A flexible Bayesian approach to modeling change across time, individuals, and groups

Timothy Ballard[a], Hector Palada[a], Mark Griffin[b], & Andrew Neal[a]

[a]The University of Queensland

[b]Curtin University

Author Note

T. B. was supported by an ARC Discovery Early Career Research Award (DE180101340). H. P. was supported by an Australian Government Research Training Program Scholarship. A. N. was supported by an ARC Discovery Project (DP150102658). The data and code necessary to conduct all the analyses presented in this paper are publicly available, and can be found at https://osf.io/4euhr/

Abstract

Although some of the most highly influential theories in organizational psychology and organizational behavior explicitly describe a dynamic process, testing dynamic theory is difficult. Standard statistical approaches often do not accurately represent the dynamic process described by the theory. As such, there is often a misalignment between theory and statistical analysis that makes theories difficult to falsify. We demonstrate a flexible Bayesian approach to developing and analyzing dynamic models that overcomes many challenges associated with conventional methods. This approach can be used to analyze models of virtually any functional form, including models with feedback loops and dynamic (i.e., stock or level) variables. It also allows one to quantify uncertainty in components of a dynamic process. Finally, this approach provides a natural way to examine variation in a process between individuals, known groups, or latent subgroups. This framework has the flexibility to capture the dynamism inherent in many theories, which we believe will facilitate theory testing, and ultimately, cumulative theoretical progress.

Keywords: Dynamic models | Latent change | Bayesian inference | Theory testing | Computational modeling

Many organizational phenomena are dynamic in nature, meaning that they evolve over time (Kozlowski, Chao, Grand, Braun, & Kuljanin, 2013; Lord, Diefendorff, Schmidt, & Hall, 2010; Neal, Ballard, & Vancouver, 2017; M. Wang, Zhou, & Zhang, 2016). Examples include individual motivation and performance, team cohesion and effectiveness, and organizational culture. There are many theories that seek to describe the processes that produce changes in these phenomena. Examples include Self Regulation Theory (Carver & Scheier, 1998), Conservation of Resources Theory (Hobfoll, 1989), the Reoccurring Phase Model of team processes (Marks & Mathieu, 2001), and the Cultural Dynamics Model (Hatch, 1993). However, dynamic theories are difficult to test. This is because the processes described by these theories can be difficult to directly observe or measure, and there may be a mixture of different processes involved that unfold in different ways for different people, or in different contexts. To further complicate matters, there may be different processes acting at different levels, that unfold over different time scales.

To test a dynamic theory, a researcher needs a temporal design that allows for the theoretical processes to be observed as they unfold over time, and a statistical model that operationalizes the processes described by that theory (Collins, 2006). Intensive longitudinal designs are becoming more widely used, and a range of statistical models have been used to analyze the data collected using these designs. Common examples include multilevel regression, latent growth curve models, and latent change score models. However, it can be difficult to test a dynamic theory using these models. As noted by Deshon (2012), multilevel and latent growth models are poorly suited for the test of a dynamic theory, because they do not describe the process by which the variables change over time. Latent change score models can be used to test certain types of dynamic theories, but their implementation is complex, and this complexity increases dramatically as the number of variables and time points increase. This can make an even relatively simple conceptual model prohibitively difficult to specify (Deshon, 2012; M. Wang et al., 2016).

In this paper, we propose a Bayesian approach to modeling dynamic processes that is more intuitive, flexible, and comprehensive than the common extensions of the SEM framework. Under this approach, there is virtually no limit to the form of the model that can

be specified. The flexibility of this approach makes it possible to construct statistical models that map more directly onto psychological theory. This enables a closer coupling between theory and model, allowing for a clearer test of the theory. In the next section, we describe some of the challenges associated with analyzing dynamic processes. After that, we demonstrate the Bayesian approach described above using data from a recent study to facilitate understanding. We begin by showing how this framework can be used to model the change process at the level of the sample, where one assumes the same process is operating for all participants. We then show how this framework can be extended to model variability across people, known groups (e.g., experimental conditions), or latent subgroups of participants.

Throughout the paper, we have aimed to keep the level of exposition accessible to the researcher who has some experience using methods such as multilevel or structural equation modeling, but who does not necessarily have any experience with platforms such as R or Stan, or with Bayesian analysis more generally. We have kept the mathematical equations to a minimum, using them only where we believed they are required. We believe that for a paper such as this to be useful to a non-methods oriented reader, there must be enough practical content for the reader to apply this framework to her or his own work. We have therefore presented the computer code required to specify every model that we demonstrate. The models are implemented in Stan, a program with which we expect many readers will be unfamiliar. So we have provided comprehensive descriptions of the code in text. We have also included all the data and code required to implement the models presented in the paper in the supplementary material, which can be downloaded from https://osf.io/4euhr/.

Understanding Dynamic Processes

A dynamic process is one in which the components of a system change over time (Neal et al., 2017; M. Wang et al., 2016). Such processes typically form a closed loop, meaning that the outputs of the system influence the inputs to the system at some future point in time. A simple example of a dynamic process is a team raising money for a charity. The input to the system is the amount of money raised each month. These inputs accumulate over time and the cumulative effect of these inputs are reflected by the total amount of money raised. One output

of the system is the amount of interest earned from the bank each month on the total amount. This output feeds back into the system because it is added to the total amount, which increases the interest earned in the future, and makes the total amount of money earned by the charity grow even faster.

An important feature of a dynamic process is that at least one variable has *memory*. This type of variable, which is often referred to as a "dynamic", "stock", or "level" variable retains its value over time. It can be added to or subtracted from, but it cannot take on new values that are implausible given its previous state (Vancouver, Tamanini, & Yoder, 2008; Weinhardt & Vancouver, 2012). In the example above, the amount of money in the bank is a dynamic variable. This variable might increase (e.g., after more donations are deposited) or decrease (e.g., after account fees are paid), but the current value of the variable always depends on its previous value.

Theorizing about a dynamic process requires a theory of change. A theory of change describes how the system evolves from one time point to the next. The theory also identifies the dynamic variables in the system and explains the rules governing how these variables change over time. There are many influential theories in organizational psychology and organizational behavior that do just that (e.g., Carver & Scheier, 1998; Hobfoll, 1989; Marks & Mathieu, 2001), and computational models are increasingly being used to formalize these theories and generate quantitative predictions (Ballard, Yeo, B. Vancouver, & Neal, 2017; Grand, Braun, Kuljanin, Kozlowski, & Chao, 2016; Vancouver, Weinhardt, & Schmidt, 2010).

Testing a dynamic theory requires translating the theory into a statistical model, and applying the model to the data. However, this process can be complicated by a number of factors. First, there is uncertainty in the parameters being estimated, arising from the fact that inferences are being made based on observations of a limited number of people over a limited amount of time. Second, there may be individual and/or group differences at play. Finally, dynamic variables are difficult to implement using standard statistical models. These complications can make it difficult to implement statistical models that accurately represent the process described by many dynamic theories, which ultimately compromises the ability of the model to test the theory.

A Flexible Bayesian Approach

In this paper, we demonstrate a Bayesian approach to modeling change processes that has the flexibility required to accurately represent the dynamism inherent in many theories. Bayesian methods have become more widely used in recent years, in part due to the emergence of Bayesian analogues of standard statistical tests, and open-source software that makes them easy to implement (e.g., JASP; JASP Team, 2018). However, the models required to analyze dynamic processes are typically more complex than these standard tests, and usually cannot be applied in an off-the-shelf manner. This makes it difficult to develop standard Bayesian implementations for models of dynamic processes. As a result, there has been relatively limited uptake in Bayesian methods for modeling dynamic phenomena within organizational psychology and organizational behavior.

The advantages of Bayesian analysis are numerous, and have been thoroughly discussed elsewhere (e.g., Dienes, 2008; Edwards, Lindman, & Savage, 1963; Kruschke, Aguinis, & Joo, 2012; Wagenmakers et al., 2018, among others). Here, we briefly address a few of these advantages that are likely to be particularly relevant when modeling dynamic processes. At its core, Bayesian inference is based on probability theory (Jaynes, 2003; Jeffreys, 1939). This connection to probability theory allows one to make probabilistic inferences (e.g., "there is a 99% probability that Variable X increases over time"). Such inferences are not possible within the frequentist framework, which only allows categorical decisions regarding whether or not to reject a null hypothesis. This ability to deliver probabilistic inferences also applies to competing sets of hypothesis or models (e.g., "the null hypothesis is more likely than the alternative given the data"). These kinds of probabilistic statements are ultimately the sorts of inferences that researchers seek to make, which is why many have cited the Bayesian paradigm as the ideal approach to scientific inference (e.g., Etz & Vandekerckhove, 2018; Joyce, 1998; Lindley, 1993).

A well-known feature of the Bayesian approach is the ability to incorporate prior information into the analysis. As we will demonstrate, the consideration of this information provides a natural way of developing hierarchical models that can capture variability between people and groups and reduce measurement error (also see Boehm, Marsam, Matzke, &

Wagenmakers, in press). Another advantage of Bayesian inference is access to information regarding the uncertainty in a parameter estimate. Conventional parameter estimation methods such as least-squares or maximum likelihood provide only a single point-estimate that represents the 'best guess' of the true parameter value, and a standard error for the estimate. A Bayesian analysis derives the full distribution on each parameter, known as the posterior distribution, which indicates the range of parameter values that are most probable given the prior information and the observed data. This information makes it easy to examine differences between people and groups in the way a dynamic process plays out over time.

Another benefit of Bayesian analysis that is perhaps less appreciated is its capacity to incorporate model complexity. Model complexity generally refers to the flexibility of a model and the diversity of possible observations for which it can account (Myung & Pitt, 1997; Vandekerckhove, Matzke, & Wagenmakers, 2015). It is often the case that a researcher will analyze longitudinal data by testing a series of competing models. The ability of each model to explain the data is typically quantified by evaluating goodness-of-fit relative to model complexity. Complexity is penalized because a more flexible model is less parsimonious, so there is a need to demonstrate that any increase in complexity is justified by an increase in the fit of the model to the data.

In conventional analyses, it is usually the case that complexity is defined based on the number of parameters. This is the case for commonly used indicies such as the AIC (Akaike, 1973), BIC, (Schwarz, 1978), and RMSEA (Steiger, 1990). These methods are problematic because they ignore complexity introduced by the functional form of the model or the size of the space of possible values that the model parameters can take on. Consider the case where two models have the same number of parameters, but Model 1 can only account for a positive relationship between two variables whereas Model 2 can account for both positive and negative relationships. In this case, Model 2 is more complex because it can fit a wider range of observations (see Myung & Pitt, 1997, for a comprehensive treatment of this topic). Failure to account for these additional sources of complexity can result in incorrect conclusions about the relative evidence for each model. A Bayesian analysis naturally accounts for these sources of complexity.

A final advantage of the Bayesian approach is that its validity is not contingent on the adherence to a pre-specified sampling plan. Classical significance tests generally require data to be collected according to a fixed procedure, for example, where the researcher specifies the target sample size ahead of time and stops collecting data when the target size is reached. Failure to adhere to the pre-specified sampling plan invalidates the statistical test (Schönbrodt, Wagenmakers, Zehetleitner, & Perugini, 2017; Wagenmakers, 2007). This requirement poses a serious problem for researchers in organizational psychology and organizational behavior who wish to analyze naturalistic data that may not have been collected for the purpose of conducting the statistical test. In these fields, it is often the case that data are obtained after the fact, and therefore usually do not meet the assumptions of traditional tests. This is not a problem in the Bayesian framework. Under a Bayesian approach, the question of how the data were obtained is irrelevant, because the validity of the analysis does not depend on the sampling plan (Lindley, 1993; Wagenmakers et al., 2018; Wagenmakers, Morey, & Lee, 2016).

Modeling a Single Group

In the sections that follow, we demonstrate the Bayesian framework described above. The aim of a Bayesian analysis is to determine the inferences that can be made in light of some empirical observation (Etz, Gronau, Dablander, Edelsbrunner, & Baribault, 2018; Kruschke et al., 2012; Lee & Wagenmakers, 2013). This involves combining information about the researcher's a priori beliefs regarding each parameter, referred to as its *prior distribution* (henceforth, prior), with information about the *likelihood* of the model given the data. This analysis results in a *posterior distribution* (henceforth, posterior) on each parameter, which represents the range of parameter values that are credible given the researcher's prior and the observed data. A highly dispersed posterior, which covers a broad range of possible values, suggests a high degree of uncertainty regarding the true parameter value. A narrow posterior, which covers only a small range of values, suggests more certainty.

Although the Bayesian approach is simple conceptually, the implementation of these methods is often complex. For most models, including the types of models likely to be of

interest in organizational psychology and organizational behavior, there is no closed-form analytic solution for calculating the posterior. The posterior must therefore be approximated using Markov chain Monte Carlo (MCMC) methods. MCMC methods refer to a class of algorithms that can be used to generate a large number of representative samples from the posterior distribution (see Kruschke, 2010; Van Ravenzwaaij, Cassey, & Brown, 2018). Broadly, these methods work by producing sequences, or chains, of sample parameter values that are generated by an algorithm that is known to approximate the posterior given a sufficient number of samples.

There are several open-source platforms for implementing Bayesian models using MCMC methods. These platforms include WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000), JAGS (Plummer, 2003), OpenBUGS (Thomas, O'Hara, Ligges, & Sturtz, 2006), and Stan (Carpenter et al., 2017). The advantage of programs such as these is that they allow the user a great deal of flexibility with regard to model specification, but the MCMC algorithm is implemented under the hood. This allows the user to focus on the development of the model itself. In this paper, we use the Stan platform. We use Stan because its MCMC algorithm, the No-U-Turn Sampler (Hoffman & Gelman, 2014), is particularly well suited for implementing more complex models with many parameters (e.g., hierarchical models; Stan Development Team, 2017).

To facilitate understanding, we use real data from a recent study by Gee, Neal, and Vancouver (2018). In their study, 60 participants each performed ten 10-minute trials of an air traffic control simulation task. The task required participants to classify aircraft pairs as a conflict or non-conflict based on their minimum distance of separation. Before each trial, participants set a goal regarding their performance on the upcoming trial. The aim of the study was to examine how people revised their goals over time.

Gee et al. proposed a formal model of goal revision, which describes the dynamic process by which people revise their goals in response to previous performances. Throughout the remainder of the paper, we use a version of this model as a running example to facilitate understanding. The model predicts that a person changes their goal based on the discrepancy between their previous goal and previous performance. This process can be expressed

formally as follows:

$$G_t = G_{t-1} + \alpha(G_{t-1} - P_{t-1}) + \beta, \tag{1}$$

where G is the goal level, and P is the actual performance. The α parameter represents the learning rate. Higher values of α mean that goal revision is more responsive to discrepancy between previous goal and previous performance. The β parameter represents the component of goal revision that is not sensitive to the discrepancy.

Testing this goal revision model requires a highly flexible statistical framework. To adequately represent this model, goal level must be treated as a dynamic variable that retains a "memory" of its previous state. This type of dynamic process is difficult to represent using standard statistical approaches. However, they are straightforward to implement within the Bayesian framework we demonstrate below. Although, we focus on this model in the examples that follow, the approach we present is highly general. It can be used to implement models with virtually any functional form, which makes it easy to implement highly customized models that capture relatively nuanced theoretical assumptions. However, this approach can also be used to implement Bayesian versions of more standard models of change such as latent change score, cross-lagged panel, latent growth curve, and multilevel regression models.

Sample-level Model

In this section, we show how to implement the goal revision model in Stan. The model has two theoretically meaningful parameters: α and β. To implement the goal revision model as a statistical model, a third parameter is required that represents the residual standard deviation of the goal level (σ). The σ parameter represents the variability in the observed goal level that is not explained by the predicted goal level. We begin by estimating these parameters at the sample level. In other words, we describe the behavior of the entire sample using a single set of parameters. The sample-level model assumes that the goal revision process plays out in the same way for each participant. In later sections, we extend this framework to construct more sophisticated models of inter-individual and inter-group variation.

The top-left panel in Figure 1 shows the model depicted as a graphical plate model. Graphical plate models are useful to illustrate the dependencies between the variables within model. The variables are represented as nodes. Shaded nodes reflect variables observed from the data, whereas unshaded nodes reflect latent parameters estimated by the model. The rectangular plates represent the sources of variability in the model. The goal and performance variables (denoted G and P respectively) are inside both plates, which indicates that they can vary across people and time. The three estimated parameters are outside both plates, which indicates that they do not vary. The arrows drawn between the nodes highlight the dependencies between variables. The precise nature of these relationships is given in Equation 1, but the arrows provide a useful summary. The straight arrows originating from P, α, β, and σ, indicate that G is in part determined by these variables. The arrow that loops from G back into itself indicates that G is also influenced by itself. The predicted value of G at a certain time feeds back to influence the predicted value at the next time point. In other words, G is a dynamic variable.

Like programs such as Mplus or Stata, Stan is a syntax-based platform, which means that the model must be expressed via computer code. The code required to specify the sample-level model is presented below:

```
data {
  int Ntotal;
  real trial[Ntotal];
  real observed_goal[Ntotal];
  real performance[Ntotal];
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}

model {
  //initialize predicted goal level
  real predicted_goal;

  //priors
  alpha ~ normal(0,1);
```

```
20    beta ~ normal(0,1);
21    sigma ~ normal(0,1);
22
23    //likelihood
24    for(i in 1:Ntotal){
25      if(trial[i]==1){
26        predicted_goal = observed_goal[i];
27      }
28      if(trial[i]>1){
29        predicted_goal += alpha*(performance[i-1]-predicted_goal) + beta;
30      }
31      observed_goal[i] ~ normal(predicted_goal,sigma);
32    }
33  }
```

As can be seen, the program requires three unique blocks of code. The first is the `data` block. Here, the user must declare the data to be used as inputs to the model. The `Ntotal` variable is a single value that indicates the total number of data points. In this dataset, `Ntotal` $= 600$ (60 participants $\times$ 10 trials). The `trial`, `observed_goal`, `performance` variables are columns of values representing the trial number, the goal set by the participant at the start of that trial, and the performance on the trial respectively. The `[Ntotal]` after variable name indicates that the number of data points in the variable is equal to `Ntotal`. Note that Stan requires the user to declare an object type for each input variable. In the above, the `int` identifier is used to declare `Ntotal` as an integer. The other variables are declared as `real`, indicating that they can take on any real value.

The second block is the `parameters` block. Here, the user must declare the parameters that are to be estimated. In this model, `alpha`, `beta`, and `sigma` indicate α, β, and σ respectively. All three parameters are declared as `real` which means they can take on any real value. The `sigma` parameter is declared with the `<lower=0>` constraint, which imposes a lower bound of 0 on this parameters (which is necessary because a standard deviation, by definition, must be positive).

The third block is the `model` block. This is where the details of the model implementation are specified. Line 16 initializes a new variable called `predicted_goal`. This variable will be used to store the goal level predicted by the model. Lines 19-21 specify the priors for the three parameters. The ~ operator is used to define a variable or parameter that

has a distribution, and can be read as *has a distribution of*. When an unknown parameter (as opposed to a known variable) is specified on the left side of ~ operator, the distribution given on the right side becomes the prior for that parameter. The `alpha` and `beta` parameters are assigned parameters normally distributed priors with a mean of 0 and standard deviation of 1. In the context of this dataset, these are *uninformative* priors, which are priors that do not place a high degree of prior belief on any particular region of the parameter space. The `sigma` parameter is also assigned a normally distributed prior with the same mean and standard deviation. However, because we imposed a lower bound of 0 on `sigma` in the `parameters` block, the algorithm will only sample positive values from this distribution.

The rest of the `model` block is where the model itself is defined. By "model", we mean the sequence of operations that are required to determine the likelihood of the data given the sampled parameter values. As can be seen, the model is constructed using a for-loop. A for-loop is a programming tool that enables a section of code to be executed repeatedly, with what is known as the *looping variable* taking on a different value in each execution. In the model above, `for(i in 1:Ntotal)` initializes a for-loop that iterates through the series of consecutive integers from 1 to `Ntotal`. The looping variable, `i`, is reassigned with each execution of the loop. In the first execution, $\texttt{i} = 1$; in the second, $\texttt{i} = 2$; and so on. The for loop therefore iterates through data point, performing a series of operations on each one.

For each data point, the model first calculates the predicted goal level and assigns the prediction to the `predicted_goal` variable. For each participants' first trial, there is no previous predicted goal level. In this case, the predicted goal is assumed to be equal to the observed goal level (Gee et al., 2018). The predicted goal level on trial 1 is specified on lines 25-27. Line 25 contains an *if statement*, which carries out a particular operation only if a condition is met. In this case, if the value of `trial` the for data point i is equal to 1, the code on line 26 will be executed. If the condition is not met, line 26 will be skipped.

On lines 28-30, a separate if statement is used to specify the predicted goal level for trials 2 through 10. In this case, the code on line 29 will be executed only if the value of `trial` for data point i is greater than 1. Line 29 specifies the change in the predicted goal level to be determined according to Equation 1. The `+=` operator treats `predicted_goal` as a dynamic

variable. This operator changes the current value of the variable by the amount given on the right hand side of the expression (e.g., if $x = 2$, then `x += 3` will change the value of x to 5).

As a final step, the model compares the predicted goal level to the observed goal level. On line 31, `observed_goal` is treated as a dependent variable. When a known variable (as opposed to an estimated parameter) is specified on the left side of the ~ operator, it is treated as an outcome variable, with the arguments on the right side representing the predicted distribution of that variable. On line 31, we specify the value of `observed_goal` for data point i to be normally distributed with a mean equal to `predicted_goal` and standard deviation equal to `sigma`. The algorithm will therefore evaluate the likelihood of the observed goal given the current values of these variables, and update the posterior accordingly. In other words, we are effectively regressing the observed goal on the predicted goal whilst fixing the intercept to 0 and the slope to 1, with `sigma` representing the residual standard deviation of the observed goal.

To run the model, the user can call Stan using R, Stata, MATLAB, Python, Julia, or the command line. For these analyses, we run Stan via the RStan package (Stan Development Team, 2016), which provides an R interface to Stan (see supplementary materials for R code used to run the model). Stan returns an object that contains samples from the posterior distribution for each parameter in the model. The RStan package provides a summary of each posterior, which is shown below. The summary contains the posterior mean, the standard error of that mean, and the standard deviation of the posterior. It also displays the 2.5%, 25%, 50%, 75%, and 97.5% quantiles of the distribution.

```
1          mean se_mean   sd    2.5%     25%     50%     75%   97.5% n_eff Rhat
2 alpha    0.51    0.00 0.03    0.45    0.49    0.51    0.53    0.57  3666    1
3 beta     0.17    0.00 0.03    0.11    0.15    0.17    0.20    0.24  3680    1
4 sigma    1.29    0.00 0.04    1.22    1.26    1.29    1.32    1.37  4000    1
```

The output also provides some information to help assess model convergence. The number of effective samples (`n_eff`) is a measure of the information contained in the posterior samples that accounts for the autocorrelation between neighboring samples in each MCMC chain. MCMC chains tend to be autocorrelated because each sample in part depends on the previous sample. The number of effective samples represents the number of independent

samples that would produce an equivalent amount of information to the set of samples obtained. If there is so little autocorrelation that samples are effectively independent, the number of effective samples will be approximately equal to the total number of samples (4000 in this example, which is the RStan default). This is the case for `sigma` in the above model. For `alpha` and `beta`, the number of effective samples is fewer than the actual number of samples, indicating the information contained in the posterior is less than it would be if the samples were completely independent. This is not necessarily problematic. However, it is important to ensure the effective sample size is large enough that the approximated posterior will be representative of the underlying distribution. Here, the effective sample sizes are sufficiently large that we can be confident in the obtained posteriors. The final piece of information provided is the potential scale reduction factor (`Rhat`; Brooks & Gelman, 1998; Gelman & Rubin, 1992). This gives an indication of the extent to which the chains converged on the same region of the parameter space. As the chains approach convergence, `Rhat` approaches one. A common rule of thumb is that `Rhat` should be less than 1.1 before any inferences are made based on the posteriors (Kruschke et al., 2012).

The information regarding the parameter posteriors can easily be broken down for more detailed exploration. It is good practice to examine the full posterior distribution on each parameter, as well as the joint posterior distribution between pairs of parameters. To illustrate, the top row of Figure 2 contains the posteriors on the `alpha` and `beta` parameters. The first two columns show an approximation of the full posterior distribution for each parameter. As can be seen in the top-left panel, the most probable values for `alpha` lie within the range of 0.4 to 0.6, suggesting that learning rates within that range are more probable. As can be seen in the top-middle panel, the most probable values for `beta` lie within the range of about 0.05 to 0.3. This suggests that people raise increase their goal by an extra–although likely small–amount regardless of the discrepancy between previous goal and previous performance. The top-right panel in Figure 2 shows the joint posterior on `alpha` and `beta`. In this plot, the inner most contours represent the most probable parameter values. The joint posterior can be useful for examining the correlation between parameters and for diagnosing model misspecification. Here, `alpha` and `beta` are largely independent.

Once the researcher is satisfied that the model has converged and that the parameters are well estimated, they will likely wish to systematically determine the most probable values for each parameter. One commonly used way of summarizing the most probable parameter values is by calculating the 95% *credible interval* (CI). The 95% CI spans the 2.5% and 97.5% quantiles of the posterior distribution. In the above model, the CI on `alpha` is 0.45 to 0.57, and the CI on `beta` is 0.11 to 0.24. The 95% *highest density interval* (HDI) is another commonly used interval for summarizing the posterior distribution. The 95% HDI represents the shortest possible interval containing 95% of the posterior density, and will therefore always include the most probable parameter values (Kruschke et al., 2012). The CI and HDI each has advantages and disadvantages (e.g., see Kruschke, 2010), but for symmetrical distributions the two intervals will be very similar.

The CI or HDI is often used to decide whether particular parameter values should be accepted or rejected. For example, intervals that exclude zero are often taken as evidence that the parameter value is different from zero (Kruschke et al., 2012). Another simple way to quantify the evidence for a parameter being either greater than or less than a particular value is to determine the percentage of the posterior density that is above or below that value (Kruschke, 2010). For example, 100% of the posterior distributions on both `alpha` and `beta` are above 0. This provides strong evidence that both of these parameters are positive. It should be noted that some have argued against making inferences based only on a parameter's posterior distribution and instead have advocated for inferences based on model comparison (e.g., Rouder, Morey, Verhagen, Province, & Wagenmakers, 2016). We return to the issue of model comparison later in the paper.

Although estimating parameters at the sample-level can be informative, this approach can only be used to examine average effects. In this model, the `alpha`, `beta`, and `sigma` parameters represent average parameter values across participants in the sample. These parameters tell us nothing about how these components vary across individuals. To do this, we require a model in which unique parameters are estimated for each person. We present such a model in the next section.

Person-level Model

Whereas making inferences purely at the sample-level can be useful, there are likely to be instances where one wishes to examine effects at the level of the individual participant. For example, one might assume that a process plays out differently for different individuals and seek to quantify the components of this process separately for each person. The model presented above can be easily extended to model processes at the person level. To illustrate, the top-right panel of Figure 1 shows the graphical representation of a model that estimates unique `alpha` and `beta` parameters for each participant (whilst still estimating `sigma` at the sample level). As can be seen, the α and β nodes are inside of the person plate. This indicates that these parameters vary across individuals. The code below shows the specification of the person-level model.

```
data {
  int Ntotal;
  real trial[Ntotal];
  real observed_goal[Ntotal];
  real performance[Ntotal];
  int Nsubj;
  int subject[Ntotal];
}

parameters {
  real alpha[Nsubj];
  real beta[Nsubj];
  real<lower=0> sigma;
}

model {
  //initialise other variables
  real predicted_goal;

  //priors
  alpha ~ normal(0,1);
  beta ~ normal(0,1);
  sigma ~ normal(0,1);

  //likelihood
  for(i in 1:Ntotal){
    if(trial[i]==1){
```

```
28        predicted_goal = observed_goal[i];
29      }
30      if(trial[i]>1){
31        predicted_goal += alpha[subject[i]]*(performance[i-1]-predicted_goal) + beta[subject[i]];
32      }
33      observed_goal[i] ~ normal(predicted_goal,sigma);
34    }
35  }
```

Only three simple changes are required. First, two new variables have been added to the `data` block. `Nsubj` is a single value representing the number of participants (in this example, $Nsubj = 60$). `subject` is a column of integers representing the participant number. Second, in the `parameters` block, the clause `[Nsubj]` has been added after the declarations of the `alpha` and `beta` parameters (lines 11 and 12). This indicates that `alpha` and `beta` are now arrays (as opposed to scalar values) with lengths equal to the number of participants. Each element of the array corresponds to one participant, so there will be a unique parameter estimated for each one. The third change is in the `model` block on line 31. As can be seen, `[subject[i]]` has been added after `alpha` and `beta`. This index means that the predicted score will be calculated based on the `alpha` and `beta` parameters associated with the participant whose data is located in the i-th row of the dataset.

This model produces a set of posterior `alpha` and `beta` samples for each participant, and a single set of `sigma` samples. The middle row of Figure 2 shows the results for the `alpha` and `beta` parameters. The first two plots show the 95% credible intervals on `alpha` and `beta` for each participant. Note that it is possible to access the full posterior for each participant, but the credible interval is more visually compact and makes it easier to compare a larger number of participants. As can be seen, there is heterogeneity between participants in both `alpha` and `beta`. The right panel shows the `alpha` and `beta` parameters for each participant plotted against each other (with the crosses representing the credible intervals for each parameter).

Hierarchical Model

Whilst parameter estimation at the person level has advantages over the sample-level model, one problem with the person-level model is that it can be difficult to make inferences about the population from which participants are drawn. Analyzing a person-level model is

akin to running a separate, single-level analysis on each participant in a sample. This approach has less power because it only considers data from one participant at a time, and ignores the rest of the sample. It also fails to capture commonalities between individuals that may arise from from participants being members of the same population.

It is quite often the case that researchers wish to examine variation between participants, but also to make inferences about the population as a whole. A hierarchical modeling approach is extremely useful for this purpose (Kruschke & Vanpaemel, 2015; Turner et al., 2013; Vincent, 2016). Hierarchical Bayesian models allow researchers to "have their cake and eat it too" by modeling the individual and population levels simultaneously (Lewandowsky & Farrell, 2011). As with the person-level model, the hierarchical model estimates unique parameters for each individual. However, unlike the person-level model, the hierarchical approach also models the distribution of the person-level parameters at the population level. As a result, person-level parameters are informed not only by the data for that one individual, but also by the population-level distribution on the relevant parameter. This increases the accuracy of the parameter estimates (Boehm et al., in press; Rouder & Lu, 2005).

The bottom-left panel of Figure 1 shows the graphical representation of a model that estimates the α and β parameters hierarchically. As can be seen, this model assumes that α and β are each informed by two higher level parameters–μ and σ–which together describe the nature of the variability in these parameters at the population level. The code below shows the `parameters` and `model` blocks required to implement the hierarchical model (the `data` block in hierarchical model is identical to the `data` block in the person-level model).

```
parameters {
  real alpha[Nsubj];
  real beta[Nsubj];
  real<lower=0> sigma;
  real alpha_mean;
  real<lower=0>  alpha_sd;
  real beta_mean;
  real<lower=0>  beta_sd;
}

model {
  //initialise other variables
  real predicted_goal;
```

```
23
24    //priors
25    alpha ~ normal(alpha_mean,alpha_sd);
26    beta ~ normal(beta_mean,beta_sd);
27    sigma ~ normal(0,1);
28    alpha_mean  ~ normal(0,1);
29    alpha_sd  ~ normal(0,1);
30    beta_mean  ~ normal(0,1);
31    beta_sd  ~ normal(0,1);
32
33    //likelihood
34    for(i in 1:Ntotal){
35      if(trial[i]==1){
36        predicted_goal = observed_goal[i];
37      }
38      if(trial[i]>1){
39        predicted_goal += alpha[subject[i]]*(performance[i-1]-predicted_goal) + beta[subject[i]];
40      }
41      observed_goal[i] ~ normal(predicted_goal,sigma);
42    }
43  }
```

As can be seen, the hierarchical model requires four new parameters. The first two are `alpha_mean` and `alpha_sd` (lines 14 and 15), which represent the mean and standard deviation of `alpha`. These parameters characterize the distribution of `alpha` at the population level. The `beta_mean` and `beta_sd` parameters (lines 16 and 17) represent the mean and standard deviation of `beta`. These parameters characterize the distribution of `beta` at the population level.

As can be seen on lines 25-26, the priors on `alpha` and `beta` differ in the hierarchical model, compared to the other models we have thus far demonstrated. Whereas in the sample-level and person-level models these priors are set using fixed values that produced broad, uninformative distributions, in the hierarchical model these priors are set using the population-level means and standard deviations. In other words, the hierarchical model uses information about the population-level distribution as the prior for the person-level parameters. The parameters that define the population-level distributions–`alpha_mean`, `alpha_sd`, `beta_mean`, and `beta_sd`–are then given their own priors on lines 28-31. The priors on the parameters that define higher level distributions are commonly referred to as

hyperprior or *parent distributions*.

The above hierarchical model produces a unique set of posterior `alpha` and `beta` samples for participant, and single sets of samples for `alpha_mean`, `alpha_sd`, `beta_mean`, `beta_sd`, and `sigma`. The bottom row in Figure 2 shows a breakdown of the posteriors on the `alpha` and `beta` parameters. In the first two columns, the red lines represent the credible intervals on `alpha` and `beta` for each participant. The blue densities represent the full posterior on `alpha_mean` and `beta_mean`.

As can be seen, the posteriors on the population-level means in the hierarchical model occupy a similar region of the parameter space as the posteriors in the sample-level model (shown in the top row of the figure). The reader may note however, that the credible intervals on the person-level parameters are less dispersed in the hierarchical model than in the person-level model (shown in the middle row of the figure). This is because in the hierarchical model, the population-level distribution imposes an additional constraint on the person-level parameters, which pulls the person-level parameters closer to the sample mean. This process is called *shrinkage*, and it can also be seen in the bivariate plot in the bottom-right panel. Shrinkage reduces measurement error because it means that parameters that are less reliably estimated become more strongly influenced by the population mean (Boehm et al., in press). This also reduces the likelihood of outliers. For example, consider Participant 22 who was a slight outlier with respect to the `alpha` parameter. For this participant, the addition of the population distribution in the hierarchical model changed `alpha` by a considerable amount–from a posterior mean of -0.25 under the person-level model to 0.16 under the hierarchical model. Contrast the above with Participant 11, for whom `alpha` was fairly typical. For this participant, mean `alpha` changed by only a small amount–from a mean of 0.57 under the person-level model to 0.53 under the hierarchical model.

Modeling Multiple Groups

The models we have addressed thus far examine variation within a single group or population. However, many research questions require the consideration of multiple groups. For example, a researcher may want to examine whether a dynamic process differs across

experimental conditions. In this case, they need to examine whether the parameters of interest differ between two or more predefined groups. Alternatively, a researcher may wish to examine whether participants naturally cluster into distinct subgroups. For example, perhaps some for participants have a relatively high learning rate whereas others have a relatively low rate.

In this section, we describe two approaches that can help to answer the types of research questions described above. The first model examines whether parameters of interest differ among known groups. The second is a mixture model, which can be used to identify naturally emerging latent subgroups into which participants cluster. These models are summarized by the bottom-right panel of Figure 1. As can be seen, the α and β nodes are inside the outer plate, which here represents the group level. This diagram indicates that α and β vary across different groups of participants.

Known Group Membership

In the study that generated the example dataset, goal framing was manipulated between participants. For half of the participants, the goal was framed in approach terms. For these participants, the goal was expressed as the minimum number of correct decisions the participant needed to achieve. For the other half, the goal was framed in avoidance terms. In this case, the goal was expressed as as the maximum number of incorrect decisions the participant was allowed to make. In the next example model, we examine whether the goal revision processes differed between these groups by estimating unique `alpha` and `beta` parameters for each group and then comparing them. For simplicity, we allow parameters to vary across groups but not across people within each group. However, this model can also be implemented as hierarchical model, where parameters can vary across people and groups. We include code to implement the hierarchical version of this model in the supplementary material.

The code that specifies this model is shown below. As can be seen, there are three changes required to convert the sample-level model described above to the multiple-group model. First, one new variable needs to be added to the `data` block. The `condition` variable

(line 6) indicates the condition for each participant (1 = Approach, 2 = Avoidance). Second, in the `parameters` block, the `alpha` and `beta` parameters are declared as arrays that each contain two elements. This means that two `alpha` and `beta` parameters will be estimated, one for each condition. The final change is on line 30. As can be seen, the indexing statement `[condition[i]]` has been added to the `alpha` and `beta` parameters. This means that the change in predicted goal level will be calculated using the parameter values associated with the experimental condition relevant to the i-th row of the dataset.

```
1  data {
2    int Ntotal;
3    real trial[Ntotal];
4    real observed_goal[Ntotal];
5    real performance[Ntotal];
6    int condition[Ntotal];
7  }
8
9  parameters {
10   real alpha[2];
11   real beta[2];
12   real<lower=0> sigma;
13 }
14
15 model {
16   //initialise other variables
17   real predicted_goal;
18
19   //priors
20   alpha ~ normal(0,1);
21   beta ~ normal(0,1);
22   sigma ~ normal(0,1);
23
24   //likelihood
25   for(i in 1:Ntotal){
26     if(trial[i]==1){
27       predicted_goal = observed_goal[i];
28     }
29     if(trial[i]>1){
30       predicted_goal += alpha[condition[i]]*(performance[i-1]-predicted_goal) + beta[condition[i]];
31     }
32     observed_goal[i] ~ normal(predicted_goal,sigma);
33   }
34 }
```

The top row of Figure 3 shows the posteriors on the `alpha` (left panel) and `beta` (middle panel) parameters for the approach and avoidance conditions. As can be seen in the top-left panel, there is separation in the posterior distributions on `alpha` between the approach and avoidance conditions. This result suggests that people are more responsive to the discrepancy between previous goal and previous performance when pursuing approach goals compared to avoidance goals. As can be seen in the top-middle panel, there is also separation in the posteriors on `beta` between the two conditions. This suggests that the component of goal revision that is independent of the discrepancy is larger when pursuing approach goals compared to avoidance goals.

To make inferences about differences in parameter values between conditions, we need to examine the posterior distribution on the *difference* between parameters in the approach and avoidance conditions. This is done by calculating, for each posterior sample, a variable that is equal to the sampled parameter value for the approach condition minus the sampled value for the avoidance condition. This yields a difference score for each posterior sample, which can be used to approximate the posterior distribution of the difference score.

The posterior on the difference scores for each parameter can be seen in the top-right panel of Figure 3. As can be seen, both parameters have difference scores that are positive. The credible interval on the difference in `alpha` between approach and avoidance ranges from 0.09 to 0.33, and approximately 99.9% of this distribution is greater than 0. The credible interval on the difference in `beta` between approach and avoidance ranges from 0.12 to 0.36, and approximately 100% of this distribution is greater than 0. These results provide strong evidence that both parameters are positive.

Unknown Group Membership

In the above example, the researcher knew beforehand the group to which each participant belonged. However, this is not always the case. Sometimes, the researcher may wish to identify subgroups of participants who behave similarly, without any prior knowledge regarding group membership. This can be achieved using a mixture model. A mixture model is used to capture behavior that results from different processes (Bartlema, Lee, Wetzels, &

Vanpaemel, 2014). For example, there may be subgroup of participants who behave according to one process, and another subgroup whose behavior is governed by a different process. These different processes are referred to as *mixtures*. The goal of the analysis is to determine the parameters that best characterize each mixture, and to make inferences about the relative influence of each mixture on each participant's behavior.

The code below specifies a mixture model in which `alpha` and `beta` differ between mixtures. For simplicity, we model only two mixtures in this example. The supplementary materials contain an example of how this model can be generalized to any number of mixtures. The `data` block in the mixture model is identical to the `data` blocks in the person-level and hierarchical models. There are two changes required to the `parameters` block. As with the multiple-group model, this mixture model estimates two unique `alpha` and `beta` parameters. However, in the mixture model, one of these parameters must be declared as an ordered vector (line 12). The ordered vector is a special Stan object type that sorts the values within it in ascending order. This is needed for model identifiability reasons. In this model, we specify `beta` as an ordered vector.

The mixture model also includes an additional parameter (line 14). This parameter is the mixture weight (`mix_weight`), which indicates the relative influence of each mixture on the behavior of each participant. This weight ranges from 0 to 1, where higher values indicate a stronger influence of Mixture 1. The model estimates a unique mixture weight for each participant.

```
10 parameters {
11   real alpha[2];
12   ordered[2] beta;
13   real<lower=0> sigma;
14   real<lower=0,upper=1> mix_weight[Nsubj];
15 }
16
17 model {
18   //initialise other variables
19   real predicted_goal[2];
20
21   //priors
22   alpha ~ normal(0,1);
23   beta ~ normal(0,1);
```

```
24    sigma ~ normal(0,1);
25
26    //likelihood
27    for(i in 1:Ntotal){
28      if(trial[i]==1){
29        predicted_goal[1] = observed_goal[i];
30        predicted_goal[2] = observed_goal[i];
31      }
32      if(trial[i]>1){
33        predicted_goal[1] += alpha[1]*(performance[i-1]-predicted_goal[1]) + beta[1];
34        predicted_goal[2] += alpha[2]*(performance[i-1]-predicted_goal[2]) + beta[2];
35      }
36
37      target += log_mix(mix_weight[subject[i]],
38                        normal_lpdf(observed_goal[i] | predicted_goal[1], sigma),
39                        normal_lpdf(observed_goal[i] | predicted_goal[2], sigma));
40    }
41  }
```

As can be seen in the `model` block, no changes to the priors are required to implement the mixture model. The reader may note that we have not explicitly assigned a prior to `mix_weight`. This is because by default Stan imposes a uniform prior, which is appropriate for `mix_weight` because the parameter is bounded on both ends. As can be seen, there are some key differences in the likelihood component of the `model` block compared to the previous models we have demonstrated. First, the model now calculates two separate predicted goals. The first is calculated based on `alpha[1]` and `beta[1]`, which are the parameters associated with Mixture 1 (lines 29 and 33). The second is calculated based on the parameters associated with Mixture 2 (lines 30 and 34).

As can be seen on lines 37-39, the expression of the likelihood itself has also changed. The Stan syntax required to define the likelihood for a mixture model is more complex than the syntax used in the previously demonstrated models. To implement a mixture model in Stan, the `mix_weight` parameter must be marginalized out of the likelihood (Stan Development Team, 2017). This means that the likelihood of the observation given the model must be calculated based on the weighted sum of the likelihood of the data under each separate mixture [1].

[1]Formally, the likelihood is expressed as $p(y|\lambda,\mu,\sigma) = \sum_{k=1}^{K} \lambda_k \times Normal(y|\mu_k,\sigma_k)$ where k rep-

The bottom row of Figure 3 displays the results for the above mixture model. The left and middle panels show the differences in `alpha` and `beta` between the two mixtures identified by the model. As can be seen, the posteriors on `alpha` overlap considerably, suggesting that this parameter is very similar between the two mixtures. However, the posteriors on `beta` differ a great deal between the two mixtures. For Mixture 1, the most probable `beta` values are just below 0. For Mixture 2, the most probable `beta` values are just below 1. The bottom-right panel shows the credible intervals on the mixture weight for each participant. These results suggest that about a half of the participants are most strongly influenced by Mixture 1 (i.e., have mixture weights that are clearly above 0.5), and about a quarter are most strongly influenced by Mixture 2 (i.e., have mixture weights that are clearly below 0.5). For the remaining participants, the relative influence of each mixture is fairly balanced.

Before making inferences based on a mixture model such as the above, it is advisable to test it against alternative models that specify different numbers of mixtures (Bartlema et al., 2014). For example, we might compare the two-mixture model above to one-mixture and three-mixture models to identify which provides the best description of the data. The goal of the model comparison is to determine the number of mixtures that is most strongly favored by the evidence. We address the issue of model comparison in the next section.

Model Evaluation

Once the researcher has specified an appropriate model and confirmed that the model has converged, he or she will need to evaluate whether the model provides a satisfactory account of the data. Parameter estimates are only meaningful to the extent that the model is a good description of the phenomenon being investigated. If the model is a poor approximation

---

resents the mixture. In the above model, λ_1 is `mix_weight`, λ_2 is `1-mix_weight`, μ_1 and μ_2 are `predicted_goal[1]` and `predicted_goal[2]` respectively, and σ_1 and σ_2 are both equal to `sigma`. Lines 37-39 implement this likelihood function automatically using Stan's built-in `log_mix()` In this model, the likelihood cannot be evaluated using a statement of the form `y ~ normal(mean,sd)`, because the existence of the separate mixtures means that `observed_goal` will not be normally distributed. Instead, the likelihood must be calculated directly and the posterior must be updated using the `target+=` statement. See Stan user's guide for further information (http://mc-stan.org/users/documentation/).

of the data, then information contained in the parameter estimates will not be representative of the process the researcher is trying to investigate. In such cases, the researcher may need to respecify the model in order to improve its ability to account for the empirical observations.

In this section, we discuss tools that researchers can use to help determine whether a model provides a satisfactory description of the data. We must emphasize that there is no one-size-fits-all approach to model evaluation, and there are often strong theoretical reasons to prefer one model over another that need to be considered. Here, we simply address a few tools that researchers have at their disposal to facilitate the evaluation process.

Visual Inspection of Model Fit

One of the simplest, yet most powerful methods of evaluating the extent to which a model adequately describes the empirical trends is by visualizing the predictions of the model in the context of the data (Heathcote, Brown, & Wagenmakers, 2015). This approach is particularly useful for ruling out poorly performing models, because mismatches between the model and the data are often very obvious.

In a Bayesian model, the model predictions are delivered in the form of a distribution, which is commonly referred to as the *posterior predictive* distribution. The posterior predictive distribution represents the predicted distribution of the outcome variable(s) that is implied by the posterior distribution on the model parameters. The posterior predictive distribution can be generated by repeatedly sampling from the posterior distribution on the model parameters and simulating the process assumed by the model under each sampled parameter set.

Figure 4 shows summary data from the example dataset superimposed over the posterior predictive distributions from the two-mixture model described above, as well as a one-mixture model that estimates a single set of parameters for all participants (the one-mixture model is identical to the sample-level model described earlier in the paper). The columns in the figure are divided according to participants' mixture weights in the two-mixture model. The left column displays the data from participants whom the two-mixture model suggested were most strongly influenced by Mixture 1 (i.e., had a mean mixture weight greater than 0.5). The right column displays the data from participants who the two-mixture model suggested were most

strongly influenced by Mixture 2 (i.e., had a mean mixture weight less than 0.5). This visualization makes it easy to compare the fit of each models' predictions to the empirical trends. As can be seen in the left panel, both models reproduce the positive trend in goal level over time. However, the two-mixture model is a better match to the data than the one-mixture model, particularly for participants whom the two-mixture model suggested were more strongly influenced by Mixture 2 (displayed in the left column). The 95% credible interval associated with the two-mixture model almost always contains the observed mean, suggesting that this model does a good job of describing the data. On the other hand, there are many cases where the observed mean is outside the 95% credible interval associated with the one-mixture model, suggesting that this model provides a poorer description of the data.

It is important to note that the standards for model-data correspondence are likely to be highly context-dependent. In basic laboratory research, where there is a high degree of experimental control, it is often expected that model predictions provide an extremely close fit to the data. In such contexts, even minor discrepancies between the posterior predictive distributions and the experimental data may be a sign that the model is not an adequate explanation of the phenomenon being investigated. By contrast, in observational or field studies, where there is more noise, discrepancies between the model and the data may be more tolerable. In general however, researchers should be particularly wary of cases where the model produces a qualitatively different trend than the one observed in the data. For example, if a model predicted that people should *decrease* their goals over time, this would be a very strong indication that the model does not provide a good explanation, and should therefore be rejected.

The final point we wish to make regarding model visualization is that this tool can be used for evaluating any model that makes predictions, not just Bayesian models. In fact, we argue that visually inspecting the fit of a model to the data should be an essential practice when modeling any longitudinal data. Without knowing whether a model adequately characterizes the process that is being investigated, it is impossible to know whether parameters from that model can be interpreted meaningfully. Failure to conduct a visual inspection may therefore lead to inappropriate conclusions.

Quantitative Model Comparison

Although visual inspection of model-data fit can be helpful for comparing models, it is usually not sufficient for discriminating between them. If more than one model produces a close fit to the data, the differences in model-data fit may be too subtle to be picked up visually. Moreover, evaluation based on visual fit alone ignores the other key question that needs to be considered: parsimony. If two models fit the data to a similar extent, but differ in terms of complexity, the simpler model should be preferred on the grounds of parsimony (Myung & Pitt, 1997; Vandekerckhove et al., 2015).

Several approaches to Bayesian model comparison have been proposed that quantify the tradeoff between fit and parsimony. The standard solution for Bayesian model comparison is the Bayes factor (Jeffreys, 1935; Kass & Raftery, 1995). The Bayes factor refers to the ratio of the probabilities of the data under each model, and provides an index of the relative evidence delivered by the data for one model against an alternative that takes into account both fit and model complexity (Kruschke & Liddell, 2018).

There are some challenges associated with the use of the Bayes factor however (Vandekerckhove et al., 2015). First, the Bayes factor can be difficult to obtain when comparing complex models. Calculating the Bayes factor requires the likelihood of the data under the model to be integrated across the entire parameter space, which can be computationally cumbersome for models with many parameters. This is especially true for models that need to be estimated using MCMC methods. In recent years, methods have been introduced for approximating Bayes factors for models estimated via MCMC methods (e.g., Evans & Brown, 2018; Gronau et al., 2017; L. Wang & Meng, 2016). However, these methods are very computationally demanding.

Another issue associated with use of the Bayes factor is prior sensitivity. The Bayes factor is often influenced by the researcher's choice of priors (Rouder, Speckman, Sun, Morey, & Iverson, 2009). This is less problematic for standard, off-the-shelf, statistical tests, where considerable thought has gone into the recommended "default" priors (e.g., see JASP Team, 2018). However, when working with models that are novel and/or more complex, the researcher may be less confident in their choice of priors because a default specification will

likely not exist. In such cases, one can assess the robustness of the results by systematically examining the impact of different priors on the Bayes factor obtained. This process is referred to as prior sensitivity analysis (e.g., Wagenmakers, Wetzels, Borsboom, & van der Maas, 2011). It is worth noting that prior sensitivity is not necessarily a bad thing. It has been argued that priors are often an important aspect of the underlying theory being tested and are therefore an important consideration when evaluating a model (Vanpaemel, 2010).

Several other approaches to quantitative model comparison have been proposed (Gelman, Hwang, & Vehtari, 2014; Piironen, Vehtari, Piironen, & Fi, 2017). Two examples are leave one out cross validation (LOO-VC; Geisser & Eddy, 1979) and the Watanabe-Akaike Information Criterion (also known as the 'Widely applicable information criterion'; Watanabe, 2010). These indices quantify the ability of a model to predict new data, whereas the Bayes factor addresses the evidence for each model given by the obtained data (and the priors). The LOO-CV index and WAIC are interpreted in a similar manner to AIC and BIC, where a lower value indicates a better trade off between fit and parsimony. It should be noted however that these methods can be suboptimal for the purposes of inference (Etz & Vandekerckhove, 2018).

Discussion

Dynamic theories have become increasingly influential in organizational psychology and organizational behavior, but this type of theory can be difficult to test. Testing a dynamic theory requires a statistical model that accurately reflects the processes described by the theory (Collins, 2006). Although a wide range of statistical approaches have been used to examine dynamic phenomena, most do not have the flexibility to enable the direct implementation of a specific theory. As a result, researchers are often forced to distill rich, dynamic theories into simple predictions regarding the direction of relationships between variables, which creates misalignment between the theory and the statistical test. Ultimately, this hinders theoretical progress because it makes theories difficult to falsify.

In this paper, we have demonstrated a Bayesian approach to modeling dynamic processes that has the flexibility required to directly test dynamic theory. To illustrate the flexibility of this approach, we considered a model of goal revision recently published by Gee

et al. (2018). An important feature of this model that is typical of many dynamic theories is the assumption that certain variables have "memory". The ability to represent dynamic variables such as these is critical for testing theories that assume feedback processes, which are a signature of many influential theories. However, dynamic variables are difficult to implement using standard statistical methods.

The Bayesian approach that we have demonstrated addresses this issue, and provides an intuitive way of implementing sophisticated models of dynamic processes. To illustrate, we began by demonstrating a sample-level model that quantifies uncertainty in the average parameter values of the goal revision model across participants in the sample. We then showed how this framework could be extended to form more sophisticated models that captured variability between individuals and groups. The person-level model quantifies the components of the goal revision process separately for each individual. The hierarchical model combines the sample- and person-level models into a single framework, enabling components to be quantified separately for each individual, but also allowing inferences to be made at the population level. The multiple-group model quantifies differences in the components of the process between known groups (e.g., levels of an experimental manipulation). Finally, the mixture model allows the researcher to identify distinct, latent subgroups of participants for whom the dynamic process plays out in a similar way.

A More Flexible Approach

Importantly, the models that we have demonstrated in this paper represent just a small slice of the large space of possible models that can be implemented using this framework. The flexibility of this approach makes it possible to implement models that have virtually any functional form. This opens the door for researchers to develop customized models that provide a more accurate representation of the theory being tested than would be provided by generic, off-the-shelf models. For example, consider the learning literature, where decades of work has focused on understanding the precise nature of the relationship between practice and skill acquisition (e.g., Estes, 1994; Thurston, 1919). The dominant models in this literature describe the relationship as obeying either exponential or power laws, which are difficult to

implement using standard statistical models. Yet these models can easily be implemented within the framework described above. For example, an exponential learning model of performance in the above experiment might take the form `predicted_performance[i] = perf_max - (perf_max - perf_init)*exp(-learning_rate * trial[i])`, where `perf_max`, `perf_init`, and `learning_rate` are parameters representing the performance assymptote, the initial level of performance, and the rate of improvement in performance respectively.

Another example is the intertemporal choice and motivation literatures where a long history of research has demonstrated that people temporally discount future rewards or deadlines, treating them as less valuable or motivating as they would be if they were more immediate (Ainslie, 1975; Steel & König, 2006). One consequence of temporal discounting might be that people apply more effort as a deadline approaches. Theories of temporal discounting often describe this relationship as hyperbolic or exponential, which can both be easily implemented using the above framework. For example, a hyperbolic model could be implemented as `predicted_effort[i] = value / (1 + discount_rate * time_available[i])`, where `value` represents the value of completing the task, `discount_rate` represents the degree of temporal discounting, and `time_available` represents the amount of time before the task must be completed. In this model, `value` could be a free parameter or a variable the depends on other factors such as the discrepancy between the goal and the current level of performance.

Another class of phenomena that is notoriously difficult to examine using conventional methods is the bottom-up process (Kozlowski et al., 2013; Kozlowski & Klein, 2000). In a bottom-up process, a higher level phenomenon results from the dynamic behavior of a lower level process as it play out over time. Bottom-up processes can operate between the person and group levels (e.g., team knowledge emerging as the result of individual members learning and sharing; Grand et al., 2016). They can also operate within individuals over time (e.g., a behavioral set point emerging from a recursive goal regulation process; Ballard et al., 2017).

Although bottom-up processes are often assumed by theory, they are difficult to examine quantitatively. Historically, most of the work on bottom-up processes has been qualitative.

Recently, some have begun to use computational modeling to generate predictions regarding bottom-up processes (e.g., Grand et al., 2016). However, in both of these cases, it is difficult to test the predictions that a theory makes. The Bayesian approach that we have demonstrated offers a way to do this. For example, one might extend the goal revision model by combining it with the model of temporal discounting and effort described above. This combined model would describe a set of two feedback loops operating at different time scales. The lower level feedback loop governs the goal striving process, in which effort is adjusted rapidly in response to the looming deadline and the changing goal-performance discrepancy. Ultimately, the effort exerted during goal striving feeds into the higher level goal revision process, because it influences the overall performance on a trial, which informs the goal that is set on the next trial. Such a model would make it possible to capture the bottom-up process by which effort regulation during goal striving influences the higher level goal-setting process.

Challenges

Despite the many opportunities we believe that this approach offers, there are some challenges that should be considered. First, readers more accustomed to programs with graphical user interfaces such as SPSS, AMOS, or JASP may find that it takes time to become familiar with the Stan syntax. On the continuum from high level platform in which a handful of built-in analyses can be executed by button press (e.g., SPSS) to lower level programming languages that require the user to code up the entire analysis from scratch, Stan lies somewhere in the middle. Stan has all the functionality required to implement the MCMC algorithm built in to the program, but it requires the user to build the statistical model from the ground up. As the reader will have seen, this is much more involved than simply specifying a set of variables that are assumed to be related. However, the fact that the user has such control over the statistical model is what makes this approach so flexible. This control gives the researcher the ability to translate a set of theoretical assumptions directly into a statistical model, which enables a stronger test of the theory to be conducted.

A second challenge is the computational demands associated with this approach. Although the Bayesian approach has the benefit of being able to quantify uncertainty in model

parameters, as opposed to simply delivering a point estimate, this advantage comes at a cost. The MCMC algorithm that underlies this approach can require a long time to run. Although none of the models presented in this paper required more than about 60 seconds, our example dataset was fairly small (60 participants $\times$ 10 observations per participant). For models that are more complex, have more parameters, or that are being run on larger datasets, it is not uncommon for the analysis to take several hours to run. This can be challenging for analyses that require comparing a large number of alternative models. Fortunately, Stan has several features that help speed up the process, such as built-in parallelization. Additionally, much of this process can be automated.

Conclusion

For cumulative theoretical progress to be made, theories must be able to be subjected to rigorous tests using models that provide an accurate representation of the process described by the theory. This makes testing dynamic theories difficult, because the processes that these theories typically describe are far too rich and complex to be adequately characterized by standard statistical models. We have presented a Bayesian approach to modeling dynamic processes that has the flexibility required to capture the dynamism inherent in many of these theories. This approach enables a shift from thinking at the level of the variables to thinking at the level of the model as a whole, which will facilitate a more seamless integration between theory and statistical model. We believe this will ultimately accelerate theoretical progress by enhancing our develop, test, reject, and refine dynamic theories.

References

Ainslie, G. (1975). Specious reward: A behavioral theory of impulsiveness and impulse control. *Psychological Bulletin*, *82*, 463–496. doi: 10.1037/h0076860

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Caski (Eds.), *Proceedings of the second international symposium on information theory* (pp. 267–281). Budapest: Akademiai Kiado. doi: 10.1007/978-1-4612-1694-0_15

Ballard, T., Yeo, G., B. Vancouver, J., & Neal, A. (2017). The dynamics of avoidance goal regulation. *Motivation and Emotion*, *41*, 1–10. doi: 10.1007/s11031-017-9640-8

Bartlema, A., Lee, M., Wetzels, R., & Vanpaemel, W. (2014). A Bayesian hierarchical mixture approach to individual differences: Case studies in selective attention and representation in category learning. *Journal of Mathematical Psychology*, *59*, 132–150. doi: 10.1016/j.jmp.2013.12.002

Boehm, U., Marsam, M., Matzke, D., & Wagenmakers, E.-J. (in press). On the importance of avoiding shortcuts in applying cognitive models to hierarchical data. *Behavior Research Methods*.

Brooks, S. P., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 435–455.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal Of Statistical Software*, *76*, 1–32. Retrieved from `http://mc-stan.org/users/documentation/`

Carver, C. S., & Scheier, M. F. (1998). *On the self-regulation of behavior*. New York, NY: Cambridge University Press.

Collins, L. M. (2006). Analysis of longitudinal data: The integration of theoretical model, temporal design, and statistical model. *Annual Review of Psychology*, *57*, 505–528. doi: 10.1146/annurev.psych.57.102904.190146

Deshon, R. P. (2012). Multivariate dynamics in organizational science. In S. W. J. Kozlowski (Ed.), *The oxford handbook of organizational psychology* (pp. 117–142). New York,

NY: Oxford University Press.

Dienes, Z. (2008). Bayesian versus orthodox statistics: Which side are you on? *Perspectives on Psychological Science*, *6*, 274–290. doi: 10.1177/1745691611406920

Edwards, W., Lindman, H., & Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychological Review*, *70*, 193–242.

Estes, W. K. (1994). Toward a statistical theory of learning. *Psychological Review*, *101*, 282–289.

Etz, A., Gronau, Q. F., Dablander, F., Edelsbrunner, P. A., & Baribault, B. (2018). How to become a Bayesian in eight easy steps: An annotated reading list. *Psychonomic Bulletin and Review*, *25*, 219–234. doi: 10.3758/s13423-017-1317-5

Etz, A., & Vandekerckhove, J. (2018). Introduction to Bayesian inference for psychology. *Psychonomic Bulletin & Review*, *25*, 5–34. doi: 10.3758/s13423-017-1262-3

Evans, N. J., & Brown, S. D. (2018). Bayes factors for the linear ballistic accumulator model of decision-making. *Behavior Research Methods*, *50*, 589–603. doi: 10.3758/s13428-017-0887-5

Gee, P., Neal, A., & Vancouver, B. (2018). A formal model of goal revision in approach and avoidance contexts. *Organizational Behavior and Human Decision Processes*, *146*, 51–61. doi: 10.1016/j.obhdp.2018.03.002

Geisser, S., & Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, *74*, 153–160. doi: 10.1080/01621459.1979.10481632

Gelman, A., Hwang, J., & Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, *24*, 997–1016. doi: 10.1007/s11222-013-9416-2

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–511. doi: 10.1214/ss/1177013604

Grand, J. A., Braun, M. T., Kuljanin, G., Kozlowski, S. W. J., & Chao, G. T. (2016). The dynamics of team cognition: A process-oriented theory of knowledge emergence in teams. *Journal of Applied Psychology*, *101*, 1353–1385.

Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., … Steingroever, H. (2017). A tutorial on bridge sampling. *Journal of Mathematical Psychology*, *81*, 80–97. doi: 10.1016/j.jmp.2017.09.005

Hatch, M. J. (1993). The dynamics of organizational culture. *Academy of Management Review*, *18*, 657–693. doi: 10.5465/AMR.1993.9402210154

Heathcote, A., Brown, S. D., & Wagenmakers, E.-J. (2015). An Introduction to good practices in cognitive modeling. In B. U. Forstmann & E. J. Wagenmakers (Eds.), *An introduction to model-based cognitive neuroscience* (pp. 25–48). New York, US: Springer.

Hobfoll, S. E. (1989). Conservation of resources: A new attempt at conceptualizing stress. *American Psychologist*, *44*, 513–524.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623.

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge, UK: Cambridge University Press.

Jeffreys, H. (1935). Some tests of significance, treated by the theory of probability. *Mathematical Proceedings of the Cambridge Philosophy Society*, *31*, 203–222.

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.

Joyce, J. M. (1998). A nonpragmatic vindication of probabililsm. *Philosophy of Science*, *65*, 575–603.

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, *90*, 773–795.

Kozlowski, S. W. J., Chao, G. T., Grand, J. A., Braun, M. T., & Kuljanin, G. (2013). Advancing multilevel research design: Capturing the dynamics of emergence. *Organizational Research Methods*, *16*, 581–615. doi: 10.1177/1094428113493119

Kozlowski, S. W. J., & Klein, K. J. (2000). A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes. In K. J. Klein & S. W. J. Kozlowski (Eds.), *Multilevel theory, research and methods in organizations: Foundations, extensions, and new directions* (pp. 3–90). San Fransisco, CA:

Jossey-Bass.

Kruschke, J. K. (2010). *Doing bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press/Elsevier Science.

Kruschke, J. K., Aguinis, H., & Joo, H. (2012). The time has come: Bayesian methods for data analysis in the organizational sciences. *Organizational Research Methods*, *15*, 722–752. doi: 10.1177/1094428112457829

Kruschke, J. K., & Liddell, T. M. (2018). Bayesian data analysis for newcomers. *Psychonomic Bulletin & Review*, *25*, 155–177. doi: 10.3758/s13423-017-1272-1

Kruschke, J. K., & Vanpaemel, W. (2015). Bayesian estimation in hierarchical models. In J. R. Busemeyer, Z. Wang, J. T. Townsend, & A. Eidels (Eds.), *The oxford handbook of computational and mathematical psychology* (pp. 279–299). Oxford, UK: Oxford University Press.

Lee, M. D., & Wagenmakers, E.-J. (2013). *Bayesian cognitive modeling: A practical course*. New York, NY: Cambridge University Press.

Lewandowsky, S., & Farrell, S. (2011). *Computational modeling in cognition: Principles and practice*. Thousand Oaks, CA: Sage.

Lindley, D. V. (1993). The analysis of experimental data: The appreciation of tea and wine. *Teaching Statistics*, *15*, 22–25.

Lord, R. G., Diefendorff, J. M., Schmidt, A. M., & Hall, R. J. (2010). Self-regulation at work. *Annual Review of Psychology*, *61*, 543–568. doi: 10.1146/annurev.psych.093008.100314

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS – A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325–337.

Marks, M. A., & Mathieu, J. E. (2001). A temporally based framework and taxonomy of team processes. *Academy of Management Review*, *26*, 356–376.

Myung, I. J., & Pitt, M. A. (1997). Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin & Review*, *4*, 79–95.

Neal, A., Ballard, T., & Vancouver, J. B. (2017). Dynamic self-regulation and multiple-goal

pursuit. *Annual Review of Organizational Psychology and Organizational Behavior*, *4*, 410–423. doi: https://doi.org/10.1146/annurev-orgpsych-032516-113156

Piironen, J., Vehtari, A., Piironen, B. J., & Fi, A. V. (2017). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, *27*, 711–735. doi: 10.1007/s11222-016-9649-y

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In K. Hornik, F. Leische, & A. Zeileis (Eds.), *Proceedings of the 3rd international workshop on distributed statistical computing (dsc 2003).* Technische Universität Wien, Vienna, Austria. doi: 10.1.1.13.3406

Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, *12*, 573–604.

Rouder, J. N., Morey, R. D., Verhagen, J., Province, J. M., & Wagenmakers, E. J. (2016). Is there a free lunch in inference? *Topics in Cognitive Science*, *8*, 520–547. doi: 10.1111/tops.12214

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, *16*, 225–237. doi: 10.3758/PBR.16.2.225

Schönbrodt, F. D., Wagenmakers, E.-J., Zehetleitner, M., & Perugini, M. (2017). Sequential hypothesis testing with Bayes Factors: Efficiently testing mean differences. *Psychological Methods*, *22*, 322–339.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461–464.

Stan Development Team. (2016). *RStan: the R interface to Stan, Version 2.10.1.* Retrieved from `http://mc-stan.org/`

Stan Development Team. (2017). *Stan Modeling Language Users Guide and Reference Manual, Version 2.17.0.* Retrieved from `http://mc-stan.org/users/documentation`

Steel, P., & König, C. J. (2006). Integrating theories of motivation. *Academy of Management*

Review, *31*, 889–913.

Steiger, J. H. (1990). Structural model evaluation and modification: An interval estimation approach. *Multivariate Behavioral Research*, *25*, 173–180. doi: 10.1207/s15327906mbr2502_4

Team, J. (2018). *JASP (Version 0.8.6).* Retrieved from `https://jasp-stats.org/`

Thomas, A., O'Hara, B., Ligges, U., & Sturtz, S. (2006). Making BUGS Open. *R News*, *6*, 12–17.

Thurston, L. L. (1919). The learning curve equation. *Psychological Monographs*, *26*, 1–51.

Turner, B. M., Forstmann, B. U., Wagenmakers, E.-J., Brown, S. D., Sederberg, P. B., & Steyvers, M. (2013). A Bayesian framework for simultaneously modeling neural and behavioral data. *NeuroImage*, *72*, 193–206. doi: 10.1016/j.neuroimage.2013.01.048

Van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to Markov Chain Monte–Carlo sampling. *Psychonomic Bulletin & Review*, *25*, 143–154. doi: 10.3758/s13423-016-1015-8

Vancouver, J. B., Tamanini, K. B., & Yoder, R. J. (2008). Using dynamic computational models to reconnect theory and research: Socialization by the proactive newcomer as example. *Journal of Management*, *36*, 764–793. doi: 0.1177/0149206308321550

Vancouver, J. B., Weinhardt, J. M., & Schmidt, A. M. (2010). A formal, computational theory of multiple-goal pursuit: Integrating goal-choice and goal-striving processes. *Journal of Applied Psychology*, *95*, 985–1008. doi: 10.1037/a0020628

Vandekerckhove, J., Matzke, D., & Wagenmakers, E.-J. (2015). Model comparison and the principle of parsimony. In J. Busemeyer et al. (Eds.), *Oxford handbook of computational and mathematical psychology* (pp. 300–317). Oxford, UK: Oxford University Press.

Vanpaemel, W. (2010). Prior sensitivity in theory testing: An apologia for the Bayes factor. *Journal of Mathematical Psychology*, *54*, 491–498. doi: 10.1016/j.jmp.2010.07.003

Vincent, B. T. (2016). Hierarchical Bayesian estimation and hypothesis testing for delay discounting tasks. *Behavior Research Methods*, *48*, 1608–1620. doi: 10.3758/s13428-015-0672-2

Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems of p values.

Psychonomic Bulletin & Review, *14*, 779–804.

Wagenmakers, E.-J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Love, J., … Morey, R. D. (2018). Bayesian inference for psychology. Part I: Theoretical advantages and practical ramifications. *Psychonomic Bulletin & Review*, *25*, 35–57. doi: 10.3758/s13423-017-1343-3

Wagenmakers, E.-J., Morey, R. D., & Lee, M. D. (2016). Bayesian benefits for the pragmatic researcher. *Current Directions in Psychological Science*, *25*, 169–176. doi: 10.1177/0963721416643289

Wagenmakers, E.-J., Wetzels, R., Borsboom, D., & van der Maas, H. L. J. (2011). Why psychologists must change the way they analyze their data: The case of psi: comment on Bem (2011). *Journal of Personality and Social Psychology*, *100*, 426–432. doi: 10.1037/a0022790

Wang, L., & Meng, X.-L. (2016). Warp bridge sampling: The next generation. *arXiv preprint*. Retrieved from `https://arxiv.org/pdf/1609.07690.pdf`

Wang, M., Zhou, L., & Zhang, Z. (2016). Dynamic modeling. *Annual Review of Organizational Psychology and Organizational Behavior*, *3*, 241–266. doi: 10.1146/annurev-orgpsych-041015-062553

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, *11*, 3571–3594.

Weinhardt, J. M., & Vancouver, J. B. (2012). Computational models and organizational psychology: Opportunities abound. *Organizational Psychology Review*, *2*, 267–292. doi: 10.1177/2041386612450455

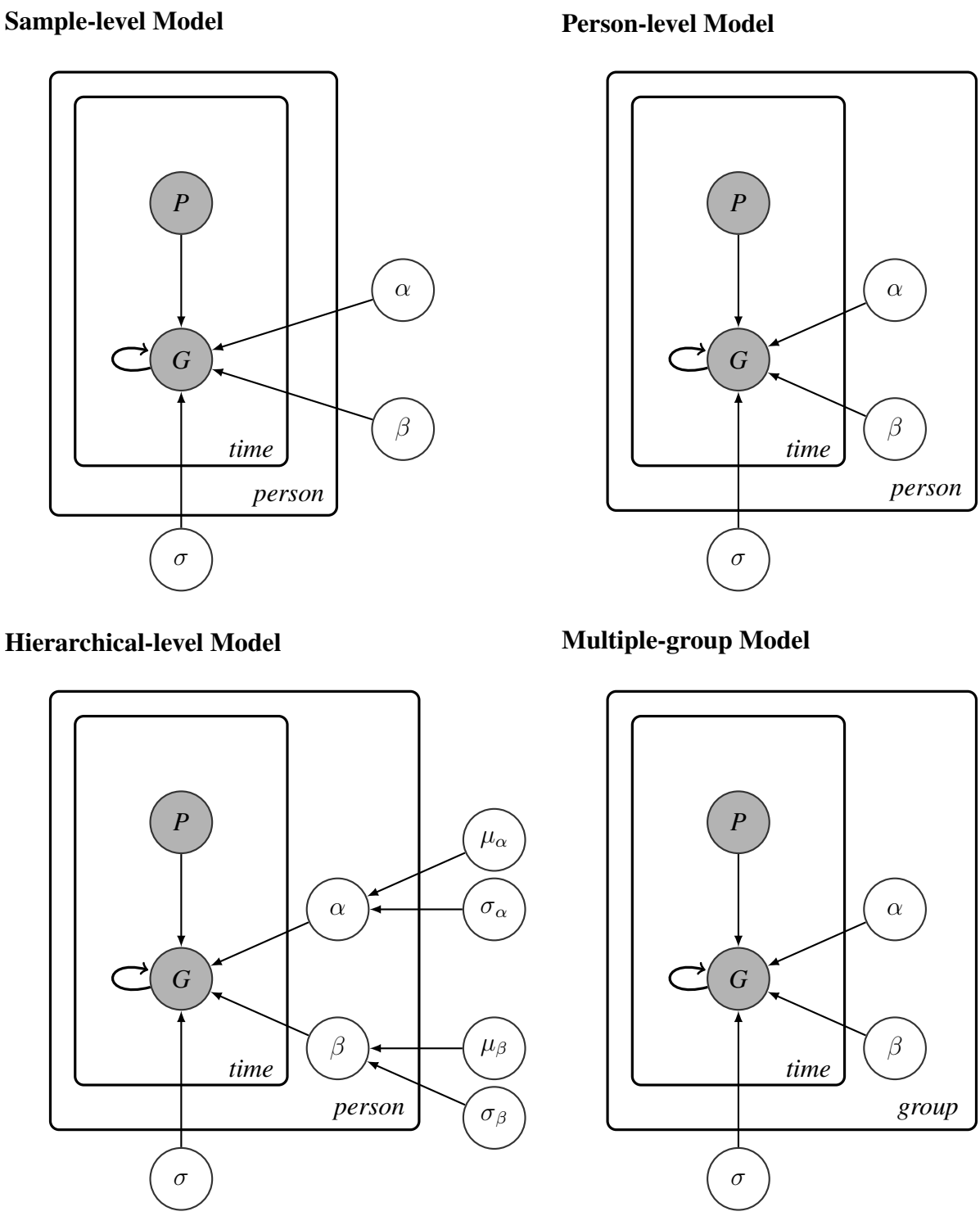

Figure 1. Graphical representation of the Bayesian models.

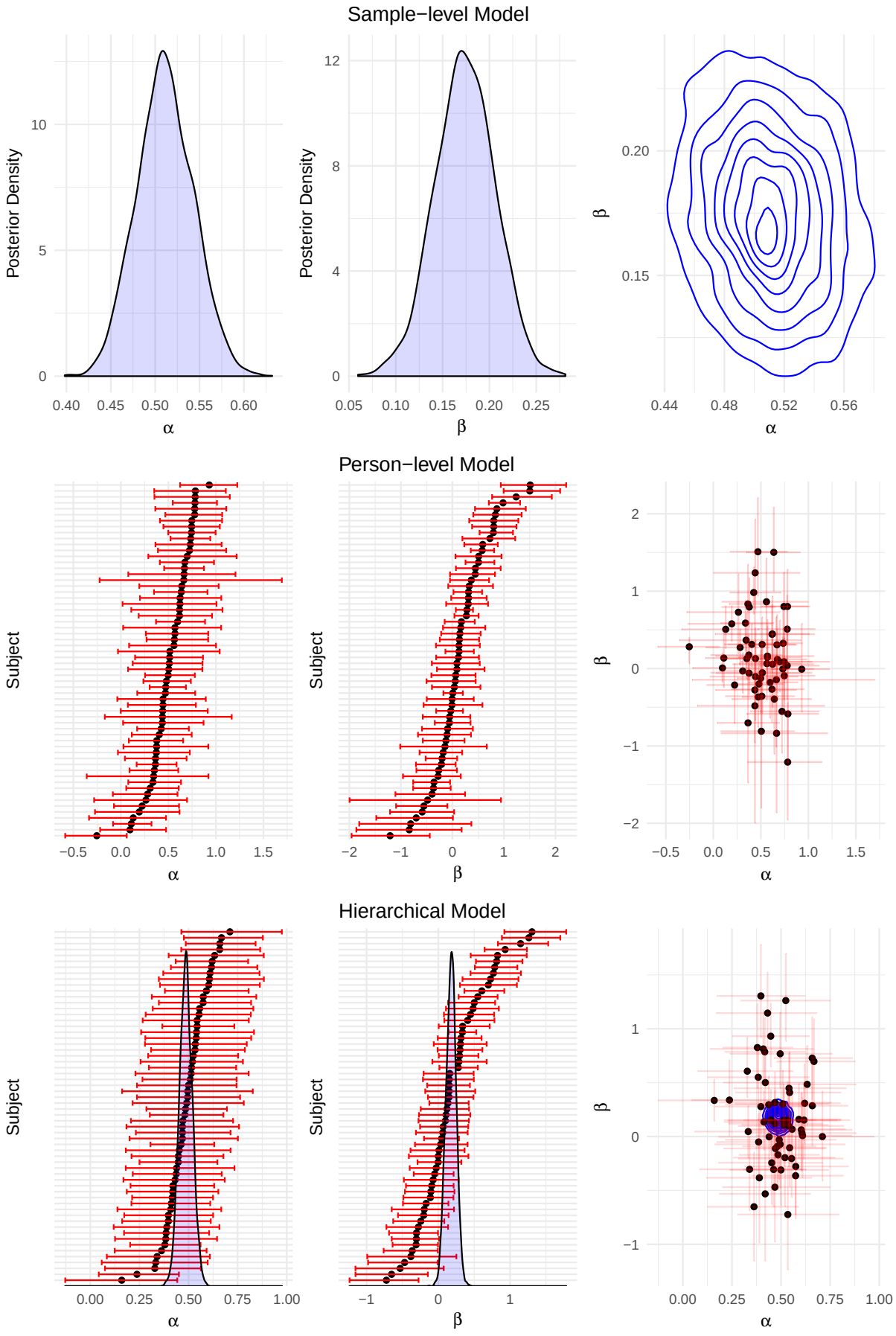

Figure 2. Posterior distributions on the α and β parameters for the sample-level, person-level, and hierarchical models.

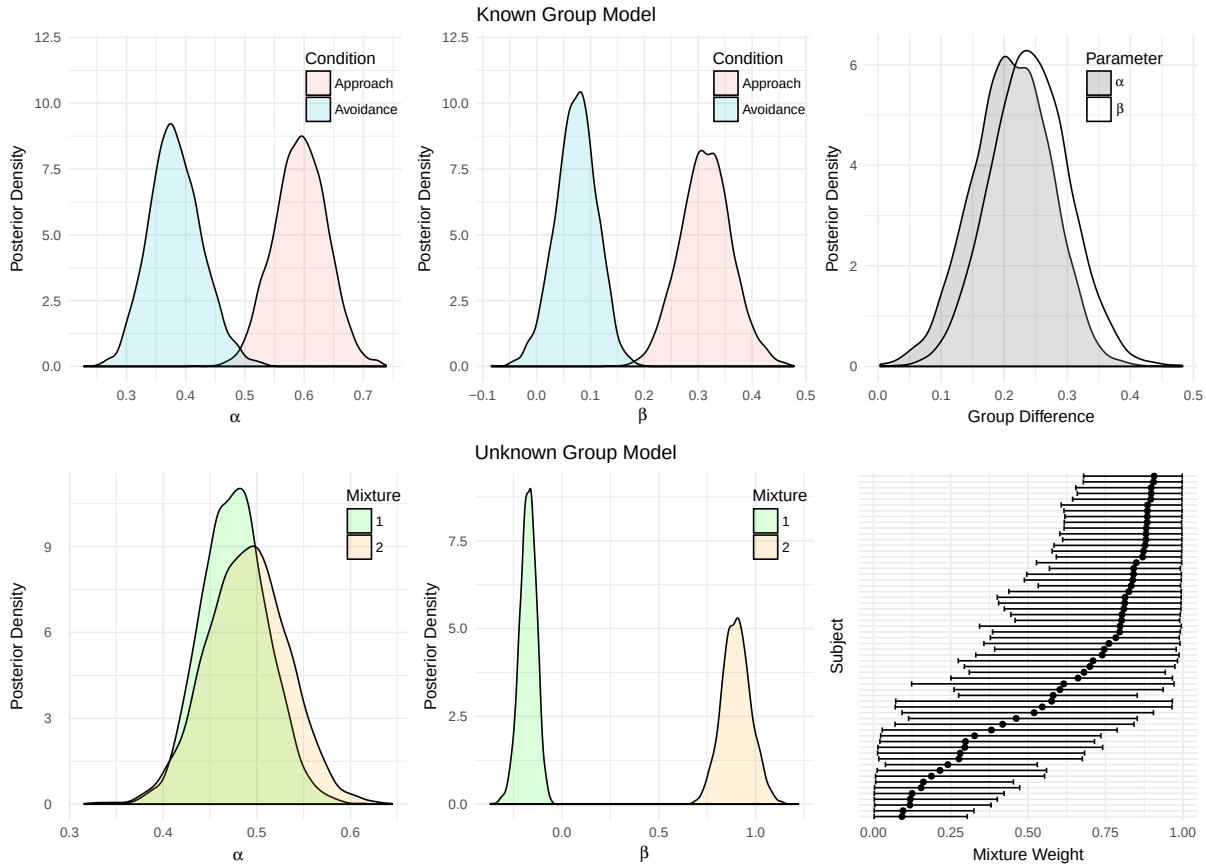

Figure 3. Posterior distributions on the α and β parameters for the known group model and the unknown group (i.e., mixture) model.

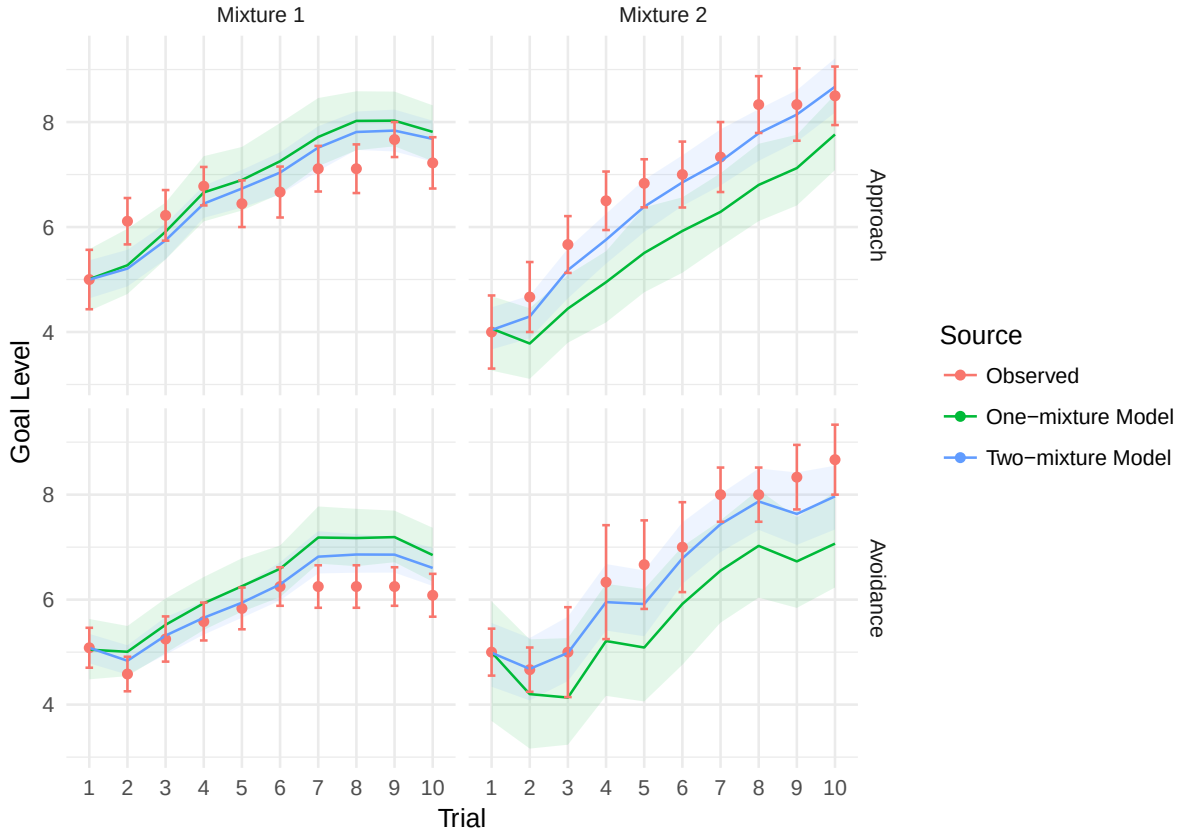

Figure 4. Summary data from the example dataset and posterior predictive distributions from one- and two-mixture models. The red dots and bars represent the observed means and standard errors respectively. The green line and ribbon represent the mean and 95% credible interval of the posterior predictive distributions from the one-mixture model. The blue line and ribbon represent the two-mixture model.