9

Bayesian parameter inference for cognitive simulators

Abstract

This chapter addresses the issue of parameter inference of computational cognitive models that simulate behaviour. Such cognitive simulators serve an important role in understanding and predicting human thought and behaviour by implementing hypotheses about the human cognitive processes and modelling these using stepwise simulations. In HCI, these models can be used for a range of applications, such as in UI evaluation and optimisation. However, their usefulness is limited without rigorous parameter fitting procedures, which permit efficient and informative parameter inference that can be used to assess confidence in parameter estimates, and easily replicated across experiments. In the usual case when the simulators are complex, common parameter fitting methods fail in this respect. In this chapter we discuss the feasibility of Bayesian parameter inference for cognitive simulators, presenting practical solutions to Bayesian parameter inference, and demonstrating its usefulness with two cognitive models simulating interactive tasks. Furthermore, we discuss the implications of efficient, informative, and robust parameter inference for the future of HCI.

9.1 Introduction

A long-standing objective of human-computer interaction (HCI) and artificial intelligence (AI) research is to facilitate the use of interactive and intelligent systems by developing a better understanding of the users of these systems. This involves taking into account the users' goals, beliefs, and abilities, in an attempt to align the information ecologies supported by interactive systems with the users' preferences. In this chapter, we consider the hard problem of

understanding the user by asking how it can be furthered by inferring parameters of computational cognitive models. These models act as *simulators* that map latent user characteristics to observed behaviour. We use the term 'simulation' to emphasise that these models make step-by-step predictions (in some units of time) of the progress of an interactive task. The validity of these simulators and thus usefulness depends on informative and robust parameter inference. The idea of these models, simply put, is that they implement a series of hypotheses about the human cognitive processes, attempting to predict behaviour based on individual traits and experiences, task description, and interactive environment characteristics, which are all represented as parameters in the model.

Given that the structure of a model is psychologically valid and its parameters are specified and interpreted correctly, it is possible to predict what the user would do in different circumstances. This is called *forward modelling*, wherein adjusting the parameters that shape the model's task environment can be used to ask "what if" questions, thus providing information to decision-making, such as adaptation of user interfaces. For instance, a cognitive model of layout learning can be used to predict visual search performance over different user interfaces (UIs) and for users with different expertise level, permitting an automated and individualised evaluation of various design choices [18, 19]. In this regard, such models can be used to better adapt technologies for the psychology of users, because they improve our understanding of the latent factors behind users' behaviours. Moreover, they are helpful in developing HCI as a science, permitting theory development and testing by forcing researchers to explicate their theoretical assumptions about users, and allowing simulating the hypothesised user behaviour.

A major challenge for the development, generalisation, and utilisation of cognitive models is that they often encompass a large number of parameters, upon which their psychological validity stands. The existence of these parameters *per se* is not the problem, but the predictions that the models make are useful only insofar it is possible to claim that the values of these parameters are set plausibly – which is a hard problem. *Parameter inference* refers to identifying values that are theoretically plausible and lead to realistic predictions. For many parameters of cognitive models used in HCI, parameter inference is fairly straightforward, as the values in question are identified on the basis of existing psychological research or derived from the specifics of the known task environment. For instance, in the aforementioned layout learning model, parameters governing the movement of eyes are psychologically established and not suspect to significant variation between tasks or individual users. Similarly, parameters dictating the task environment, such as the size and locations

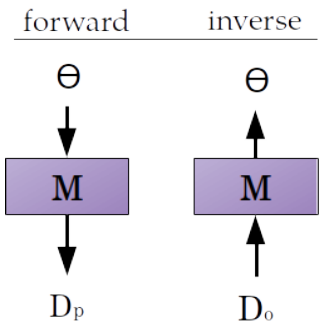

Figure 9.1 In forward modelling, the model M is run with fixed parameter values θ to produce predictions D_p. Conversely, in inverse modelling, observations D_o are used to infer plausible values for θ.

of visual layout elements, are not difficult to set as long as the specifics of the UI are fully known to the modeller. However, there are two cases where a parameter cannot be fixed a priori: (1) when the static parameter partially encloses or "hides" dynamic cognitive mechanisms, therefore requiring recalibration with new tasks; and (2) when variance in the parameter value can be connected to individual differences in behaviour. In both instances, parameter inference becomes an *inverse modelling* problem: given observed behaviour, what are the plausible parameter values? The difference between forward and inverse modelling is illustrated in Figure 9.1.

This chapter focuses on the use of Bayesian parameter estimation for inferring parameters of cognitive simulators based on observed behavioural data. The advantage of the Bayesian approach is in its ability to express prior information and uncertainty about parameter estimates using probability theory. The outcome of parameter estimation is a posterior distribution that expresses probabilities for different parameter values and combinations when the observed behaviour is taken into account. Usually, the reason these simulators are applied and their parameters inferred is to predict and understand user behaviour. This can be accomplished by generating a posterior distribution of predicted behaviours using the inferred parameter distributions. In practice, posterior estimation for parameters of cognitive simulator models is carried out with likelihood-free inference (LFI). LFI methods provide a way to respond to a critical problem with Bayesian inference of parameters of complex simulators: because the models are not closed-form equations, but stepwise simula-

tors involving stochasticity, fit to data under certain parameters can only be established by running the simulator with said parameter values. While Bayesian parameter inference and LFI methods have been received with some interest in cognitive science [41, 40, 21, 9, 42], the ultimate goal of this chapter is to facilitate a wider use of these techniques within the computational interaction modelling community.

In what follows, we first define and give examples of computational cognitive models, as used as simulators in HCI to generate predictions of user behaviour. We then discuss the problem of parameter inference for these simulators, and review the ways that it can be accomplished. Our focus is on Bayesian parameter inference. We discuss common LFI approaches and demonstrate the use of approximate Bayesian computation (ABC) as an instance of Bayesian parameter inference applicable to cognitive simulators. As we will argue, this approach is appealing for two reasons: (1) Because it permits formally incorporating prior knowledge to inform parameter estimation; and (2) Because it provides an informative posterior, which can be used to assess the amount of confidence in the parameter estimates. Both of these features increase our capacity to learn about parameters of cognitive models, especially when these parameters are individually determined and critical for predicting task behaviour, and implement interventions for better facilitation of user adaptation.

9.2 Bayesian parameter inference for cognitive simulator models

This section formalises Bayesian parameter inference and how it can be used to infer the parameters of cognitive simulator models. We first outline the process of parameter fitting for cognitive simulators (Section 9.2.1) and then discuss how Bayesian parameter inference can be used efficiently and informatively in this process (Section 9.2.2). This is achieved with the likelihood-free methods discussed in Section 9.2.3.

9.2.1 Parameter Inference for Cognitive Simulator Models

In our formal investigation of parameter inference for cognitive simulators, we define a model M with parameters θ. Executing the model maps the input parameters into data that consist of predictions made by the model. Because cognitive models generally involve stochastic elements, the same parameter values do not necessarily generate the same output upon subsequent executions of the model. As a result, it is often necessary to run the model multiple times

with the same parameter values, resulting in a more informative distribution of predictions. The prediction data D_p from the model are therefore sampled from the parametrised model $M(\theta)$, instead of representing a deterministic mapping from it.

Compared to testing with human participants, the major benefit of computational cognitive models for HCI is that they are simulators, which can be executed as many times as required. Varying the parameters of the models, especially those pertaining to the task environment, such as the UI, allows researchers and designers to investigate how various UI changes impact behaviour. We present some examples of how cognitive simulators have been used in HCI in Table 9.1. Because these simulators implement hypotheses about how the human cognition processes information and how this process results in behaviour, simulation of human-like behaviour is often possible even for scenarios or interventions for which no existing human data are available. Furthermore, given that a parameter can be posited a role in determining individual behaviour, varying its value permits the prediction of behaviour from various types of users.

However, these simulations are only good insofar the parameters that govern cognitive processes are set correctly. Where the value or distributions of values of a parameter cannot be deduced from the task or reused from existing research, they must be inferred from observed behaviour. Based on the short review of parameter fitting procedures of cognitive models used in HCI, there is currently no standard and easily replicated inference methodology in place. In case of cognitive models, when parameters have been inferred from observed data, this inference is often either not reported, or it is reported as having been done by adjusting the parameter values by hand until model fit to human data was deemed acceptable [21]. There are some exceptions to this in HCI, such as [33] using genetic optimisation, and [15] a simplex method, but generally the field lacks a standard for parameter inference. To our knowledge, there are only few publications in HCI that report using Bayesian parameter estimation [11, 20].

Observing human behaviour, either via psychological experimentation or from observations made during real-life interaction, provides us with data D_o, which can be then used to infer θ. This assumes that the data observed from users and generated by simulating the model are similar in that they describe the same behaviours. For instance, the data can be in terms of aggregate data, such as mean task times or error counts, or more detailed observations, such as logs of user inputs from completing a single task. The most common practice in HCI is to collect data from multiple users, and then create *summary statistics* by aggregating first within and then across individuals, thus abstract-

Table 9.1 *A list of example computational cognitive models in HCI and how their parameters have been set (F = Fit to observation data, L = based on literature, U = Undefined, D = Default values)*

Ref	**Target**	**Applications**	**Cognitive architecture / theory**	**Example parameter**	**Inf.**
[14]	Visual search in graphical user interfaces (GUI)	GUI design, predictive tools for testing designs	EPIC	Recoding time for text	L
[10]	Web navigation via hyperlinks	Web design	SNIF-ACT	Attentional weight	F L D
[26]	Information search via search engine	Search engines	ACT-R	Retrieval threshold	D,U
[35]	User multitasking	In-vehicle user interface design and testing	ACT-R	Steering style	F
[2]	Effects of interruptions on user cognition	Interface design for managing interruptions	ACT-R	Time to store a new representation into problem state	L
[33]	Memory-based interaction obstacles	Workload management	CMM	Degree of memory decay	F
[3]	Aviation surface operations and performance	Pilot performance and error prediction	ACT-R	Noise in production utility computation	U
[30]	Decision-making of air traffic controller (ATC)	ATC system design and training	ACT-R	Activation noise	D,U
[6]	Human error	Human reliability analysis	SHERPA	Scale and shape parameters of error probability density function	L
[39]	Human communication	Human-robot communication and interaction	ACT-R/E	Waiting time until switching attention	L
[29]	Impact and use of mobile health applications	Mobile health applications	ACT-R	ACT-R parameters	D
[15]	Safety incident reporting decisions	Mobile crowdsourcing applications	DDM	Response bias	F
[17]	Multitasking in driving	In-vehicle user interface design and testing	CR	Action noise	L

ing the relevant behavioural aspects of the interaction into few selected point estimates.

Parameter inference for cognitive simulators can be used to determine user or population-level characteristics based on the observed data. Traditionally,

the aim is to estimate parameters $\hat{\theta}$ that maximise a model fit measure. For example, maximum likelihood estimation is based on the idea that parameters θ are supported by the observed data D_o in proportion to the likelihood $P(D_o|\theta)$, which is the probability that the model M with parameters θ generated the observed data D_o, $P(D_p = D_o|\theta)$. Thus, even though the likelihood does not provide direct information about how much a parameter value is supported, this proportionality permits comparing different parameter values according to their likelihood. The maximum likelihood estimate $\hat{\theta}$ is calculated as the parameters θ that maximise $P(D_o|\theta)$. However, when a complex simulation model is used to describe the dependencies between the parameters and simulation outcomes, the observation probabilities $P(D_p = D|\theta)$ can be hard or impossible to determine. This is due to the models being simulators, which cannot be expressed analytically but must be executed in a stepwise fashion, with each step processing information according to complex rules, with potentially multiple sources of added noise. In this case, parameter estimates $\hat{\theta}$ must be determined based on other model fit measures, such as prediction error or discrepancy between summarised D_o and D_p.

While parameter estimation is possible based on model fit measures also when complex simulator models are studied, a problem with point estimates $\hat{\theta}$ is that since the set of observed data is limited in size, the parameters that best match the available observations may not be the ones that best describe the user or make the most accurate predictions about user behaviour in new situations. In addition, a point estimate does not express the amount of confidence that the estimate is correct, and therefore it does not let researchers and designers to consider how strongly to trust predictions made from a model parametrised in this way. These problems associated with point estimates can be avoided by applying Bayesian parameter inference.

9.2.2 Bayesian Parameter Inference

Bayesian parameter inference is grounded on the fact that we generally cannot know the exact parameter values that best describe the observed data and user(s) behind it. However, it is possible to obtain information about the parameter values, and this information can be represented as a probability distribution over the possible parameter values. Possible information about parameters θ includes both what can be learned based on observations D_o, as discussed in the previous section, and our expectations about plausible parameter values based on what we know about the simulator model. The idea is to express, prior to making observations, what we know about the parameters as a distribution $P(\theta)$, and then make observations to update this expectation. Given the prior

probabilities $P(\theta)$ and observation likelihood $P(D_o|\theta)$, posterior probabilities are defined as

$$P(\theta \mid D_o) = \frac{P(D_o \mid \theta)P(\theta)}{P(D_o)} \tag{9.1}$$

where $P(D_o)$ is the marginal likelihood $P(D_o) = \int P(D_o|\theta)P(\theta)d\theta$. The posterior $P(\theta|D_o)$ is a probability distribution over parameter values, and it represents what we know about the unknown parameters when we take into account that these parameters produced the observations D_o. Since observations are expected to reduce uncertainty about the parameter values, the posterior distribution is usually more concentrated than the prior. In case where the observations D_o are not informative about the unknown parameters, the posterior will remain close to the prior.

The prior distribution quantifies the modeller's assumptions about plausible parameter values before any comparison between model predictions and observed data are made. It can encode known psychological or physiological constraints, information about known dependencies between parameter values, such that one parameter value cannot exceed another, or information about the most probable values. For example, if previous experiments provide information about how a parameter value varies in a population, this can be used as prior information about what value the parameter takes in a new individual. That such prior information is taken into account in parameter inference is especially important when we wish to make predictions based on a small observation set, because parameter values chosen based on the observations alone could be nonsensical.

Posterior probabilities provide an intuitive way for the modeller to understand which parameter values could describe the observed user. This means that while the posterior can indicate the probable parameter values, it also provides information about uncertainties and allows us to answer questions like what is the probability that the unknown parameter value is below certain threshold. Moreover, when we want to predict how the observed user behaves in a new situation, the posterior can be used to calculate a *posterior predictive distribution*. In practice, when predictions are generated with a cognitive simulator model, the posterior predictive distribution is sampled by drawing parameter values from the posterior and running the simulator parametrised with these values. With enough samples drawn and simulations run, the predictions from the simulation runs form the posterior predictive distribution. Generating the predictive distribution based on a posterior sample rather than a point estimate allows us to take into account the uncertainties in parameter values and ensure that the predictive distribution variance is not underestimated. Figure

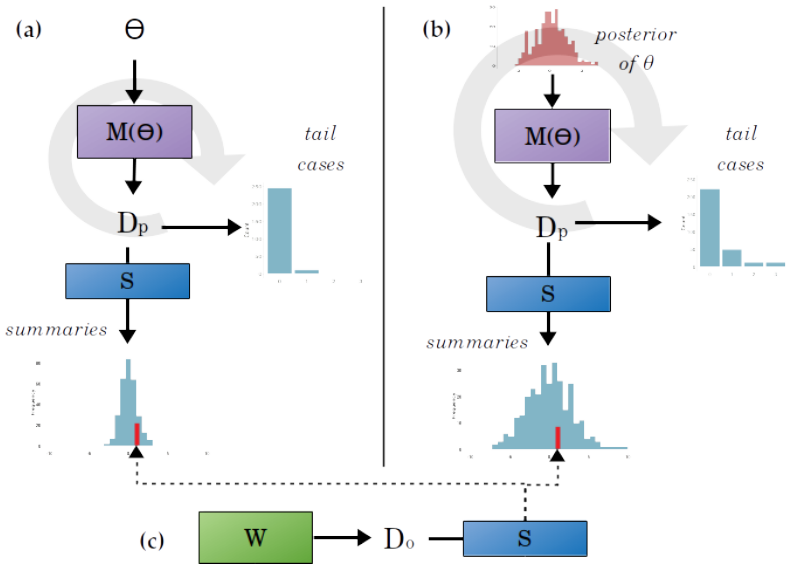

Figure 9.2 **(a)** A simulator M with fixed parameters θ can be run multiple times to produce a series of predictions D_p, which are each summarised, resulting in a distribution of summaries. **(b)** When a posterior of parameters θ has been created using Bayesian parameter inference, a posterior predictive distribution can be generated by repeatedly sampling values of θ from the posterior and summarising the resulting predictions D_p. Note that posterior predictive inference may be able to more plausibly estimate the occurrence of tail cases of some model features. Concentrations of observable model features in (b) illustrate that the simulator with fixed parameters can underestimate the amount of tail cases. **(c)** Observing a process in the world W produces data D_o that can then be summarised using summary statistics S.

9.2 summarises how distributions of summary statistics can be generated with either fixed parameters or by sampling the posterior.

To summarise, posterior probabilities combine prior information with observation likelihoods and provide more information for the modeller than a point estimate. A problem with the posterior is that it can be expensive or impossible to compute. Even when the likelihoods $P(D_o|\theta)$ and prior probabilities $P(\theta)$ can be evaluated, $P(D_o)$ may not be computable in practice. Hence it is common to work with unnormalised posterior values $P(D_o|\theta)P(\theta)$. The unnormalised posterior does not associate parameter values with actual probabilities, but can be used much like the likelihood function to compare posterior support to parameter values and to determine maximum a posteriori estimates. It can also be used to sample the posterior distribution, and a sample with N

parameter values can be used to calculate descriptors like posterior distribution mean. Also running the simulator with the sampled parameter values produces a sample from the posterior predictive distribution.

Finally, a problem with many cognitive simulator models is that the likelihood $P(D_o|\theta)$ cannot be determined based on the model. In this case posterior estimation has to be carried out with likelihood-free methods that determine approximate posterior probabilities based on prior information encoded in $P(\theta)$ and repeated simulation experiments.

9.2.3 Likelihood-Free Inference

Likelihood-free inference provides means to estimate posterior probabilities over parameter values when it is not possible to calculate the observation likelihood. This section introduces basic ideas in likelihood-free posterior estimation based on approximate Bayesian computation (ABC) and its recent alternatives. ABC methods substitute likelihood evaluations with direct comparisons between observed and simulated data. These classic methods have been reviewed in [41, 23, 37]. The other methods introduced in this section use simulations to learn a distribution model that can be used to compute approximate posterior probabilities based on the observed data. These have been reviewed and discussed in [5].

ABC can be carried out with a rejection sampler that constructs an approximate posterior sample as follows. First, candidate parameter values θ are sampled from the prior distribution $P(\theta)$, and the simulator that predicts user behaviour based on parameters θ is executed with θ to generate simulated user interactions D_p. The simulated data D_p are then compared to the observed data D_o, and the candidate parameter value is accepted in the approximate posterior sample if difference $\Delta(D_o, D_p)$ between the simulated and observed interactions is below certain threshold ϵ. The process can continue until N candidate parameters have been accepted or until the simulation count exceeds a predetermined maximum.

The above method substitutes likelihood evaluations with direct comparisons between observed and simulated data. In practice the likelihood $P(D_p = D_o|\theta)$ is approximated as $P(\Delta(D_o, D_p) < \epsilon|\theta)$. This would be exact with $\epsilon = 0$ and $\Delta(D, D')$ that is a distance metric such that $\Delta(D, D') = 0$ when $D = D'$ and $\Delta(D, D') > 0$ otherwise. However since we work with stochastic simulators, innumerable trials could be needed to produce even one simulation outcome D_p that matches the observed data D_o under these conditions. Hence we must allow some approximation error, and the difference measure

$\Delta(D, D')$ and tolerance threshold ϵ are needed to define the conditions under which simulation data are considered acceptable match with the observed data.

The difference measure $\Delta(D, D')$ formalises what makes observations similar. A common approach is to compress observations into informative features called summary statistics and define $\Delta(D, D')$ as distance between the summaries $S(D)$ and $S(D')$. Summarisation should reduce variation that is not informative about the unknown parameters θ so that the distance between $S(D)$ and $S(D')$ is close to zero when D and D' are data simulated with the same parameters. The summaries are often constructed based on domain information about what features in the observed and simulated data could be sensitive to the unknown parameter values, but it is also possible to derive and choose between candidate statistics based on simulation experiments. For more information on summary statistics selection, we recommend [31].

Problem with the rejection sampler is that the acceptance rate is expected to be low: since posterior distributions tend to be more concentrated than the priors, most candidate parameter values will not be accepted when these are chosen based on their prior probabilities. The tolerance threshold can be used to increase the acceptance rate, but this increases approximation error and makes the posterior distribution wider and closer to the prior distribution. A solution is found in iterative methods that take into account what has been learned about the previous trials when choosing the next candidate parameters. Iterative solutions include methods like ABC population Monte Carlo (ABC PMC), which is discussed and evaluated with cognitive simulator models in [41]. This method starts with the prior distribution and uses the samples that produced the lowest $\Delta(D_o, D_p)$ to determine the next proposal distribution. Since a proposal distribution constructed in this manner is expected to become more and more concentrated around the posterior as iterations proceed, fewer simulations are run with parameters that have low posterior probabilities. The same idea motivates advanced methods like BOLFI [13], which uses sequential model-based optimisation to locate parameter values that minimise expected $\Delta(D_o, D_p)$. BOLFI has been used to estimate cognitive simulator model parameters in, for instance, [20, 11].

Alternatives to the ABC methods that compare simulated and observed data include methods that use simulated data to learn about statistical dependencies between the parameters and simulation outcomes. In practice the dependencies can be encoded in a distribution model that is fitted to the simulated data and used to determine an approximate posterior model $\hat{P}(\theta|D)$ that can be evaluated at $D = D_o$. Methods based on this principle include the synthetic likelihood and its extensions [43, 32] and methods that utilise kernel density estimation or neural density estimation to model the simulated data distribu-

tion. Density estimation has been evaluated with cognitive simulator models in [40]. Related methods also include density-ratio estimation where the approximate posterior estimation problem is converted into a classification problem [8].

Density estimation and other approaches that use simulations to estimate an approximate distribution model operate on the simulated data or summaries, and do not require a difference measure or tolerance threshold to be determined. The methods require numerous simulations as training data, but sequential versions exist that utilise the observations D_o to avoid excess simulations with parameters that have low posterior probabilities. These include methods like sequential neural likelihood [28] and automatic posterior transformation [12]. Moreover there are applications in HCI wherein the total computation time or total simulation count matters less than the response time between when the user interaction occurs and when the posterior estimate is available for adaptation or other such purposes. These applications could find it valuable that, when observations are not used in the training process, the approximate posterior model can be learned based on simulations offline, and online computations reduce to evaluation at $D = D_o$.

Finally, a concern with the likelihood-free methods is whether posterior estimation can be carried out when the simulator model has many unknown parameters. The standard ABC methods discussed earlier in this section work best with low-dimensional summary statistic which cannot capture information about numerous parameters. This means that the summaries dimension increases hand in hand with the parameter dimension, and the standard methods are not applicable in problems with more than a few parameters. However the density and density-ratio estimation methods can be less sensitive to problem dimension, and the standard methods also have variants that decompose difference measure and posterior distribution in order to handle high-dimensional problems. These methods are discussed in [25].

9.3 Using Bayesian parameter inference with cognitive simulator models

In this section, we demonstrate the use of Bayesian likelihood-free inference for fitting parameters of computational cognitive models that simulate user behaviour in HCI tasks. Of the two examples provided, the first demonstrates fitting of the model to aggregate data. The second example is then presented to demonstrate how to infer parameter values for individual users. Posterior estimation is carried out with ELFI [24] tools in both examples.

9.3.1 Parameter estimation of a menu search model

We replicate an experiment presented in previous work [21] where a menu search model was fitted to behavioural data both with methods that optimise a traditional model fit measure and with the ABC method BOLFI [13]. The likelihood-free inference experiments with BOLFI were used to estimate posterior distributions over parameter values based on aggregate and individual observations collected in earlier work [1]. The present demonstration focusses on the population level posterior distribution that is estimated based on aggregate data. Our aim is to fit the model parameters governing eye movement time and memory recall.

The task model

The model studied in this experiment simulates the visual search of menus [4]. The model predicts eye movements in a task where users fixate on the elements of a drop-down menu to find a cued target. Importantly, it predicts how eye movement patterns are a result of adapting to the UI design and cognitive constraints. Therefore, as strategies emerge as adaptations to the UI design, the model permits investigating how different menu designs impact behaviour. The task and menu environment are described as a Markov decision process, and the simulated user is represented as a computational agent. The agent's actions include fixating on any of the menu items, which causes a saccadic eye movement towards that item, and its subsequent encoding. The task ends upon encoding of the cued target. Additionally, the agent can decide to quit, which is necessary for ending the task in cases where the target is absent. The agent receives negative rewards for time spent on the menu search and a positive reward for finding the item or correctly ending the task, and reinforcement learning is used to discover the optimal search policy.

The menu search model used in the present experiment is based on the version proposed in [20]. Here the task completion times depend on the search path that the model learns to optimise and two parameters that describe the user: duration associated with each fixation and selection delay that occurs at task completion. The model also takes into account that users sometimes remember the whole menu based on the first item, in which case the optimum behaviour is to directly fixate the target item. This is modelled as menu recall probability. Further model parameters control how shape and semantic relevance can be observed with peripheral vision, and encode properties of the searched menu. These were set to replicate the previous experiment [21].

Finally the parameters that control learning and data collection were set as follows. The behavioural pattern that is expected to minimise the menu search

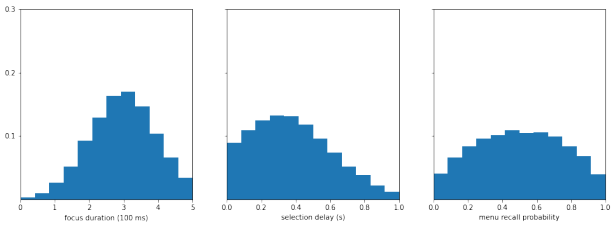

Figure 9.3 Prior distributions over menu search model parameters.

time under given user and menu parameters was learned based on 500000 training episodes. The limited training episodes mean that the simulator may not learn the exact same behaviour each time even when called with the same parameters. This is one reason we see variation in the simulated task completion times. Another reason is that the simulated data is collected over 100 menu search tasks with random menus and target location. This means that two datasets simulated with the same parameters do not describe task completion times in the exact same tasks. In this experiment, eight items were always present in the menu, with most of the time (90%) the cued target being present, and sometimes (10%) not.

Parameter estimation

The parameters inferred based on observed behavioural data include focus duration, selection delay, and menu recall probability. The focus duration and selection delay parameters are associated with the same prior distributions that were used in previous work [21]: focus duration has a normal distribution prior with mean 300 ms and standard deviation 100 ms truncated to interval [0, 500] ms and selection delay a normal distribution prior with mean 0.3 s and standard deviation 0.3 s truncated to interval [0, 1] s. Menu recall probability is associated with a beta distribution prior with parameters $\alpha = \beta = 1.5$. The prior distributions are visualised in Figure 9.3.

The observation data for this example case come from a study where human participants conducted the menu search task [1]. The observations used in posterior estimation include the task completion times and whether or not the cued target was present in the menu in each trial. The observed and simulated data are compared based on the summaries and distance proposed in [20]. Task completion times are compressed into mean and standard deviation calculated across trials where the target was present and across trials where the target was not present, and comparison between observed and simulated data

is based on squared distance between the means and absolute distance between the standard deviations.

Posterior estimation is carried out with the BOLFI method available in ELFI [24]. BOLFI [13] uses a surrogate model to describe dependencies between parameter values and $\Delta(D_o, D_p)$. Gaussian process regression with a squared exponential kernel is used in the present experiment, and the unknown simulator parameters are each associated with a separate surrogate model parameter to encode how sensitive $\Delta(D_o, D_p)$ is to variation in that parameter. These length scale parameters were also associated with Gamma prior distributions with shape $\alpha = 2$ and rate $\beta = 10/(b - a)$, where a and b denote the parameter minimum and maximum value considered in this experiment, to ensure reasonable predictions when the model is fitted based on limited data. The probabilities $P(\Delta(D_o, D_p) < \epsilon)$ that are used as approximate likelihoods can be calculated based on the surrogate model, which means that the surrogate model can substitute the simulator when we sample the approximate posterior. BOLFI initialises the surrogate model with simulations run with candidate parameters sampled from the prior distribution, but runs most simulations with parameter values chosen based on the current model estimate. 50 parameter combinations were sampled from the prior and 450 were selected based on the lower confidence bound acquisition rule in the current experiment. Alternative acquisition rules are discussed in [16].

Results and discussion

We sampled the approximate posterior determined based on the surrogate model learned in BOLFI with $\epsilon = 9$. The sample is used to estimate the approximate posterior mean, which as a parameter estimate minimises the expected squared error between the estimate and true parameter value. The estimated posterior mean is located at parameter values focus duration 250 ms, selection delay 0.29 s, and menu recall probability 0.53.

The marginal distribution over individual parameter values in the posterior sample is visualised in Figure 9.4. We observe that the comparison between observed and simulated data has narrowed down the focus duration and selection delay parameter distributions compared to the prior (Figure 9.3). This means that some parameter values that were considered in the prior distribution have not been able to explain the observed data. The posterior distribution over menu recall probability is close to the prior distribution, which indicates that we were not able to extract much new information about this parameter based on the observed data.

Figure 9.5 shows the marginal distributions calculated over parameter pairs. Here we can see that the posterior captures a trade-off between the focus du-

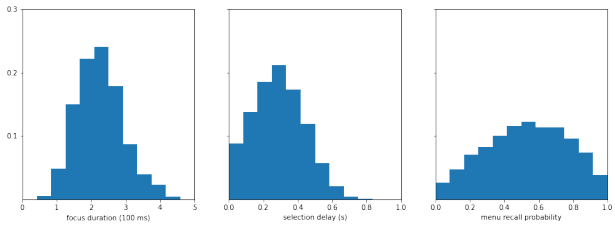

Figure 9.4 Marginal posterior distributions over individual menu search model parameters.

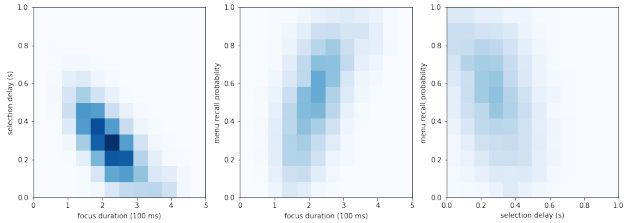

Figure 9.5 Marginal posterior distributions over menu search model parameter pairs.

ration and selection delay parameters that both contribute to the observed task times. This has been learned based on the observed data, since the prior distributions did not encode correlation between parameters, and it means that while we cannot be too certain about the exact parameter values, which is observed as variance in the individual marginal distributions, we can narrow down the most probable combinations for focus duration and selection delay.

9.3.2 Individual parameter estimation of a driving model

Our second example demonstrates estimation of person-specific parameters to a cognitive simulator. In comparison to the previous example, where each summary statistics was computed as an aggregate over all participants, the goal of individual parameter estimation is to capture the idiosyncracies of a single user. A cognitive simulator, if correctly parametrised to an individual, can be used to explore how the user would react to different interfaces or changes in the task circumstances. However, especially when making decisions on individual basis, it is important to be able to assess the confidence that can be placed on

the model's predictions. To that end, we demonstrate here how the posterior produced by Bayesian parameter estimation can be used to investigate how probable certain user behaviours are, as predicted by the model.

The task model

We utilise a model of driving that has been used in simulating how drivers share visual attention between the driving and an in-car search task when multitasking [17]. The model is based on a similar idea as the menu search model in the previous example: in-car glancing behaviour is assumed to emerge as an adaptation to task and cognitive constraints. For instance, as the speed of the car increases, in-car glances become shorter because of the increased visual demand of the driving task. Such an adaptive model can be used to predict how circumstances of driving, the design of the in-car interface, and the abilities of the user impact driving safety.

The original study did not make any attempts at fitting the parameters of the model to human data; instead, they were set by observing the model's behaviour and determining a parameter value that produced simulations that on the face value looked realistic. However, the model includes multiple parameters that the authors discuss might vary between drivers and use cases. Here, we focus on one, action related noise σ. This parameter dictates how precise the driver's steering movements are, that is, how accurately the car is controlled. Larger parameter values result in more swayed driving, as the driver needs to take small corrective actions to control for the noisy steering. Similarly to the original paper [17], we hypothesise here that the model's driving noise parameter can be varied in order to simulate variance in lateral stability between individual drivers. However, contrary to the original study, we provide empirical support to this evidence by applying Bayesian parameter inference.

Parameter estimation

We describe our prior expectations about the action related noise parameter σ as a Gamma distribution $P(\sigma) = \Gamma(3, 0.5)$, illustrated in Fig 9.6. The intuition behind this definition is that we assume the following: (1) every individual's action related noise is greater than 0, due to inherent noise in the human motor system; (2) we expect that this noise is distributed fairly normally around its mode value, but with a long right tail to accommodate for individuals with a lot of action related noise, such as elderly drivers or those with motor impairments; (3) we fix the mode of the distribution to be at 1.0, because tests with the driving model indicate that, on the face value, the resulting predicted behaviour seems a reasonable approximation of normal driving. It is important to note that none of these points, with the exception of the first one, are strongly

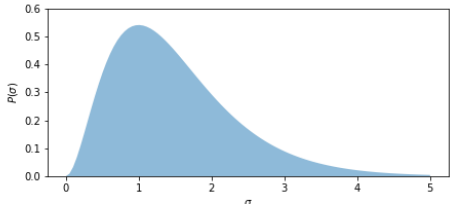

Figure 9.6 The gamma distribution used as a prior in the case example.

grounded in existing research. Regardless, they provide information, even if weak, to the parameter estimation procedure. Furthermore, this process makes explicit our assumptions about the prior, permitting future revisions of it, based on more evidence and reasoning.

The human data used in this example are taken from the original study[1]. The experiment proper had 12 participants conducting in-car visual search while driving the car. In addition, there was a practice session at the start of the experiments, where the participants got comfortable with the controls of the car simulator by driving a slightly curving road. The experiment had two speed conditions, with the car's speed fixed to either 60 km/h or 120 km/h, and so the practice session had these two speeds as well. Here, we use the data from the practice session only, and attempt to fit the driving model's σ parameter to individual drivers based on observing a few minutes of driving with both speeds used.

Both the driving simulator used in the behavioural study and the driving model studied in this experiment generate detailed time series data of the car's lateral position on the road. The data are snapshots of the state of the driving simulator or the model in intervals of 150 ms. Since the behaviour studied in this example is driving stability of individual participants, we use standard deviation of lateral road position to summarise the observations. It is simple to compute from both simulated and observed data, and provides an intuitive measure of the amount of instability in the driving. As the study with human participants included two conditions of fixed speed, we compute the summary statistic twice. The standard deviations of lateral offset calculated based on individual observations ranged between 0.11–0.25 in the 60 km/h condition and 0.16–0.40 in the 120 km/m condition. The standard deviations are shown in Figure 9.7.

[1] The data are available at `https://gitlab.com/jokinenj/multitasking-driving`.

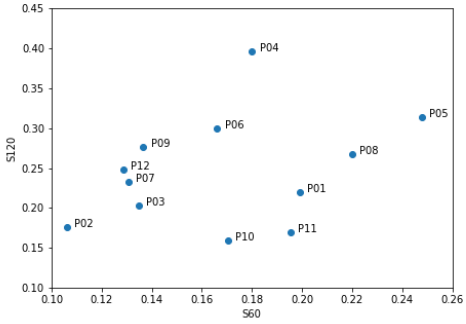

Figure 9.7 Observed individual summary statistics.

Fitting parameters on individual basis can be computationally expensive in case there are several individuals to model and the whole inference process is repeated with each observation set. The present experiment uses the LFIRE method [38] available as an ELFI extension PYLFIRE [22]. This is a density-ratio estimation method that uses classification between simulated datasets to learn the ratio between likelihood and marginal likelihood values. The method calculates approximate posterior probabilities at predetermined parameter values. Here the values were selected at 0.1 interval between 0 and 3. Gaussian process classifiers were trained as the likelihood-ratio model using 100 simulations with each parameter value and 100 simulations with parameters sampled from the prior distribution. The model was then used to calculate approximate posterior probabilities based on all individual observations. This means that while the model training took time, individual posteriors could be calculated based on the model without extra simulation costs.

Results and discussion

The posterior distributions estimated based on individual observations are each concentrated around mean values between 1.0–1.3. The mean values are recorded in Table 9.2. We use each individual posterior distribution to sample $N = 100$ parameter values and run simulation experiments to determine the posterior predictive distribution over driver behaviours. Selected sample distributions over σ and predicted offset in car position are shown in Figure 9.8. The posteriors have clear centres that are at different parameter values. This indicates that the parameter inference procedure was able to capture individual differences in driving behaviour via the action noise parameter σ. We also see that

Table 9.2 *Estimated individual posterior mean values.*

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12
$\hat{\sigma}$	1.2	1.0	1.2	1.3	1.3	1.2	1.2	1.2	1.2	1.1	1.1	1.2

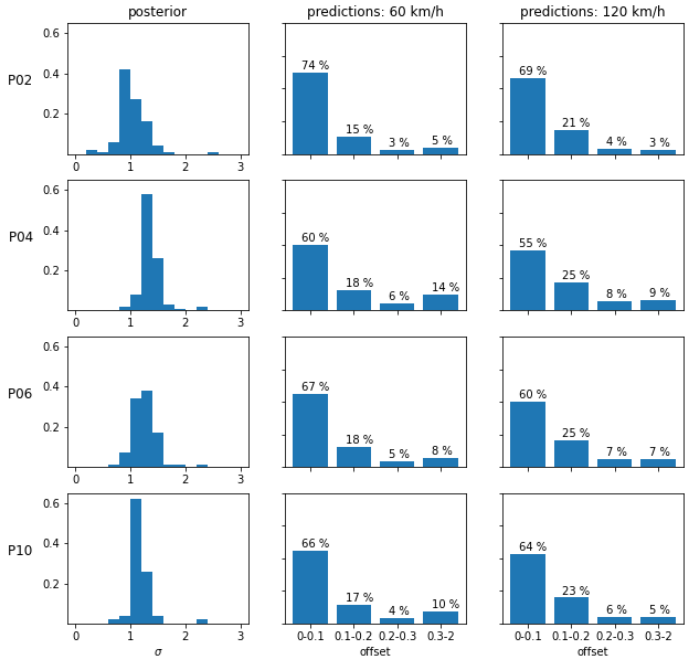

Figure 9.8 Selected posterior and posterior predictive distributions.

some observations have been more informative about the unknown parameter value than others, as there are differences in the posterior variance. Moreover, individual differences between the participants' action noise values are now visible as differences in the predicted offset in car position.

The posterior and posterior predictive distributions permit the modeller to consider the plausibility of different parameter values and thus to assess confidence in predictions. For instance, if the interest is in deploying individually tuned driving aids, such as a lane departure warning system, the model can

be used to predict on an individual basis when to signal a warning such that the driver has enough time to respond to it. In such an application, it is important that decisions are made based on all the available information and that uncertainties are also taken into account.

9.4 Discussion

Computational cognitive models make it possible for HCI designers and researchers to better understand users by providing psychologically realistic simulations of interactive behaviour. This chapter discussed the role of parameter inference in using such simulators in HCI. Manipulating the parameters of a model makes it flexible, permitting quick prototyping of different interactive scenarios, and predicting behaviour of individuals with various abilities and goals, but only insofar as the parameters are set correctly and informatively. Especially important this is in cases where noisy and sparse observations of user-generated data are used in inferring the values of the parameters. Bayesian parameter inference is attractive in this regard as it allows the modeller to take into account prior information, and provides a posterior distribution over parameter values.

While the focus of this chapter is on parameter inference, parameters themselves are rarely the final interest of research. The practical purpose of cognitive modelling in HCI is to make predictions based on parametrising these models. How much a certain parameter varies within the user population is therefore not as pertinent question for UI design as is the range of *actual behaviours* that can be predicted (and observed) based on the distribution of the parameter in question. When these parameters are inferred, it makes sense to provide the whole range of plausible predictions, given what is known about the parameter prior to parameter estimate and after outcomes of simulations with different parameter values are compared to observed data. Bayesian parameter inference facilitates this process by formalising how the posterior parameter distribution is inferred on the basis of a prior and evidence. The posterior can then be sampled for parameter values, which are used to parametrise the simulator to generate a range of predictions called the posterior predictive distribution. For instance, in our example above about inferring the action related noise parameter of individual drivers, the value of the parameter was only of secondary importance compared to predictions of that driver's behaviour (i.e., variability in lane offset). When such concrete predictions can be made from individual users, it becomes feasible to run the simulator under various

task circumstances and thus generate a range of possible future scenarios to inform decision making.

Bayesian parameter inference is also powerful tool to analyse model flexibility. The problem of overly flexible cognitive models is that if the free parameters of the model can be adjusted such that the model produces any kind of behaviour, then the model's fit to observation data is not a persuasive argument in support of the validity of the model [34]. Solutions to this include determining the predictions of the model over the whole range of the parameter space, accounting for the variability of the data, and showing that the model's predictions are restricted in some sense. Bayesian parameter inference allows for all of these in a formalised manner. The posterior predictive distribution is an intuitive way to investigate the whole range of a model's predictions, and because generated in light of prior assumptions about the parameter, such as its plausible range, and the observed data, including their variability, the posterior can be used to investigate and demonstrate model flexibility. Especially in cases where the predictions generated by cognitive models are used to make critical decisions, it is important to be able to assess our confidence in the model's predictions and its validity.

Likelihood-free inference provides the practical tools to estimate posterior probabilities over simulator model parameters. The most accessible to new practitioners is the basic ABC sampler. Here candidate parameters are sampled from the prior distribution and parameters that produce simulated user interactions similar to the observed interactions are accepted in the posterior. This method is not complicated and works when the estimation problem is low-dimensional and the individual simulations are not too expensive to compute. The more advanced methods that use previous simulations to decide the next candidate parameters can reduce the total simulation count a lot, but have more parameters that need to be controlled. The same applies to methods that learn a density or density-ratio model based on simulated data. These methods are attractive because the same model can be reused when we want to estimate posterior probabilities based on new observations, but again there is the need to control more parameters and to evaluate model fit to ensure a reliable estimation outcome. Overall the various likelihood-free methods make posterior estimation feasible for diverse simulation models and applications, and provide also means to achieve related tasks like comparison between alternative simulators [7].

The choice of summary statistics is critical for useful parameter inference. In traditional experimental research, where participants conduct well-specified tasks in controlled environments, the choice of dependent variables and their summarisation is largely dependent on the nature of this environment. Often,

an individual task is short, and is repeated multiple times by the same participant for the purposes of obtaining reliable estimates of the variable of interest under varying task conditions. Summary statistics, such as mean and standard deviation of the dependent variable, are then used to evaluate how the selected experimental manipulations impact the dependent variables. While this paradigm is useful for research under controlled settings, it is not necessary applicable for real-life parameter inference, where observation data may be noisy and sparse, and no clear task boundaries can be specified. In our example of inferring individual action related noise during driving, we used a statistic that describes variability of the car's lateral position over the whole driving episode. In choosing the correct summary statistics for parameter inference, attention must also be paid to the fact that summarisation might hide some important aspects of the interactive process being simulated. For instance, an average number of errors during a particular interaction may hide whether the errors are clustered around some critical moment or spread evenly throughout the episode. Furthermore, the modeller must pay mind to the fact that informative parameter inference is only possible insofar variation in the parameters inferred causes variation in the summarisation of the simulated predictions.

Many of the elements of parameter inference discussed here, such as efficiency, analysis of model flexibility, and selection of summary statistics, are of importance when discussing the future use of cognitive simulators to facilitate adaptation of interactive systems. While computational optimisation of user interfaces is a promising area of research [27], the quality of such optimisation is dependent on a careful specification of an objective function which tells how acceptable a solution is. Predictions made by simulators can provide flexible objective functions, but as we have argued, the quality of this is highly dependent on how the parameters of the simulator model are set. Especially important this becomes when optimising interfaces for individual users, with a focus on certain user abilities or idiosyncracies [36]. The ability to pre-train an approximate posterior model and then quickly conduct parameter inference, as showcased in our second example, could offer breakthroughs in online, human-in-the-loop optimisation and adaptation.

References

[1] G. Bailly, A. Oulasvirta, D. P. Brumby, and A. Howes. Model of visual search and selection time in linear menus. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, pages 3865–3874. ACM, 2014.

[2] J. P. Borst, N. A. Taatgen, and H. van Rijn. What makes interruptions disruptive?

a process-model account of the effects of the problem state bottleneck on task interruption and resumption. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 2971–2980, 2015.

[3] M. D. Byrne and A. Kirlik. Using computational cognitive modeling to diagnose possible sources of aviation error. *The international journal of aviation psychology*, 15(2):135–155, 2005.

[4] X. Chen, G. Bailly, D. P. Brumby, A. Oulasvirta, and A. Howes. The emergence of interactive behavior: A model of rational menu search. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4217–4226. ACM, 2015.

[5] K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *PNAS*, 2020.

[6] V. Di Pasquale, S. Miranda, R. Iannone, and S. Riemma. A simulator for human error probability analysis (sherpa). *Reliability Engineering & System Safety*, 139:17–32, 2015.

[7] X. Didelot, R. G. Everitt, A. M. Johansen, and D. J. Lawson. Likelihood-free estimation of model evidence. *Bayesian Analysis*, 6:49–76, 2011.

[8] C. Durkan, I. Murray, and G. Papamakarios. On contrastive learning for likelihood-free inference. 2020.

[9] C. R. Fisher, J. W. Houpt, and G. Gunzelmann. Developing memory-based models of ACT-R within a statistical framework. *Journal of Matchematical Psychology*, 98, 2020.

[10] W.-T. Fu and P. Pirolli. Snif-act: A cognitive model of user navigation on the world wide web. *Human–Computer Interaction*, 22(4):355–412, 2007.

[11] C. Gebhardt, A. Oulasvirta, and O. Hilliges. Hierarchical reinforcement learning explains task interleaving behavior. *Computational Brain & Behavior*, 2020.

[12] D. Greenberg, M. Nonnenmacher, and J. Macke. Automatic posterior transformation for likelihood-free inference. volume 97 of *PMLR*, pages 2404–2414, 2019.

[13] M. U. Gutmann and J. Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47, 2016.

[14] T. Halverson and A. J. Hornof. A computational model of "active vision" for visual search in human–computer interaction. *Human–Computer Interaction*, 26(4):285–314, 2011.

[15] Y. Huang, C. White, H. Xia, and Y. Wang. A computational cognitive modeling approach to understand and design mobile crowdsourcing for campus safety reporting. *International Journal of Human-Computer Studies*, 102:27–40, 2017.

[16] M. Järvenpää, M. U. Gutmann, A. Pleska, A. Vehtari, and P. Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *Bayesian Analysis*, 14(2):595–622, 2019.

[17] J. P. Jokinen, T. Kujala, and A. Oulasvirta. Multitasking in driving as optimal adaptation under uncertainty. *Human Factors*, page article 0018720820927687, 2020.

[18] J. P. Jokinen, S. Sarcar, A. Oulasvirta, C. Silpasuwanchai, Z. Wang, and X. Ren. Modelling learning of new keyboard layouts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4203–4215. ACM, 2017.

[19] J. P. Jokinen, Z. Wang, S. Sarcar, A. Oulasvirta, and X. Ren. Adaptive feature guidance: Modelling visual search with graphical layouts. *International Journal of Human–Computer Studies*, 136:102376, 2020.

[20] A. Kangasrääsiö, K. Athukorala, A. Howes, J. Corander, S. Kaski, and A. Oulasvirta. Inferring cognitive models from data using approximate Bayesian computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1295–1306. ACM, 2017.

[21] A. Kangasrääsiö, J. P. Jokinen, A. Oulasvirta, A. Howes, and S. Kaski. Parameter inference for computational cognitive models with approximate Bayesian computation. *Cognitive Science*, 43(6):e12738, 2019.

[22] J. Kokko, U. Remes, O. Thomas, H. Pesonen, and J. Corander. PYLFIRE: Python implementation of likelihood-free inference by ratio estimation. *Wellcome Open Research*, 4:197, 2019.

[23] J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Fundamentals and recent developments in approximate Bayesian computation. *Systematic Biology*, 66(1):e66, 2017.

[24] J. Lintusaari, H. Vuollekoski, A. Kangasrääsiö, K. Skytén, M. Järvenpää, P. Marttinen, M. U. Gutmann, A. Vehtari, J. Corander, and S. Kaski. ELFI: Engine for Likelihood-Free Inference. *Journal of Machine Learning Research*, 19(16):1–7, 2018.

[25] D. J. Nott, V. M.-H. Ong, Y. Fan, and S. A. Sisson. High-dimensional ABC. In S. A. Sisson, Y. Fan, and M. A. Beaumont, editors, *Handbook of approximate Bayesian computation*, pages 211–241. CRC Press, 2019.

[26] M. O'Brien and M. T. Keane. Modeling user behavior using a search-engine. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 357–360, 2007.

[27] A. Oulasvirta, N. R. Dayama, M. Shiripour, M. John, and A. Karrenbauer. Combinatorial optimization of graphical user interface designs. *Proceedings of the IEEE*, 108(3):434–464, 2020.

[28] G. Papamakarios, D. Sterratt, and I. Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. volume 89 of *PMLR*, pages 837–848, 2019.

[29] P. Pirolli, G. M. Youngblood, H. Du, A. Konrad, L. Nelson, and A. Springer. Scaffolding the mastery of healthy behaviors with fittle+ systems: evidence-based interventions and theory. *Human–Computer Interaction*, pages 1–34, 2018.

[30] C. Pompanon and É. Raufaste. The intervention trigger model: Computational modelling of air traffic control. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31, 2009.

[31] D. Prangle. Summary statistics. In S. A. Sisson, Y. Fan, and M. A. Beaumont, editors, *Handbook of approximate Bayesian computation*, pages 125–152. CRC Press, 2019.

[32] L. F. Price, C. C. Drovandi, A. Lee, and D. J. Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27:1–11, 2018.

[33] F. Putze, M. Salous, and T. Schultz. Detecting memory-based interaction obstacles with a recurrent neural model of user behavior. In *23rd International Conference on Intelligent User Interfaces*, pages 205–209, 2018.

[34] S. Roberts and H. Pashler. How persuasive is a good fit? a comment on theory testing. *Psychological review*, 107(2):358, 2000.

[35] D. D. Salvucci, M. Zuber, E. Beregovaia, and D. Markley. Distract-r: Rapid prototyping and evaluation of in-vehicle interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 581–589, 2005.

[36] S. Sarcar, J. P. Jokinen, A. Oulasvirta, Z. Wang, C. Silpasuwanchai, and X. Ren. Ability-based optimization of touchscreen interactions. *IEEE Pervasive Computing*, 17(1):15–26, 2018.

[37] S. A. Sisson, Y. Fan, and M. A. Beaumont, editors. CRC Press, 2019.

[38] O. Thomas, R. Dutta, J. Corander, S. Kaski, and M. U. Gutmann. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, Advance publication, 2020.

[39] J. G. Trafton, M. D. Bugajska, B. R. Fransen, and R. M. Ratwani. Integrating vision and audition within a cognitive architecture to track conversations. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 201–208, 2008.

[40] B. M. Turner and P. B. Sederberg. A generalized, likelihood-free method for posterior estimation. *Psychonomic bulletin & review*, 21(2):227–250, 2014.

[41] B. M. Turner and T. Van Zandt. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56(2):69–85, 2012.

[42] S. Vasishth. Using approximate Bayesian computation for estimating parameters in the cue-based retrieval model of sentence processing. *MethodsX*, 2020.

[43] S. N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.