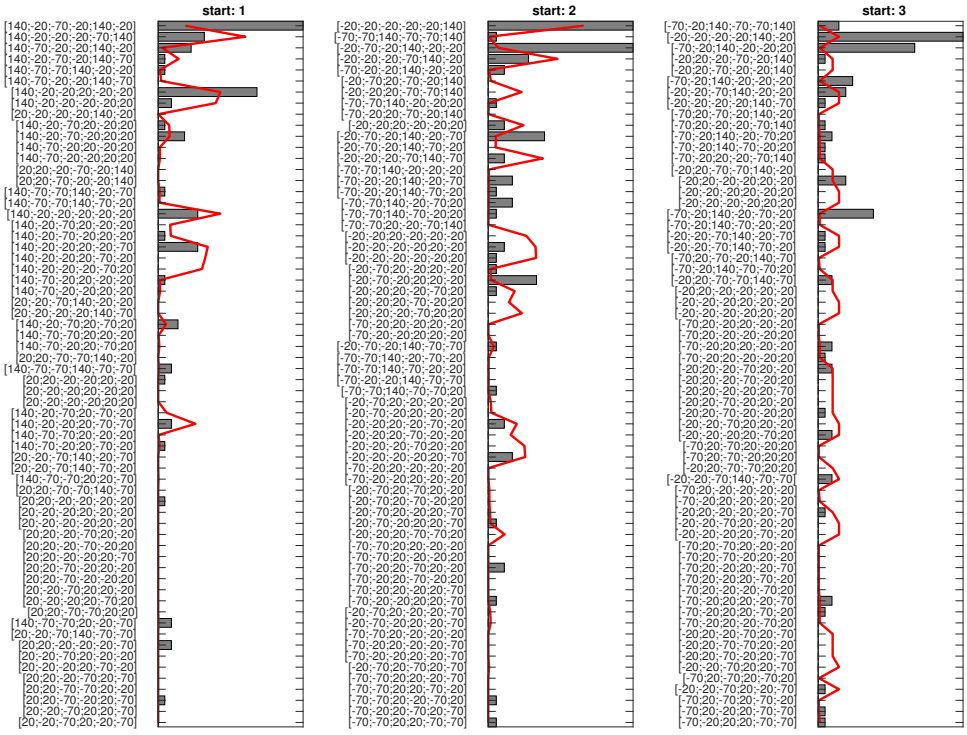


Figure 70: Choice sequence length 6, starting states 1-3.

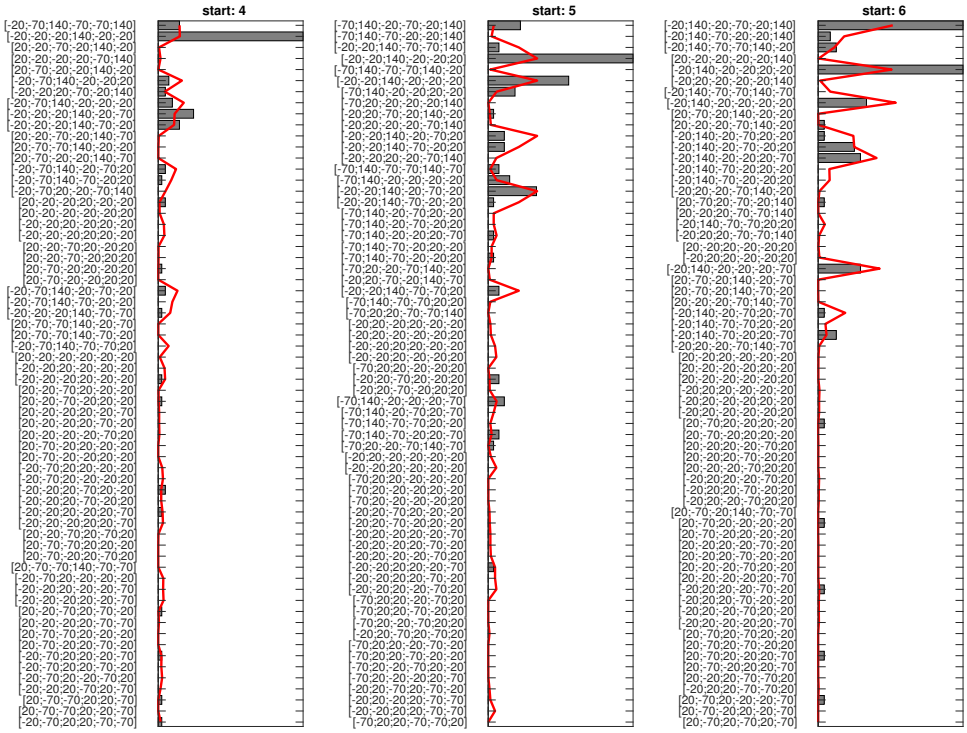


Figure 71: Choice sequence length 6, starting states 4-6.

Appendix B Examples of generated search trees

For some intuition of which search trees are generated by the model defined in section 6.5.2, we depict some examples here. We fixed the lengths of action sequences to 3, and randomly sampled 10000 search trees and corresponding action sequences with parameters $\theta = \exp(-5.0), \beta = \exp(-3.5), \rho = -10$, for each of the 6 possible game states (which are numbered according to Figure 30). The Figures below show the 24 most likely search trees for each game state. Each plot represents one search tree, and the game states corresponding to the tree nodes are shown in black squares. Red lines mark internal evaluations, so the search tree is shown in red; the complete underlying decision-tree is shown with grey lines. The order in which evaluations occur is given by the numbers displayed next to the red lines. 'Checking evaluations', should they occur, are shown in the smaller trees in the top left and right corners. Here, the starting state for that checking sequence is marked with a black circle. Since each edge can be traversed multiple times, the lines are drawn slightly displaced from the actual edge of the tree, such that following the red line starting from the black circle leads through the checking sequence. Additional, the probabilities of each action sequence conditioned on the generated search tree are shown as bars below each tree leaf, such that the height of each bar corresponds to the probability of the action sequence leading towards that leaf.

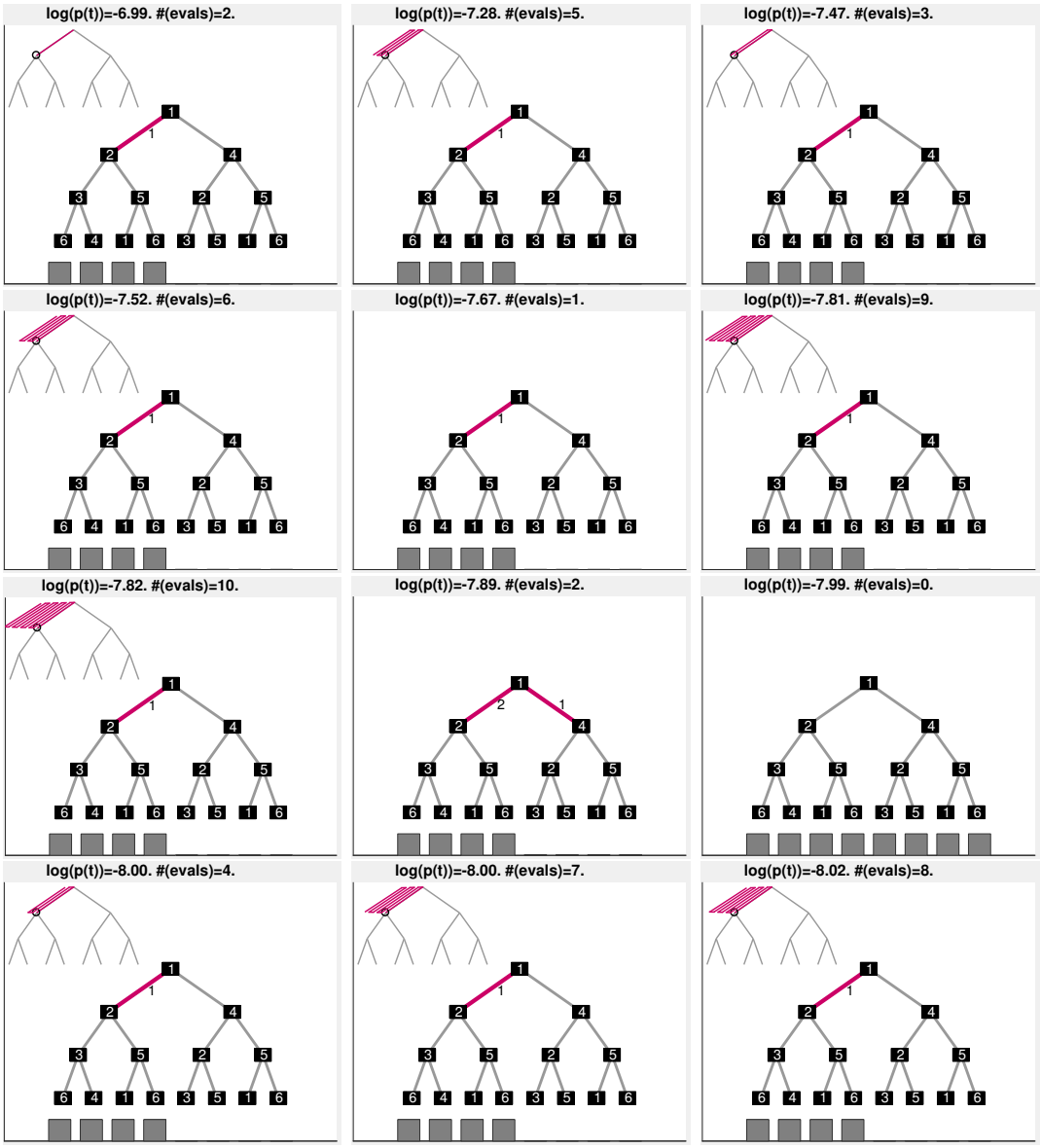


Figure 72: Starting state 1, search trees 1-12.

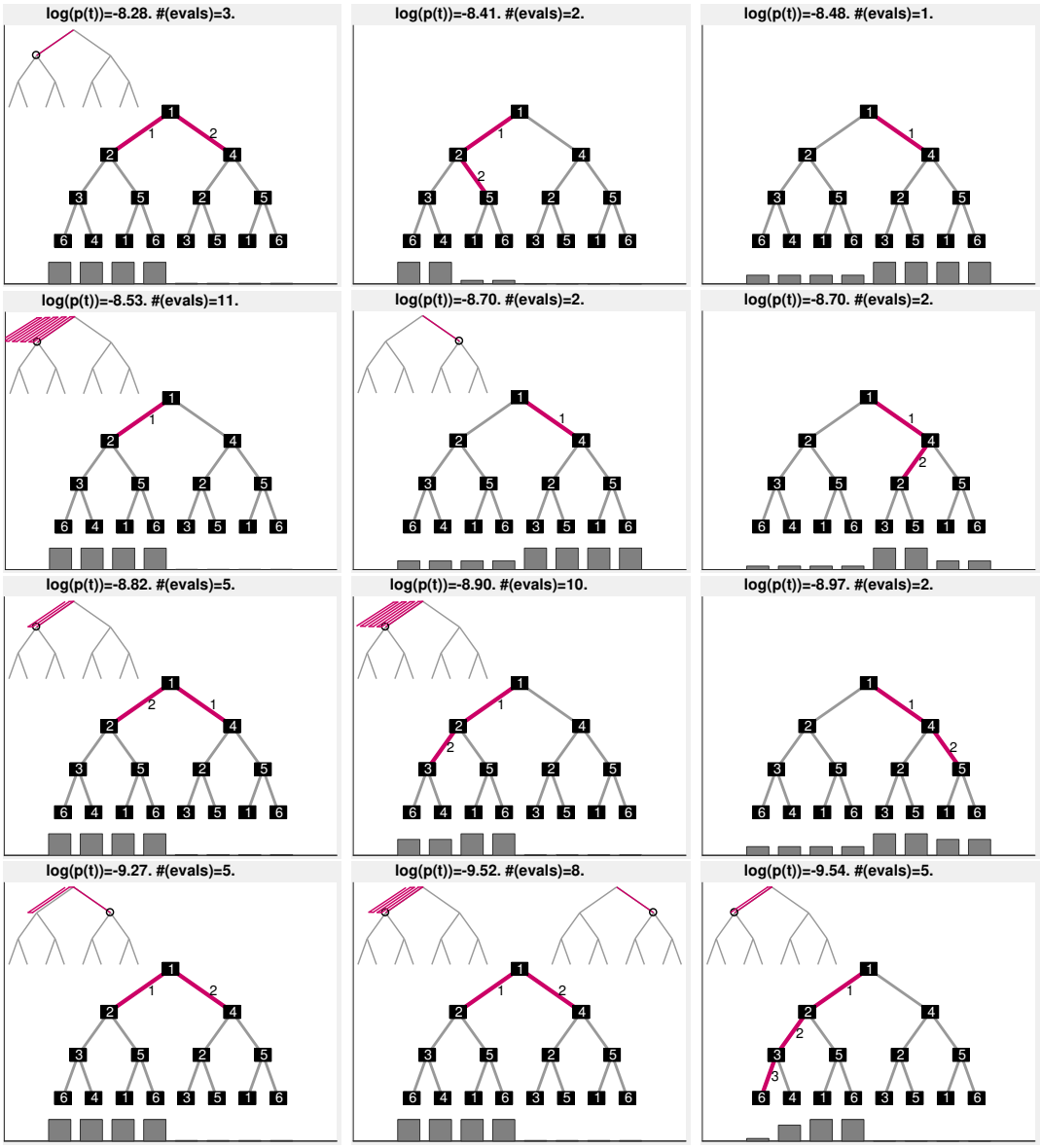


Figure 73: Starting state 1, search trees 13-24.

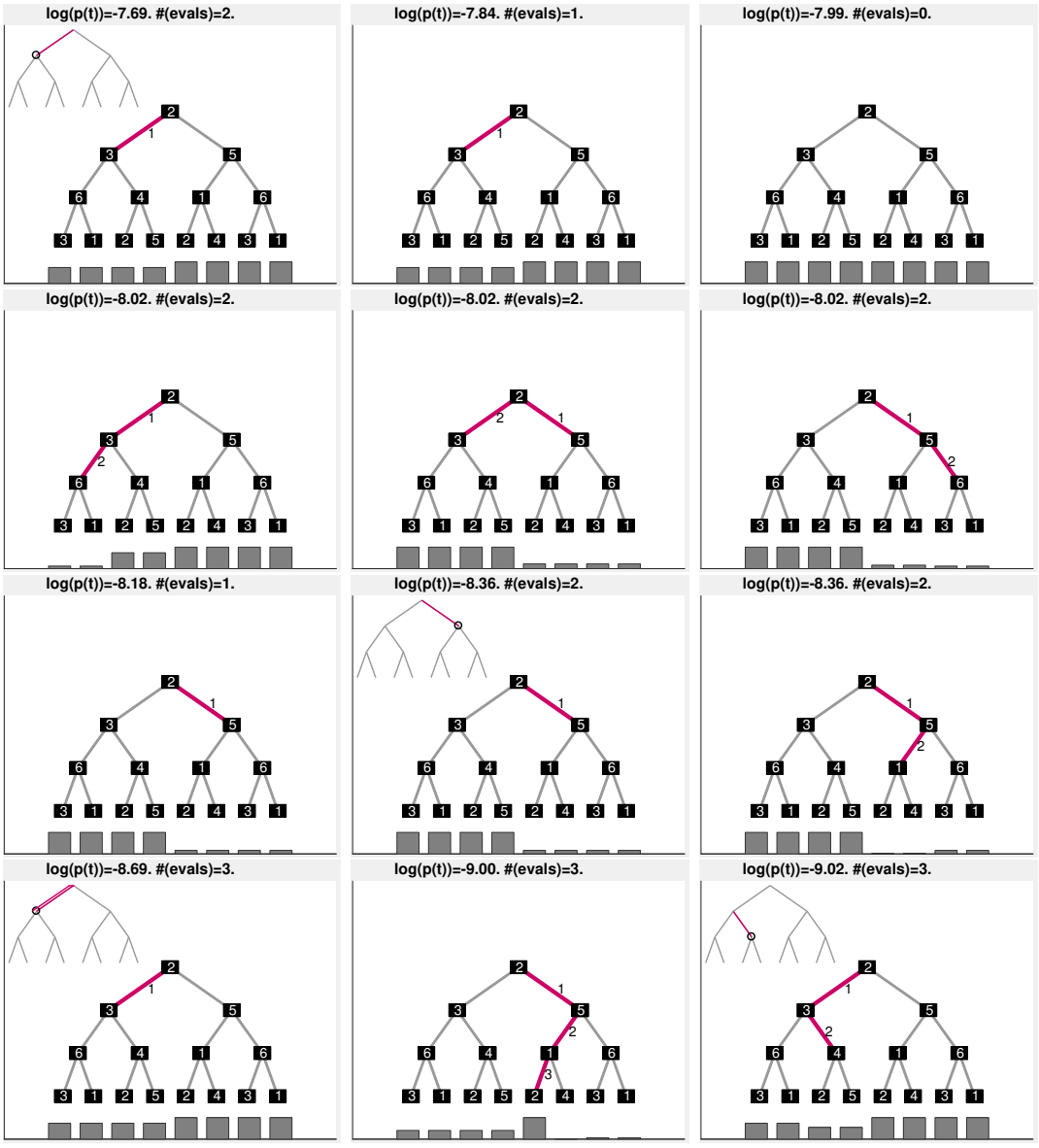


Figure 74: Starting state 2, search trees 1-12.

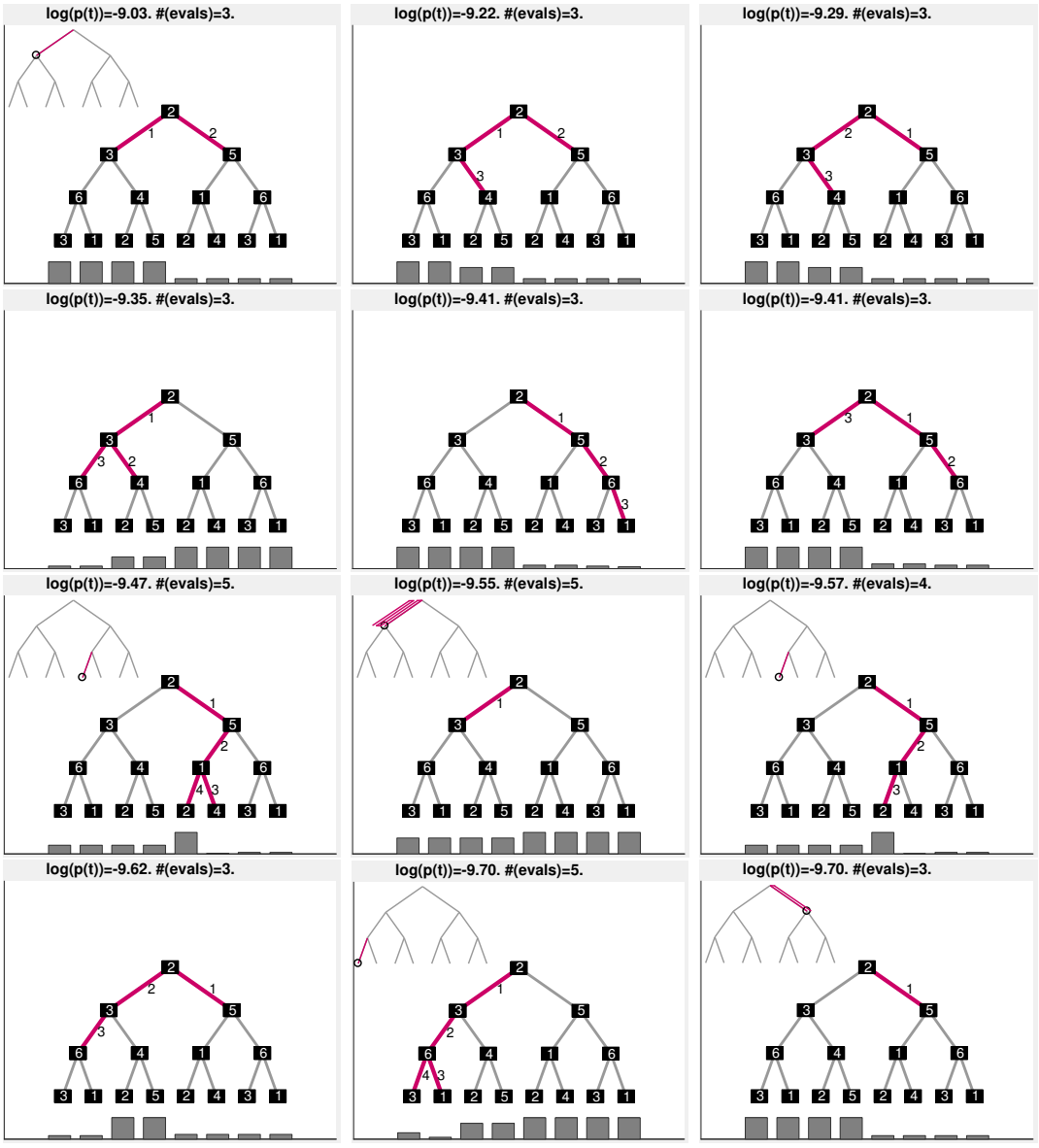


Figure 75: Starting state 2, search trees 13-24.

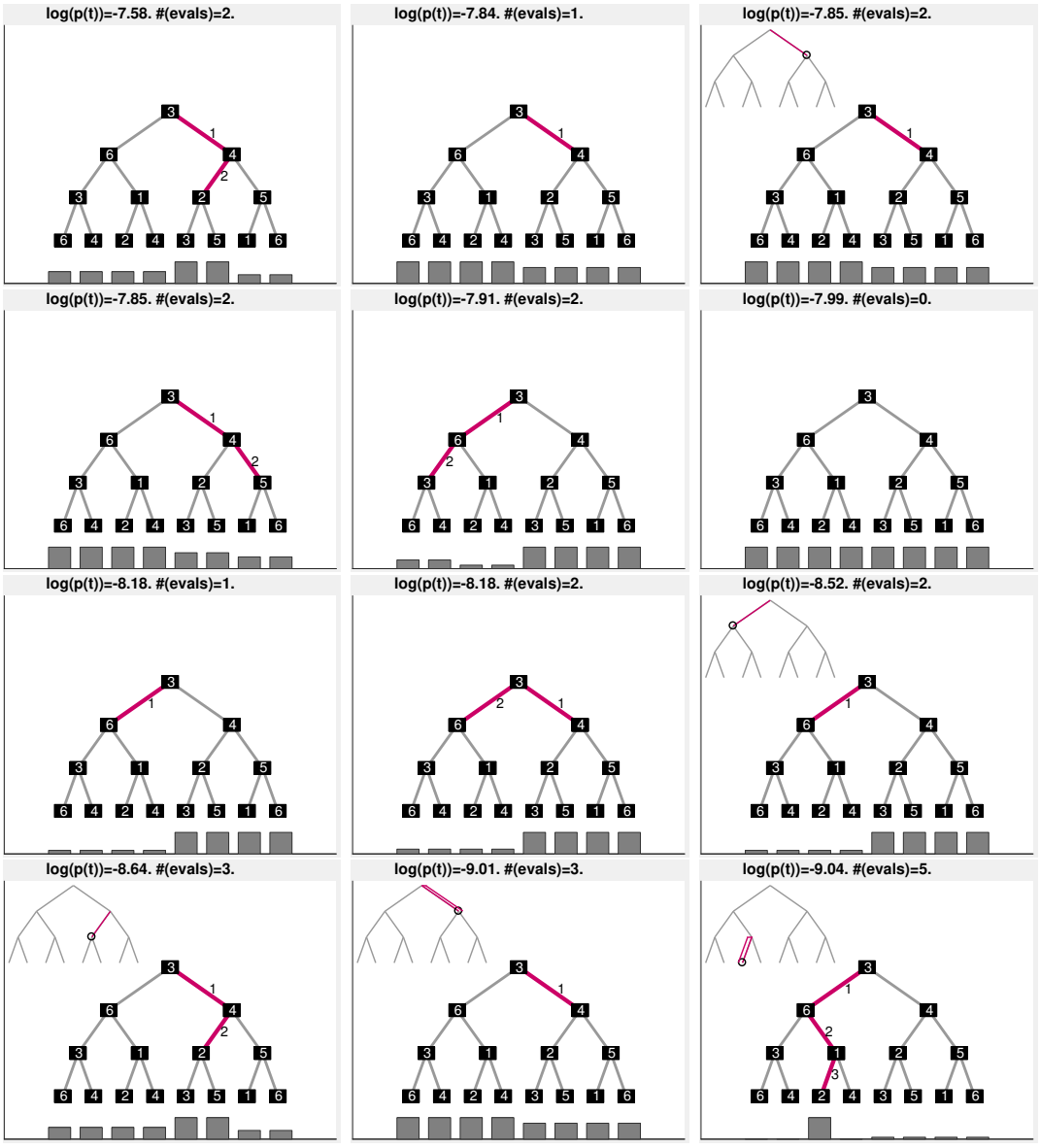


Figure 76: Starting state 3, search trees 1-12.

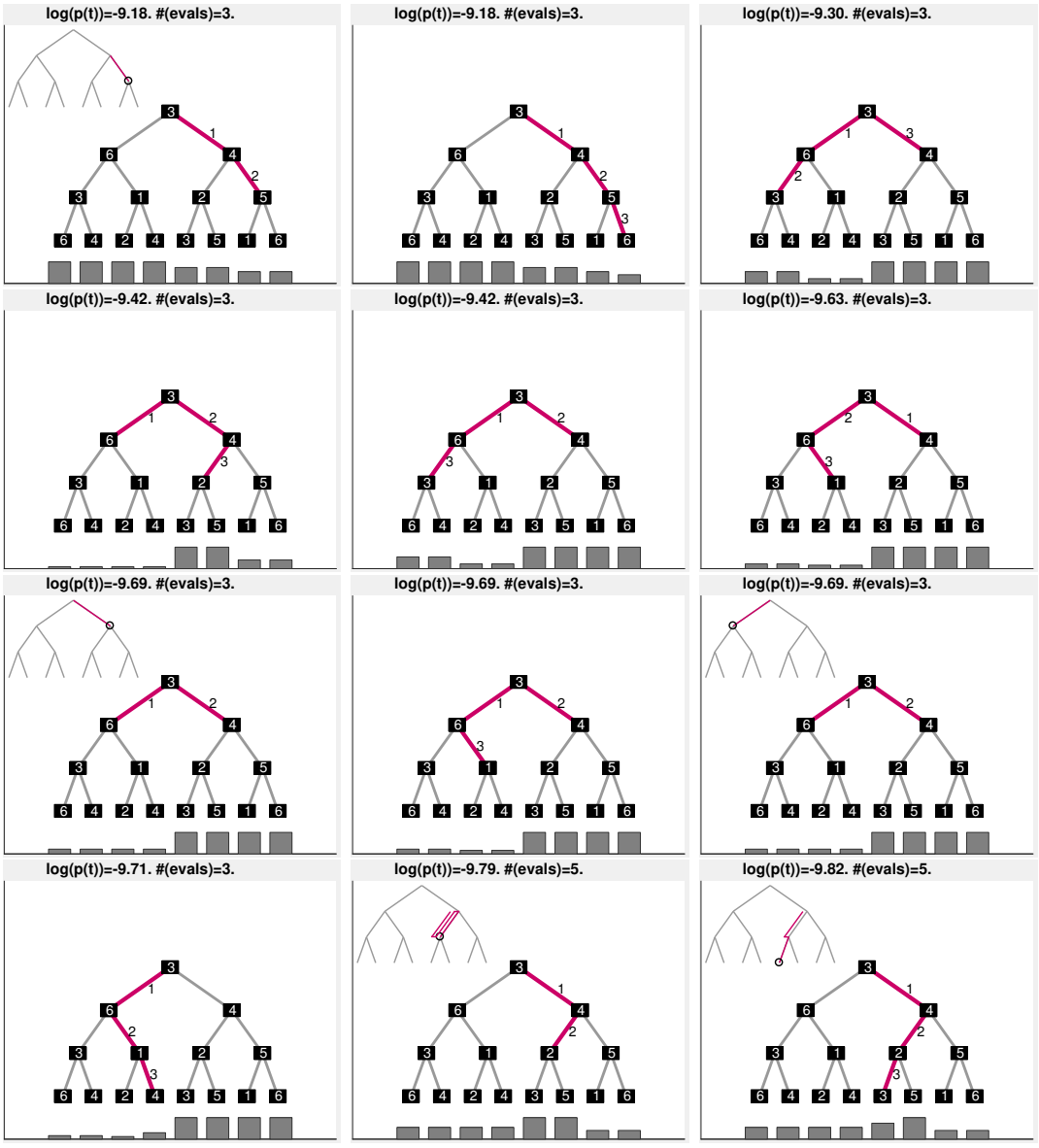


Figure 77: Starting state 3, search trees 13-24.

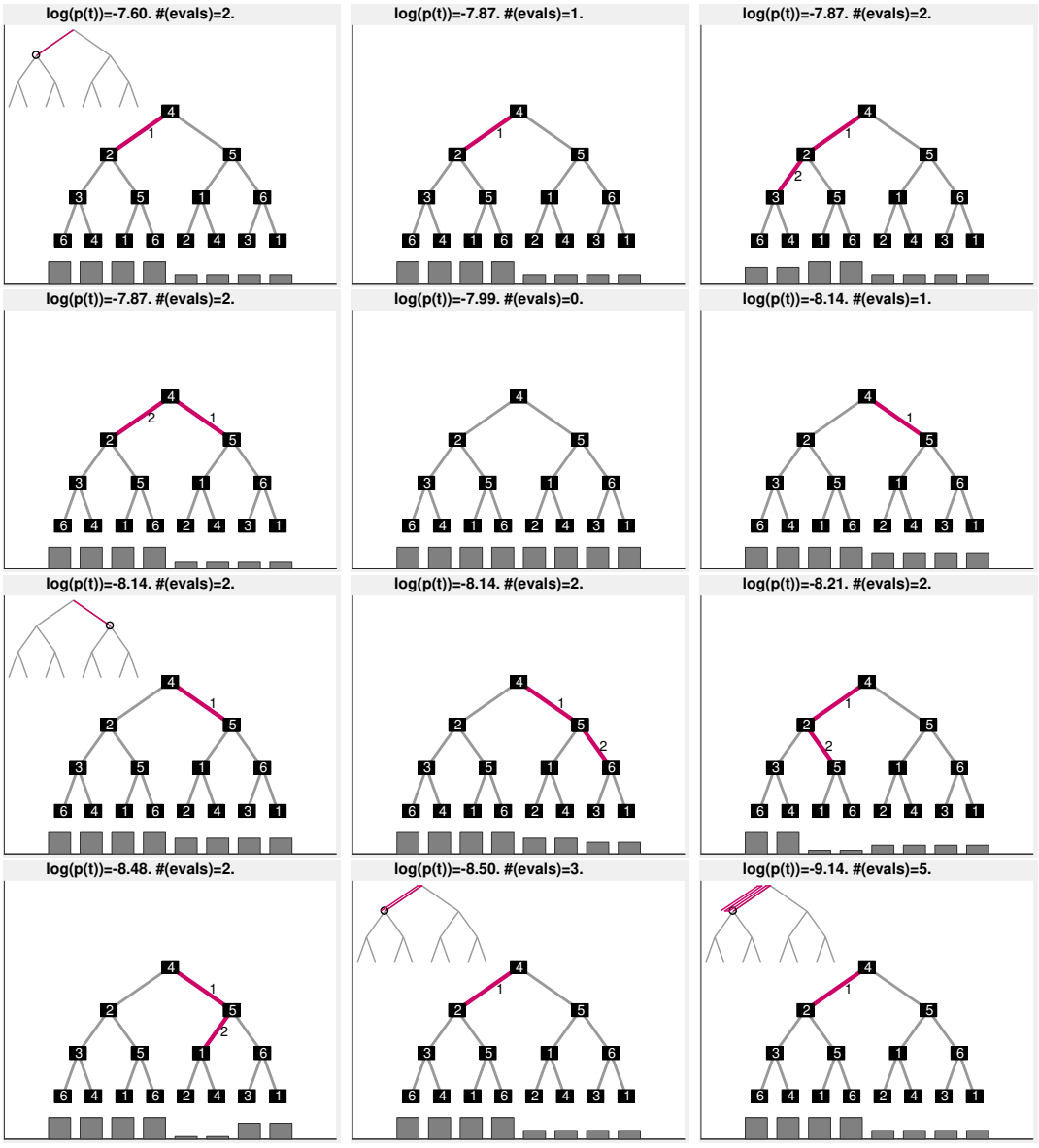


Figure 78: Starting state 4, search trees 1-12.

Figure 79: Starting state 4, search trees 13-24.

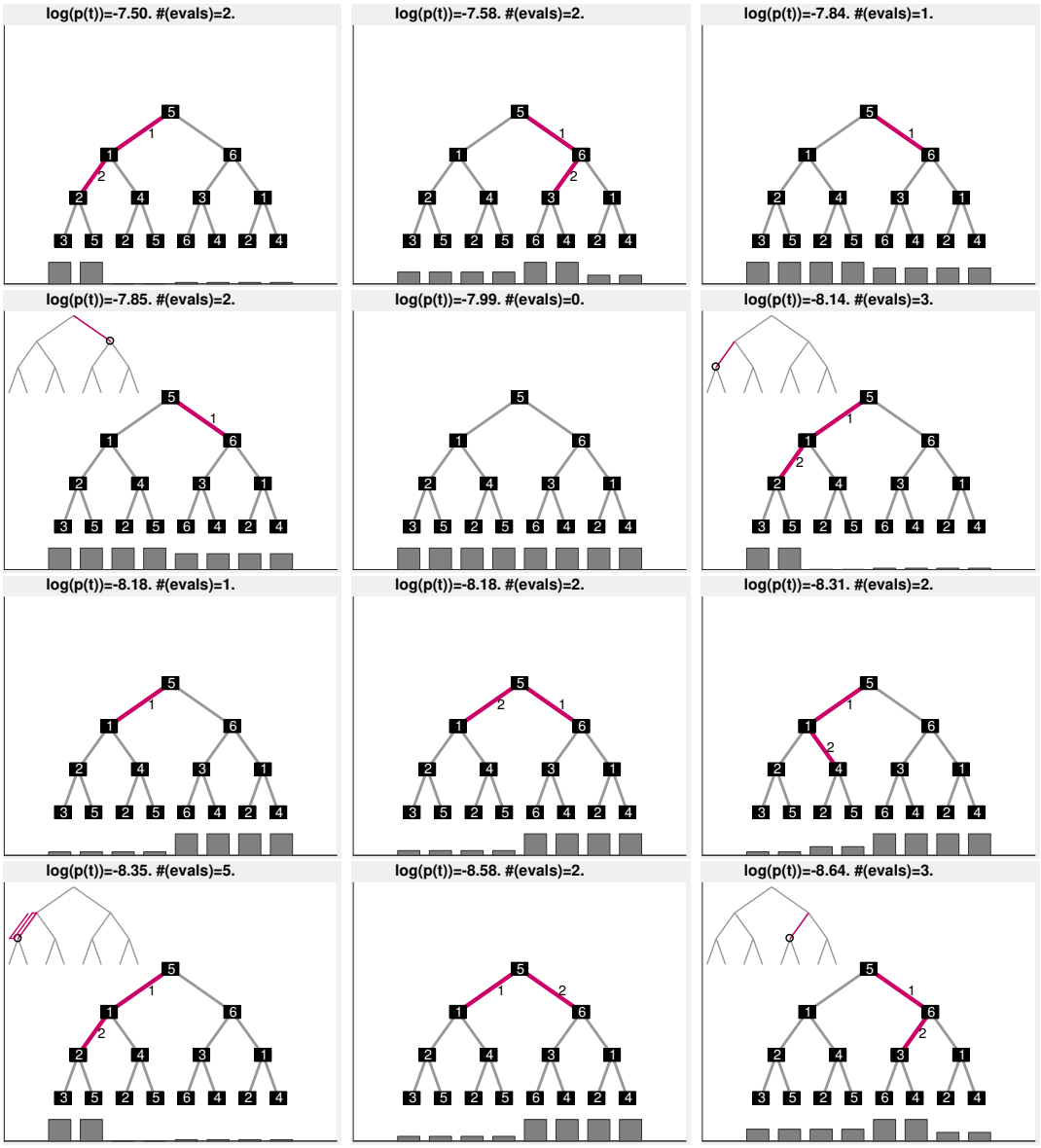


Figure 80: Starting state 5, search trees 1-12.

Figure 81: Starting state 5, search trees 13-24.

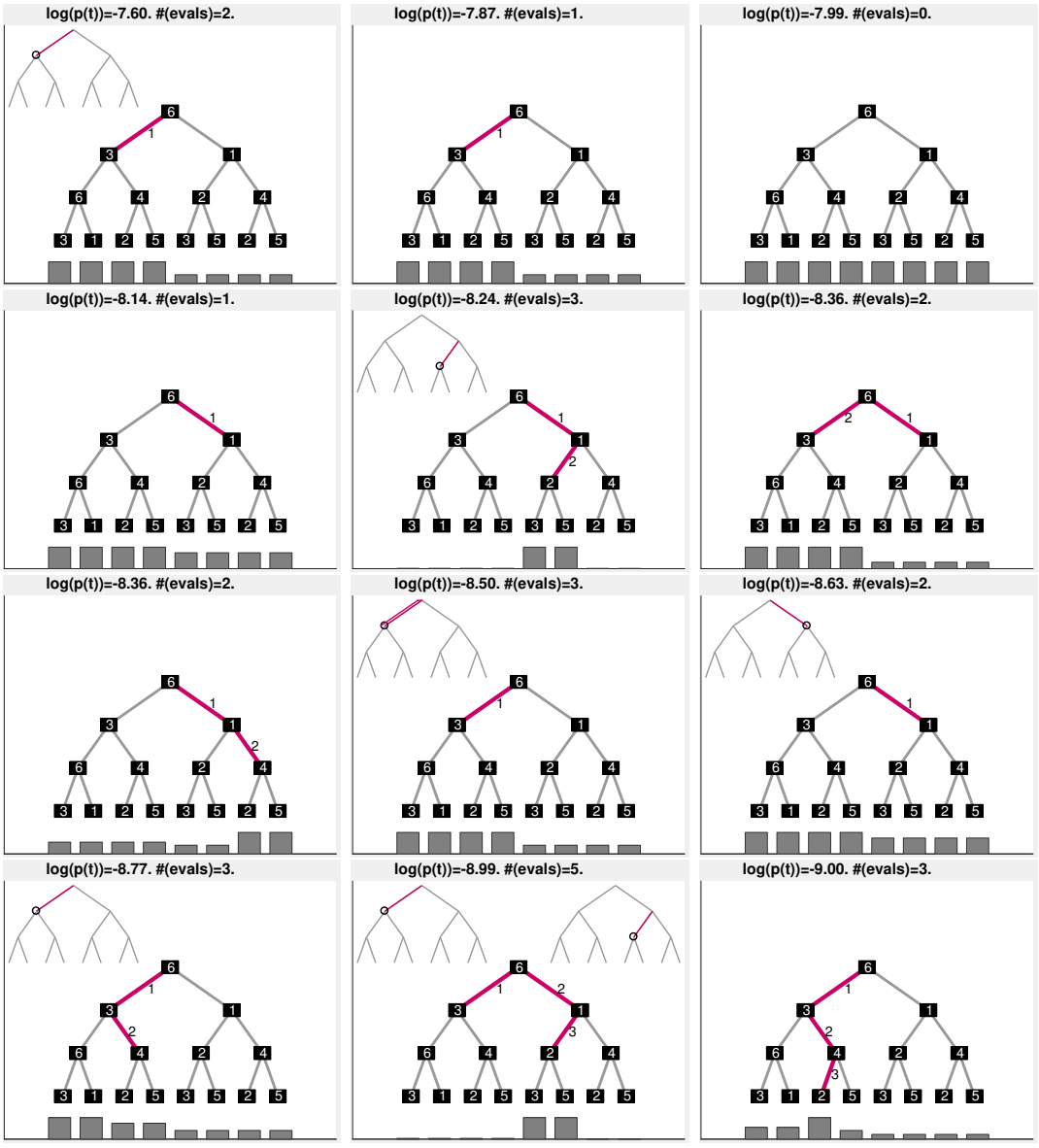


Figure 82: Starting state 6, search trees 1-12.

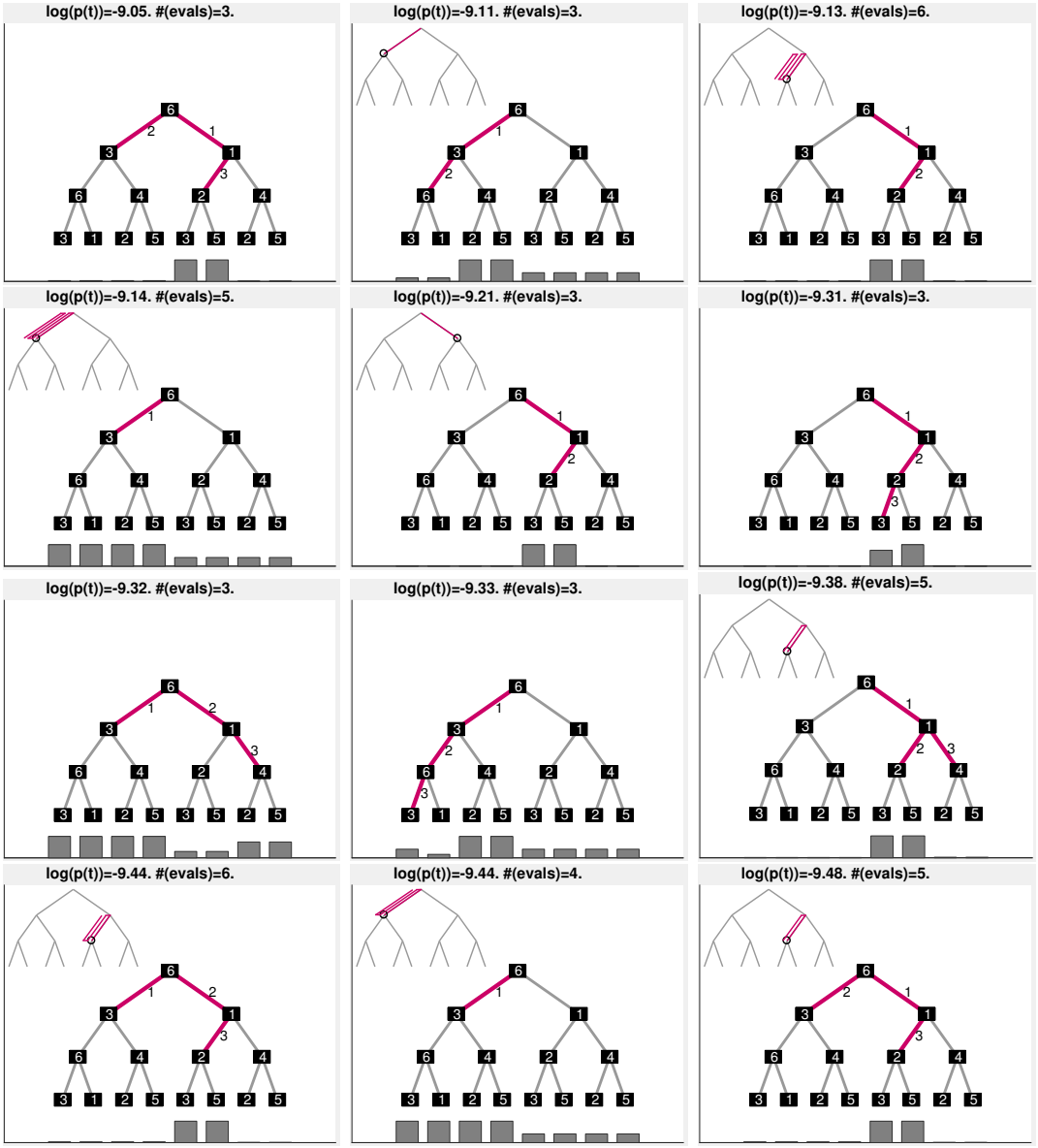


Figure 83: Starting state 6, search trees 13-24.

References

Afzali, M. H., Sunderland, M., Teesson, M., Carragher, N., Mills, K., and Slade, T. (2017). A network approach to the comorbidity between posttraumatic stress disorder and major depressive disorder: The role of overlapping symptoms. *Journal of Affective Disorders*, 208:490–496.

Alvarez, M. and Lawrence, N. D. (2009). Sparse Convolved Gaussian Processes for Multi-output Regression. *Advances in Neural Information Processing Systems*, pages 57—-64.

Anderson, M. L. and Oates, T. (2007). A Review of Recent Research in Metareasoning and Metalearning. *AI Magazine*, 28(1):7–16.

Andrews, P. W., Kornstein, S. G., Halberstadt, L. J., Gardner, C. O., and Neale, M. C. (2011). Blue again: perturbational effects of antidepressants suggest monoaminergic homeostasis in major depression. *Frontiers in psychology*, 2(July):159.

Angst, J., Gamma, A., Sellaro, R., Lavori, P. W., and Zhang, H. (2003). Recurrence of bipolar disorders and major depression. A life-long perspective. *European archives of psychiatry and clinical neuroscience*, 253(5):236–40.

Apanasovich, T. V., Genton, M. G., and Sun, Y. (2012). A Valid Matérn Class of Cross-Covariance Functions for Multivariate Random Fields With Any Number of Components. *Journal of the American Statistical Association*, 107(April 2014):180–193.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

Barber, D. (2011). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Barry, E. S., Naus, M. J., and Rehm, L. P. (2004). Depression and implicit memory: Understanding mood congruent memory bias.

Bayes, T. and Price, R. (1763). An Essay towards Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society B*, 53:370–418.

Beck, A. T. (1967). *Depression: Clinical, experimental and theoretical aspects*. University of Pennsylvania Press.

Beekman, A. T., De Beurs, E., Van Balkom, A. J., Deeg, D. J., Van Dyck, R., and Van Tilburg, W. (2000). Anxiety and depression in later life: Co-occurrence and communality of risk factors. *American Journal of Psychiatry*, 157(1):89–95.

Bellman, R. (1957). A Markovian decision process. *Journal Of Mathematics And Mechanics*, 6:679–684.

Bellman, R. E. (1961). *Adaptive control processes: A guided tour*. Princeton University Press.

Bentall, R. P. and Beck, A. T. (2005). *Madness Explained: Psychosis and Human Nature*. Penguin Global.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer, New York.

Bernardo, J. M. and Smith, A. F. M. (2008). *Bayesian Theory*. Wiley Series in Probability and Statistics.

Berry, D. A. and Fristedt, B. (1985). *Bandit problems: sequential allocation of experiments*. Chapman and Hall London.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York.

Bonilla, E., Chai, K. M., and Williams, C. (2008). Multi-task Gaussian Process Prediction. *Advances in Neural Information Processing Systems*, 20:153–160.

Bonilla, E. V., Agakov, F. V., and Williams, C. K. I. (2007). Kernel multi-task learning using task-specific features. *The 11th International Conference on Artificial Intelligence and Statistics*, pages 43–50.

Bora, E., Fornito, A., Pantelis, C., and Yücel, M. (2011). Gray matter abnormalities in Major Depressive Disorder: A meta-analysis of voxel based morphometry studies. *Journal of Affective Disorders*, 138(1-2):9–18.

Borsboom, D. (2008). Psychometric perspectives on diagnostic systems. *Journal of Clinical Psychology*, 64(9):1089–1108.

Borsboom, D. (2015). Problems Attract Problems: A Network Perspective on Mental Disorders. *Emerging Trends in the Social and Behavioral Sciences*, pages 1–15.

Borsboom, D., Cramer, A. O., Schmittmann, V. D., Epskamp, S., and Waldorp, L. J. (2011). The Small World of Psychopathology. *PLoS ONE*, 6(11).

Borsboom, D. and Cramer, A. O. J. (2013). Network Analysis: An Integrative Approach to the Structure of Psychopathology. *Annual Review of Clinical Psychology*, 9(1):91–121.

Bourgault, G. and Marcotte, D. (1991). Multivariable Variogram and Its Application To the Linear-Model of Coregionalization. *Mathematical Geology*, 23(7):899–928.

Box, G. E. P. (1950). Problems in the analysis of growth and wear data. *Biometrics*, 6:362–389.

Breslow, N. E. (1975). Analysis of survival data under the proportional hazards model. *International Statistical Review*, 43(1):45–57.

Brunsdon, C., Fotheringham, A. S., and Charlton, M. E. (1996). Geographically Weighted Regression: A method for exploring spatial nonstationarity. *Geographical Analysis*, 28(4):281–298.

Bubeck, S., Munos, R., and Stoltz, G. (2009). *Pure exploration in multi-armed bandits problems*. Springer.

Bussas, M., Sawade, C., Scheffer, T., and Landwehr, N. (2015). Varying-coefficient models with isotropic Gaussian process priors. *arXiv preprint arXiv:1508.07192*, pages 1–17.

Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–75.

Chekroud, A. M. (2017). Bigger Data, Harder QuestionsOpportunities Throughout Mental Health Care. *JAMA Psychiatry*, 74(12):1183–1184.

Chekroud, A. M., Gueorguieva, R., Krumholz, H. M., Trivedi, M. H., Krystal, J. H., and McCarthy, G. (2017a). Reevaluating the efficacy and predictability of antidepressant treatments: A symptom clustering approach. *JAMA Psychiatry*, 74(4):370–378.

Chekroud, A. M., Lane, C. E., and Ross, D. A. (2017b). Computational Psychiatry: Embracing Uncertainty and Focusing on Individuals, Not Averages.

Chekroud, A. M., Zotti, R. J., Shehzad, Z., Gueorguieva, R., Johnson, M. K., Trivedi, M. H., Cannon, T. D., Krystal, J. H., and Corlett, P. R. (2016). Cross-trial prediction of treatment outcome in depression: a machine learning approach. *The Lancet Psychiatry*, 3(3):243–250.

Chou, C.-P., Bentler, P. M., and Pentz, M. A. (1998). Comparisons of two statistical approaches to study growth curves: The multilevel model and the latent curve analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 5(3):247–266.

Cipriani, A., Furukawa, T. A., Salanti, G., Geddes, J. R., Higgins, J. P., Churchill, R., Watanabe, N., Nakagawa, A., Omori, I. M., McGuire, H., Tansella, M., and Barbui, C. (2009). Comparative efficacy and acceptability of 12 new-generation antidepressants: a multiple-treatments meta-analysis. *The Lancet*, 373(9665):746–758.

Clementz, B. A., Sweeney, J. A., Hamm, J. P., Ivleva, E. I., Ethridge, L. E., Pearlson, G. D., Keshavan, M. S., and Tamminga, C. A. (2016). Identification of distinct psychosis biotypes using brain-based biomarkers. *American Journal of Psychiatry*, 173(4):373–384.

Conitzer, V. and Sandholm, T. (2003). Definition and complexity of some basic metareasoning problems. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1099–1106.

Cox, D. R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society*, 34:187–220.

Cox, D. R. (1975). Partial likelihood. *Biometrika*, 62(2):269–276.

Cramer, A. O. J., van Borkulo, C. D., Giltay, E. J., van der Maas, H. L. J., Kendler, K. S., Scheffer, M., and Borsboom, D. (2016). Major Depression as a Complex Dynamic System. *Plos One*, 11(12):e0167490.

Cramer, A. O. J., Waldorp, L. J., Van Der Maas, H. L. J., and Borsboom, D. (2010). Comorbidity: A network perspective.

Curran, P. J. (2003). Have Multilevel Models Been Structural Equation Models All Along? *Multivariate Behavioral Research*, 38(4):529–569.

Curran, P. J., Obeidat, K., and Losardo, D. (2010). Twelve frequently asked questions about growth curve modeling. *Journal of Cognitive Development*, 11(2):121–136.

Damasio, A. R. (1997). Neuropsychology. Towards a neuropathology of emotion and mood. *Nature*, 386:769–770.

Davis, J., Maes, M., Andreazza, A., McGrath, J. J., Tye, S. J., and Berk, M. (2015). Towards a classification of biomarkers of neuropsychiatric disease: From encompass to compass.

Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., and Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215.

Dayan, P. and Huys, Q. J. M. (2008). Serotonin, inhibition, and negative mood. *PLoS computational biology*, 4(2):e4.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38.

DeRubeis, R. J., Siegle, G. J., and Hollon, S. D. (2008). Cognitive therapy versus medication for depression: treatment outcomes and neural mechanisms. *Nature reviews. Neuroscience*, 9(10):788–96.

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Networks*, 12(7-8):961–974.

Ebmeier, K. P., Donaghey, C., and Steele, J. D. (2006). Recent developments and current controversies in depression. *Lancet*, 367(9505):153–67.

Elliott, R., Zahn, R., Deakin, J. F. W., and Anderson, I. M. (2011). Affective cognition and its disruption in mood disorders. *Neuropsychopharmacology*, 36(1):153–82.

Etkin, A. and Wager, T. D. (2007). Functional neuroimaging of anxiety: a meta-analysis of emotional processing in PTSD, social anxiety disorder, and specific phobia. *The American journal of psychiatry*, 164(10):1476–1488.

Fan, J. and Zhang, W. (2008). Statistical Methods with Varying Coefficient Models. *Statistics and its interface*, 1(1):179–195.

Ferrari, A. J., Charlson, F. J., Norman, R. E., Patten, S. B., Freedman, G., Murray, C. J., Vos, T., and Whiteford, H. a. (2013). Burden of Depressive Disorders by Country, Sex, Age, and Year: Findings from the Global Burden of Disease Study 2010. *PLoS Medicine*, 10(11):e1001547.

Finley, A. O., Banerjee, S., and Carlin, B. P. (2007). spBayes: An R Package for Univariate and Multivariate Hierarchical Point-referenced Spatial Models. *Journal of statistical software*, 19(4):1–24.

Finley, A. O., Banerjee, S., and Gelfand, A. E. (2015). spBayes for Large Univariate and Multivariate Point-Referenced Spatio-Temporal Data Models. *Journal of Statistical Software*, 63(13):1–28.

Fisher, R. A. (1925). Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5).

Forand, N. R. and DeRubeis, R. J. (2014). Extreme response style and symptom return after depression treatment: The role of positive extreme responding. *Journal of consulting and clinical psychology*, 82(3):500.

Fried, E. I. and Nesse, R. M. (2015). Depression is not a consistent syndrome: An investigation of unique symptom patterns in the STARD study. *Journal of Affective Disorders*, 172:96–102.

Fried, E. I., van Borkulo, C. D., Cramer, A. O. J., Boschloo, L., Schoevers, R. A., and Borsboom, D. (2016). Mental disorders as networks of problems: a review of recent insights. *Social Psychiatry and Psychiatric Epidemiology*, 58(12):7250–7.

Frijda, N. H. (1987). *The emotions: Studies in emotion and social interaction*. Cambridge University Press.

Friston, K. J., Penny, W., Phillips, C., Kiebel, S., Hinton, G., and Ashburner, J. (2002). Classical and Bayesian inference in neuroimaging: theory. *NeuroImage*, 16(2):465–483.

Gartlehner, G., Hansen, R. A., Morgan, L. C., Thaler, K., Lux, L., van Noord, M., Mager, U., Thieda, P., Gaynes, B. N., Wilkins, T., Strobelberger, M., Lloyd, S., Reichenpfader, U., and Lohr, K. N. (2011). Comparative benefits and harms of second-generation antidepressants for treating major depressive disorder: An updated meta-analysis.

Geddes, J. R., Carney, S. M., Davies, C., Furukawa, T. a., Kupfer, D. J., Frank, E., and Goodwin, G. M. (2003). Relapse prevention with antidepressant drug treatment in depressive disorders: a systematic review. *The Lancet*, 361(9358):653–661.

Gelfand, A. E., Kim, H.-J., Sirmans, C. F., and Banerjee, S. (2003). Spatial Modeling With Spatially Varying Coefficient Processes. *Journal of the American Statistical Association*, 98(462):387–396.

Gelfand, A. E. and Schliep, E. M. (2016). Spatial statistics and Gaussian processes: A beautiful marriage. *Spatial Statistics*, 18:86–104.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall/CRC.

Genton, M. G. and Kleiber, W. (2015). Cross-Covariance Functions for Multivariate Geostatistics. *Statistical Science*, 30(2):147–163.

Gershman, S. J., Markman, A. B., and Otto, A. R. (2014). Retrospective revaluation in sequential decision making: a tale of two systems. *Journal of experimental psychology: General*, 143(1):182–94.

Glue, P., Donovan, M. R., Kolluri, S., and Emir, B. (2010). Meta-analysis of relapse prevention antidepressant trials in depressive disorders. *The Australian and New Zealand journal of psychiatry*, 44(8):697–705.

Gneiting, T., Kleiber, W., and Schlather, M. (2010). Matérn Cross-Covariance Functions for Multivariate Random Fields. *Journal of the American Statistical Association*, 105(April 2014):1167–1177.

Goldstein, H. (2011). *Multilevel Statistical Models*. Wiley.

Gotlib, I. H. and Joormann, J. (2010). Cognition and depression: current status and future directions. *Annual review of clinical psychology*, 6:285–312.

Gotway, C. A. and Wolfinger, R. D. (2003). Spatial prediction of counts and rates. *Statistics in Medicine*, 22(9):1415–1432.

Griffiths, T. L., Lieder, F., and Goodman, N. D. (2015). Rational Use of Cognitive Resources: Levels of Analysis Between the Computational and the Algorithmic. *Topics in Cognitive Science*, 7(2):217–229.

Grisanzio, K. A., Goldstein-Piekarski, A. N., Wang, M. Y., Rashed Ahmed, A. P., Samara, Z., and Williams, L. M. (2017). Transdiagnostic Symptom Clusters and Associations With Brain, Behavior, and Daily Function in Mood, Anxiety, and Trauma Disorders. *JAMA Psychiatry*.

Gross, J. J. (2002). Emotion regulation : Affective, cognitive, and social consequences. *Psychophysiology*, 39:281–291.

Haeffel, G. J., Gibb, B. E., Metalsky, G. I., Alloy, L. B., Abramson, L. Y., Hankin, B. L., Joiner, T. E., and Swendsen, J. D. (2008). Measuring cognitive vulnerability to depression: Development and validation of the cognitive style questionnaire. *Clinical Psychology Review*, 28(5):824–836.

Hannah, L. A., Blei, D. M., and Powell, W. B. (2011). Dirichlet Process Mixtures of Generalized Linear Models. *Journal of Machine Learning Research*, 1:1–33.

Harville, D. A. (1977). Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. *Journal of the American Statistical Association*, 72(358):320–338.

Hastie, T. and Tibshirani, R. (1993). Varying-Coefficient Models. *Journal of the Royal Statistical Society Series B-Methodological*, 55(4):757–796.

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hay, N. J. and Russell, S. J. (2011). Metareasoning for Monte Carlo Tree Search. Technical Report UCB/EECS-2011-119, EECS Department, University of California, Berkeley.

Hertz-Picciotto, I. and Rockhill, B. (1997). Validity and efficiency of approximation methods for tied survival times in Cox regression. *Biometrics*, 53(3):1151–6.

Hoffman, L. and Rovine, M. J. (2007). Multilevel models for the experimental psychologist: foundations and illustrative examples. *Behavior research methods*, 39(1):101–117.

Hogan, J. W., Lin, X., and Herman, B. (2004). Mixtures of varying coefficient models for longitudinal data with discrete or continuous nonignorable dropout. *Biometrics*, 60(4):854–864.

Hox, J. and Stoel, R. D. (2005). Multilevel and SEM Approaches to Growth Curve Modeling. In *Encyclopedia of Statistics in Behavioral Science, Vol 3*, pages 1296–1305. Wiley.

Hox, J. J., Moerbeek, M., and van de Schoot, R. (2010). *Multilevel analysis: Techniques and applications*. Routledge.

Huang, B., Wu, B., and Barry, M. (2010). Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices. *International Journal of Geographical Information Science*, 24(3):383–401.

Huys, Q. J. M. (2007). *Reinforcers and control*. PhD thesis, University College London.

Huys, Q. J. M. (2018). Computational Psychiatry: pragmatic and explanatory. *JAMA Psychiatry*, In Press.

Huys, Q. J. M., Cools, R., Gölzer, M., Friedel, E., Heinz, A., Dolan, R. J., and Dayan, P. (2011). Disentangling the roles of approach, activation and valence in instrumental and pavlovian responding. *PLoS computational biology*, 7(4):e1002028.

Huys, Q. J. M., Daw, N. D., and Dayan, P. (2015a). Depression : A Decision-Theoretic Analysis. *Annual review of neuroscience*, 38:1–23.

Huys, Q. J. M. and Dayan, P. (2009). A Bayesian formulation of behavioral control. *Cognition*, 113(3):314–28.

Huys, Q. J. M., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., and Roiser, J. P. (2012). Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS computational biology*, 8(3):e1002410.

Huys, Q. J. M., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J., Dayan, P., and Roiser, J. P. (2015b). Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, page 201414219.

Huys, Q. J. M., Maia, T. V., and Frank, M. J. (2016). Computational psychiatry as a bridge from neuroscience to clinical applications. *Nature Neuroscience*, 19(3):404–413.

Huys, Q. J. M. and Renz, D. (2017). A formal valuation approach to emotions and their control. *Biological psychiatry*, 82(6):413–420.

Ipser, J. C., Singh, L., and Stein, D. J. (2013). Meta-analysis of functional brain imaging in specific phobia.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London A*, 186(1007).

Kapur, S., Phillips, A. G., and Insel, T. R. (2012). Why has it taken so long for biological psychiatry to develop clinical tests and what to do about it.

Karch, J. D. (2016). *A machine learning perspective on repeated measures: Gaussian process panel and person-specific EEG modeling.* PhD thesis, Humboldt Universität zu Berlin, Germany.

Kearns, M., Mansour, Y., and Ng, A. Y. (1999). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *International Joint Conference on Artificial Intelligence*, volume 2, pages 1324–1331.

Keller, M. C., Neale, M. C., and Kendler, K. S. (2007). Association of different adverse life events with distinct patterns of depressive symptoms. *American Journal of Psychiatry*, 164(10):1521–1529.

Kendler, K. S., Gardner, C. O., Gatz, M., and Pedersen, N. L. (2007). The sources of co-morbidity between major depression and generalized anxiety disorder in a Swedish national twin sample. *Psychological Medicine*, 37(3):453–462.

Kenward, M. G. and Roger, J. H. (1997). Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*, 53(3):983–97.

Kessler, R. C., Berglund, P., Demler, O., Jin, R., Merikangas, K. R., and Walters, E. E. (2005a). Lifetime prevalence and age-of-onset distributions of DSM-IV disorders in the national comorbidity survey replication.

Kessler, R. C., Chiu, W. T., Demler, O., Merikangas, K. R., and Walters, E. (2005b). Prevalence, severity, and comorbidity of 12-month DSM-IV disorders in the National Comorbidity Survey Replication. *Archives of general psychiatry*, 62(6):617–627.

Knuth, D. E. and Moore, R. W. (1975). An analysis of alpha-beta pruning. *Artificial Intelligence*, 6(4):293–326.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. *Proceedings of ECML*, pages 282–203.

Koolschijn, P. C. M. P., Van Haren, N. E. M., Lensvelt-Mulders, G. J. L. M., Hulshoff Pol, H. E., and Kahn, R. S. (2009). Brain volume abnormalities in major depressive disorder: A meta-analysis of magnetic resonance imaging studies. *Human Brain Mapping*, 30(11):3719–3735.

Krystal, J. H., Murray, J. D., Chekroud, A. M., Corlett, P. R., Yang, G., Wang, X. J., and Anticevic, A. (2017). Computational psychiatry and the challenge of schizophrenia.

Kupfer, D. J., Frank, E., and Phillips, M. L. (2012). Major depressive disorder: new clinical, neurobiological, and treatment perspectives. *Lancet*, 379(9820):1045–55.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974.

Lamers, F., Burstein, M., He, J.-p., Avenevoli, S., Angst, J., and Merikangas, K. R. (2012). Structure of major depressive disorder in adolescents and adults in the US general population. *The British Journal of Psychiatry*, 201(2):143–150.

Lauritzen, S. (1981). Time series analysis in 1880: A discussion of contributions made by TN Thiele. *Statistics*, 49(3):319–331.

Lázaro-Gredilla, M., Quinonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11:1865–1881.

LeDoux, J. (2012). Rethinking the Emotional Brain.

Lee, E., Wang, J., Kleinbaum, D. G., and Collett, D. (2003). *Statistical methods for survival data analysis*. Wiley.

Lee, S.-Y. (2007). *Structural Equation Modeling: A Bayesian Approach*. Wiley.

Lesage, J. P. (2004). A family of geographically weighted regression models. *Advances in spatial econometrics. Methodology, tools and applications.*, pages 241–264.

LeSage, J. P. (2008). An Introduction to Spatial Econometrics. *Revue d'économie industrielle*, 123(123):19–44.

Liang, F., Liu, C., and Carroll, R. J. (2007). Stochastic Approximation in Monte Carlo Computation. *Journal of the American Statistical Association*, 102(477):305–320.

Lin, D. Y. (2007). On the Breslow estimator. *Lifetime Data Analysis*, 13(4):471–480.

Lin, X. and Carroll, R. J. (2000). Nonparametric function estimation for clustered data when the predictor is measured without/with error. *Journal of the American Statistical Association*, 95(450):520–534.

MacKay, D. J. C. (2005). *Information Theory, Inference, and Learning Algorithms David J.C. MacKay*. Cambridge University Press, Cambridge.

Maia, T. V. and Frank, M. J. (2011). From reinforcement learning models to psychiatric and neurological disorders. *Nature neuroscience*, 14(2):154–162.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). Multivariate Analysis. *Analysis*, 97(9):1–4.

Marr, D. C. and Poggio, T. (1976). From Understanding Computation to Understanding Neural Circuitry. Technical report.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.

Mathers, C. D. and Loncar, D. (2006). Projections of global mortality and burden of disease from 2002 to 2030. *PLoS medicine*, 3(11):e442.

McArdle, J. J. (1988). Dynamic but structural equation modeling of repeated measures data. In *Handbook of multivariate experimental psychology*, pages 561–614. Springer.

McMahon, F. J. (2014). Prediction of treatment outcomes in psychiatry-where do we stand? *Dialogues in Clinical Neuroscience*, 16(4):455–464.

Melkumyan, A. and Ramos, F. (2011). Multi-kernel Gaussian processes. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1408–1413.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209(441458):415–446.

Merikangas, K. R., Mehta, R. L., Molnar, B. E., Walters, E. E., Swendsen, J. D., Aguilar-Gaziola, S., Bijl, R., Borges, G., Caraveo-Anduaga, J. J., Dewit, D. J., Kolody, B., Vega, W. A., Wittchen, H. U., and Kessler, R. C. (1998). Comorbidity of substance use disorders with mood and anxiety disorders: Results of the international Consortium in Psychiatric Epidemiology. *Addictive Behaviors*, 23(6):893–907.

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal Kernels. *Journal of Machine Learning Research*, 7:2651–2667.

Montague, P. R., Dolan, R. J., Friston, K. J., and Dayan, P. (2012). Computational psychiatry. *Trends in Cognitive Sciences*, 16(1):72–80.

Mostert, J. C., Hoogman, M., Onnink, A. M. H., van Rooij, D., von Rhein, D., van Hulzen, K. J. E., Dammers, J., Kan, C. C., Buitelaar, J. K., Norris, D. G., and Franke, B. (2015). Similar Subgroups Based on Cognitive Performance Parse Heterogeneity in Adults With ADHD and Healthy Controls. *Journal of Attention Disorders*.

Moya, J. G., Leibfried, F., Genewein, T., and Braun, D. A. (2016). Planning with information-processing constraints and model uncertainty in markov decision processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 475–491. Springer International Publishing.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Nemeroff, C. B., Heim, C. M., Thase, M. E., Klein, D. N., Rush, A. J., Schatzberg, A. F., Ninan, P. T., McCullough, J. P., Weiss, P. M., Dunner, D. L., Rothbaum, B. O., Kornstein, S., Keitner, G., and Keller, M. B. (2003). Differential responses to psychotherapy versus pharmacotherapy in patients with chronic forms of major depression and childhood trauma. *Proceedings of the National Academy of Sciences*, 100(24):14293–14296.

Nierenberg, A. A., Husain, M., Trivedi, M., Fava, M., Warden, D., Wisniewski, S., Miyahara, S., and Rush, A. (2010). Residual symptoms after remission of major depressive disorder with citalopram and risk of relapse: a STAR*D report. *Psychol. Med.*, 40(1):41–50.

Nolen-Hoeksema, S., Wisco, B. E., and Lyubomirsky, S. (2008). Rethinking Rumination. *Perspectives on Psychological Science*, 3(5):400–424.

Olbert, C. M., Gala, G. J., and Tupler, L. A. (2014). Quantifying heterogeneity attributable to polythetic diagnostic criteria: Theoretical framework and empirical application. *Journal of Abnormal Psychology*, 123(2):452–462.

Omar, R. Z., Wright, E. M., Turner, R. M., and Thompson, S. G. (1999). Analysing repeated measurements data: A practical comparison of methods. *Statistics in Medicine*, 18(13):1587–1603.

Park, B. U., Mammen, E., Lee, Y. K., and Lee, E. R. (2013). Varying coefficient regression models: A review and new developments. *International Statistical Review*, 83(1):36–64.

Paulus, M. P. (2015). Pragmatism instead of mechanism: A call for impactful biological psychiatry.

Perlis, R. H. (2016). Abandoning personalization to get to precision in the pharmacotherapy of depression. *World Psychiatry*, 15(3):228–235.

Perlis, R. H., Brown, E., Baker, R. W., and Nierenberg, A. A. (2006). Clinical features of bipolar depression versus major depressive disorder in large multicenter trials. *American Journal of Psychiatry*, 163(2):225–231.

Petzschner, F. H., Weber, L. A., Gard, T., and Stephan, K. E. (2017). Computational Psychosomatics and Computational Psychiatry: Toward a Joint Framework for Differential Diagnosis.

Pfeiffer, B. E. and Foster, D. J. (2013). Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):1–8.

Preacher, K. J. (2008). *Latent growth curve modeling*. SAGE Publications Inc.

Quitkin, F. M., Rabkin, J. G., Ross, D., and Stewart, J. W. (1984). Identification of true drug response to antidepressants: use of pattern analysis. *Archives of general psychiatry*, 41(8):782–786.

Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(January):4–16.

Rao, C. R. (1958). Some statistical methods for comparison of growth curves. *Biometrics*, 14(1):1–17.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Raudenbush, S. W. and Bryk, A. S. (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. SAGE Publications Inc.

Reinsel, G. (1984). Estimation and Prediction in a Multivariate Random Effects Generalized Linear Model. *Journal of the American Statistical Association*, 79(386):406–414.

Renz, D., Perez-Cruz, F., Stephan, K., and Huys, Q. J. (2018). Trajectory Estimation Using Gaussian Process Priors Over Feature Influence Across Time. *arxiv*, Submitted.

Rhebergen, D., Lamers, F., Spijker, J., De Graaf, R., Beekman, A. T. F., and Penninx, B. W. J. H. (2012). Course trajectories of unipolar depressive disorders identified by latent class growth analysis. *Psychological Medicine*, 42(7):1383–1396.

Roiser, J. P., Elliott, R., and Sahakian, B. J. (2012). Cognitive mechanisms of treatment in depression. *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology*, 37(1):117–36.

Rolls, E. T. (2005). *Emotion explained*. Oxford University Press.

Rossi, P. E., Allenby, G. M., and McCulloch, R. (2006). *Bayesian Statistics and Marketing*. Wiley.

Rothman, K. J. and Greenland, S. (2005). Causation and causal inference in epidemiology.

Rush, A. J. (2015). Narrowing the gaps between what we know and what we do in psychiatry. *Journal of Clinical Psychiatry*, 76(10):1366–1372.

Rush, A. J., Trivedi, M. H., Ibrahim, H. M., Carmody, T. J., Arnow, B., Klein, D. N., Markowitz, J. C., Ninan, P. T., Kornstein, S., Manber, R., Thase, M. E., Kocsis, J. H., and Keller, M. B. (2003). The 16-item Quick Inventory of Depressive Symptomatology (QIDS), clinician rating (QIDS-C), and self-report (QIDS-SR): A psychometric evaluation in patients with chronic major depression. *Biological Psychiatry*, 54(5):573–583.

Rush, a. J., Trivedi, M. H., Wisniewski, S. R., Nierenberg, A. a., Stewart, J. W., Warden, D., Niederehe, G., Thase, M. E., Lavori, P. W., Lebowitz, B. D., McGrath, P. J., Rosenbaum, J. F., Sackeim, H. a., Kupfer, D. J., Luther, J., and Fava, M. (2006). Acute and longer-term outcomes in depressed outpatients requiring one or several treatment steps: a STAR*D report. *The American journal of psychiatry*, 163(11):1905–17.

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach (3rd edition)*. Pearson Education.

Russell, S. and Wefald, E. (1991). *Do the right thing*. The MIT Press.

Russell, S. J. (1997). Rationality and intelligence. *Artificial Intelligence*, 94(1-2):57–77.

Schafer, J. L. and Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147–177.

Schatzberg, A. F., DeBattista, C., Lazzeroni, L. C., Etkin, A., Murphy, G. M., and Williams, L. M. (2015). ABCB1 genetic effects on antidepressant outcomes: A report from the iSPOT-D trial. *American Journal of Psychiatry*, 172(8):751–759.

Scherer, K. R. (2005). What are emotion? And how can they be measured? *Social Science Information, Sage Publications*, 44(4):695–729.

Schmaal, L., Marquand, A. F., Rhebergen, D., Van Tol, M. J., Ruhé, H. G., Van Der Wee, N. J., Veltman, D. J., and Penninx, B. W. (2015). Predicting the Naturalistic Course of Major Depressive Disorder Using Clinical and Multimodal Neuroimaging Information: A Multivariate Pattern Recognition Study. *Biological Psychiatry*, 78(4):278–286.

Schmittmann, V. D., Cramer, A. O., Waldorp, L. J., Epskamp, S., Kievit, R. A., and Borsboom, D. (2013). Deconstructing the construct: A network perspective on psychological phenomena. *New Ideas in Psychology*, 31(1):43–53.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138.

Singh, I. and Rose, N. (2009). Biomarkers in psychiatry.

Skrondal, A. and Rabe-Hesketh, S. (2007). Latent variable modelling: A survey. *Scandinavian Journal of Statistics*, 34(4):712–745.

Smith, C. A., Haynes, K. N., Lazarus, R. S., and Pope, L. K. (1993). In search of the" hot" cognitions: attributions, appraisals, and their relation to emotion. *Journal of Personality and Social Psychology*, 65(5):916–929.

Southwick, S. M. and Charney, D. S. (2012). The science of resilience: implications for the prevention and treatment of depression. *Science (New York, N.Y.)*, 338(6103):79–82.

Stephan, K. E., Schlagenhauf, F., Huys, Q. J. M., Raman, S., Aponte, E. A., Brodersen, K. H., Rigoux, L., Moran, R. J., Daunizeau, J., Dolan, R. J., Friston, K. J., and Heinz, A. (2017). Computational neuroimaging strategies for single patient predictions. *NeuroImage*, 145(Pt B):180–199.

Sun, H., Lui, S., Yao, L., Deng, W., Xiao, Y., Zhang, W., Huang, X., Hu, J., Bi, F., Li, T., Sweeney, J. A., and Gong, Q. (2015). Two patterns of white matter abnormalities in medication-naive patients with first-episode schizophrenia revealed by diffusion tensor imaging and cluster analysis. *JAMA Psychiatry*, 72(7):678–686.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT press.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211.

Tabachnick, B. G. and Fidell, L. S. (2012). *Using multivariate statistics*. Pearson.

Tobin, J. (1958). Estimation of Relationships for Limited Dependent Variables. *Econometrica*, 26(1):24–36.

Torous, J. and Baker, J. T. (2016). Why psychiatry needs data science and data science needs psychiatry connecting with technology.

Treadway, M. T. and Leonard, C. V. (2016). Isolating biomarkers for symptomatic states: Considering symptom-substrate chronometry.

Treynor, W., Gonzalez, R., and Nolen-Hoeksema, S. (2003). Rumination reconsidered: A psychometric analysis. *Cognitive Therapy and Research*, 27(3):247–259.

Trivedi, M. H. (2013). Modeling predictors, moderators and mediators of treatment outcome and resistance in depression.

Trivedi, M. H., McGrath, P. J., Fava, M., Parsey, R. V., Kurian, B. T., Phillips, M. L., Oquendo, M. A., Bruder, G., Pizzagalli, D., Toups, M., Cooper, C., Adams, P., Weyandt, S., Morris, D. W., Grannemann, B. D., Ogden, R. T., Buckner, R., McInnis, M., Kraemer, H. C., Petkova, E., Carmody, T. J., and Weissman, M. M. (2016). Establishing moderators and biosignatures of antidepressant response in clinical care (EMBARC): Rationale and design.

van Loo, H. M., Cai, T., Gruber, M. J., Li, J., De Jonge, P., Petukhova, M., Rose, S., Sampson, N. A., Schoevers, R. A., Wardenaar, K. J., Wilcox, M. A., Al-Hamzawi, A. O., Andrade, L. H., Bromet, E. J., Bunting, B., Fayyad, J., Florescu, S. E., Gureje, O., Hu, C., Huang, Y., Levinson, D., Medina-Mora, M. E., Nakane, Y., Posada-Villa, J., Scott, K. M., Xavier, M., Zarkov, Z., and Kessler, R. C. (2014). Major depressive disorder subtypes to predict long-term course. *Depression and Anxiety*, 31(9):765–777.

van Loo, H. M., de Jonge, P., Romeijn, J.-W., Kessler, R. C., and Schoevers, R. A. (2012). Data-driven subtypes of major depressive disorder: a systematic review. *BMC Medicine*, 10(1):156.

van Reekum, C. M. and Scherer, K. R. (1997). Chapter 6 Levels of processing in emotion-antecedent appraisal. *Advances in Psychology*, 124(C):259–300.

Veatch, O. J., Veenstra-Vanderweele, J., Potter, M., Pericak-Vance, M. A., and Haines, J. L. (2014). Genetically meaningful phenotypic subgroups in autism spectrum disorders. *Genes, Brain and Behavior*, 13(3):276–285.

Viguera, A. C., Baldessarini, R. J., and Friedberg, J. (1998). Discontinuing antidepressant treatment in major depression. *Harv.Rev.Psychiatry*, 5(1067-3229 (Print)):293–306.

Wan Lee, S., Shimojo, S., and O'Doherty, J. P. (2014). Neural Computations Underlying Arbitration between Model-Based and Model-free Learning. *Neuron*, 81(3):687–699.

Wang, C. and Neal, R. (2012). Gaussian Process Regression with Heteroscedastic or Non-Gaussian Residuals. *arXiv preprint arXiv:1212.6246*, pages 1–19.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference.* Springer Science & Business Media.

Weiskopf, D. A. (2011). Models and mechanisms in psychological explanation. *Synthese*, 183(3):313–338.

Wheeler, D. C. and Calder, C. A. (2006). *Bayesian Spatially Varying Coefficient Models in the Presence of Collinearity.* American Statistical Association.

Wiener, N. (1948). Cybernetics. *Scientific American*, 179:14–18.

Willet, J. B. and Sayer, A. G. (1994). Using covariance structure analysis to detect correlates and predictors of change over time. *Quantitative Methods in Psychology*, 116(2):363–381.

Williams, C. K. I. and Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8:514–520.

Wishart, J. (1938). Growth-rate determinations in nutrition studies with the bacon pig, and their analysis. *Biometrika*, 30(1/2):16–28.

Wu, C. O. and Yu, K. F. (2002). Nonparametric varying-coefficient models for the analysis of longitudinal data. *International Statistical Review*, 70(3):373–393.

Yu, Y. (2011). Structural Properties of Bayesian Bandits with Exponential Family Distributions. *arXiv preprint arXiv:1103.3089*, pages 1–19.

Zbozinek, T. D., Rose, R. D., Wolitzky-Taylor, K. B., Sherbourne, C., Sullivan, G., Stein, M. B., Roy-Byrne, P. P., and Craske, M. G. (2012). Diagnostic overlap of generalized anxiety disorder and major depressive disorder in a primary care sample. *Depression and Anxiety*, 29(12):1065–1071.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.