Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. 135

Approximate Bayesian Inference for Psychometric Functions using MCMC Sampling

Malte Kuss[1], Frank Jäkel[2], Felix A. Wichmann[3]

January 2005

[1] Department Schölkopf, email: malte.kuss@tuebingen.mpg.de
[2] Department Schölkopf, email: frank.jaekel@tuebingen.mpg.de
[3] Department Schölkopf, email: felix.wichmann@tuebingen.mpg.de

This report is available in PDF–format via anonymous ftp at ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/TR-135.pdf. The complete series of Technical Reports is documented at: http://www.kyb.tuebingen.mpg.de/techreports.html

Approximate Bayesian Inference for Psychometric Functions using MCMC Sampling

Malte Kuss, Frank Jäkel, Felix A. Wichmann

Abstract. In psychophysical studies the psychometric function is used to model the relation between the physical stimulus intensity and the observer's ability to detect or discriminate between stimuli of different intensities. In this report we propose the use of Bayesian inference to extract the information contained in experimental data estimate the parameters of psychometric functions. Since Bayesian inference cannot be performed analytically we describe how a Markov chain Monte Carlo method can be used to generate samples from the posterior distribution over parameters. These samples are used to estimate Bayesian confidence intervals and other characteristics of the posterior distribution. In addition we discuss the parameterisation of psychometric functions and the role of prior distributions in the analysis. The proposed approach is exemplified using artificially generated data and in a case study for real experimental data. Furthermore, we compare our approach with traditional methods based on maximum-likelihood parameter estimation combined with bootstrap techniques for confidence interval estimation. The appendix provides a description of an implementation for the R environment for statistical computing and provides the code for reproducing the results discussed in the experiment section.

---

1 Introduction

Psychophysics explores the connection between physical stimuli and subjective responses. The psychometric function relates the stimulus intensity ("physics") on the abscissa with the observer's response ("psychology") on the ordinate and is the central function in the analysis of data obtained from psychophysical studies. This is true not only in classical psychophysical settings in experimental psychology but equally true in clinical or developmental studies where the datasets are typically even smaller, and thus proper statistical procedures are even more important. It is also true in awake-behaving neurophysiology studies where the datasets may be larger but the problem of stimulus-independent errors or "lapses" (Wichmann and Hill, 2001a) may be more pronounced.

Given that psychophysical experiments tend to be time consuming and tiring for the observers, many methods have been developed to estimate only a single point of the psychometric function, typically a point in the interval between 50% and 90% correct performance termed the *threshold*. These so-called *adaptive methods* vary the stimulus strength based on previous responses of the observer; adaptive methods can be divided into non-parametric (Wetherill and Levitt, 1965; Rose et al., 1970; Taylor, 1971; Garcia-Perez, 1998) and parametric (Pentland, 1980; Watt and Andrews, 1981; Watson and Pelli, 1983; Madigan and Williams, 1987; Pelli, 1987; King-Smith and Rose, 1997; Snoeren and Puts, 1997; Kontsevich and Tyler, 1999; Alcalá-Quintana and Garcia-Pérez, 2004), the latter including some methods that are explicitly Bayesian (Kontsevich and Tyler, 1999; Alcalá-Quintana and Garcia-Pérez, 2004). For a review of some of the most common adaptive methods the paper by Treutwein (1995) is recommended.

However, in many cases it is important not only to know a single point of the psychometric function but to estimate it in its entirety. Differences between experimental conditions may not lead to different threshold values but the slope of the psychometric functions could have changed significantly (Green and Swets, 1966; Wichmann, 1999). In principle all the trials taken during a run of an adaptive method could be used to estimate the complete psychometric function, but this is not recommended as the sampling,

optimised to estimate a single point only, is sub-optimal for complete function estimation (Kaernbach, 2001).

There exists a fairly comprehensive literature on estimating the psychometric function (e.g. O'Regan and Humbert (1989); Treutwein and Strasburger (1999); Klein (2001)), with some papers additionally covering sampling issues and goodness-of-fit (e.g. Wichmann and Hill (2001b,a)) or non-parametric estimation methods (e.g. Miller and Ulrich (2001)). Comparatively few papers, however, have investigated how to obtain reliable confidence intervals for the estimated parameters of psychometric functions (Finney, 1971; McKee et al., 1985; Maloney, 1990; Foster and Bischof, 1991, 1997; Wichmann and Hill, 2001b,a). There appears to exist a general consensus, however, that bootstrap methods offer more reliable confidence-intervals than methods based on asymptotic considerations due to small datasets typical in psychophysical research (between 50 and 1000 trials per psychometric function). In this paper we present experiments indicating that Bayesian inference methods are superior to bootstrap-based methods and are thus the method of choice for estimating the psychometric function and its confidence-intervals.

2 The Binomial Mixture Model

In this section we formally derive a basic statistical model of the process that generates the data. The object of interest is a *parametric psychometric function* $F(x, \boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta}$ which maps the stimulus intensity x to the $[0, 1]$ interval. This function is commonly chosen to have a sigmoidal form like cumulative density functions of various probability distributions. We will discuss several common choices in Section 4.

The psychometric function relates the observers response to stimulus intensity. In an nAFC experimental setting there is a *chance probability* π_c that the observer "guesses" the correct answer independent of the stimulus. This probability of making the correct guess is usually $\pi_c = \frac{1}{n}$ where n is the number of possible choices (the n in nAFC). In a long sequence of experimental trials the observer occasionally *lapses*, i.e. makes a random choice independent of the stimulus. In vision experiments an obvious example is blinking while the stimulus is presented. This probability of lapsing π_l is a nuisance parameter but it is necessary to take its effect into account in statistical modelling as shown by Wichmann and Hill (2001a,b).

We now have all quantities for a basic model to relate the psychometric function F to the probability of giving the correct answer in a single nAFC stimulus presentation. Given the stimulus intensity x the event of correct discrimination is a Bernoulli variable with probability of success equal to

$$\Psi(x, \boldsymbol{\theta}, \pi_c, \pi_l) = (1 - \pi_l)\left[(1 - \pi_c)F(x, \boldsymbol{\theta}) + \pi_c\right] + \pi_c\pi_l \tag{1}$$

where $F(x, \boldsymbol{\theta})$ characterises the change of discriminability as a function of the stimulus intensity. The model comes in the form of a mixture of two Bernoulli distributions, which is again a Bernoulli distribution. With probability π_l the observer lapses and has chance π_c to guess the correct answer. With probability $(1 - \pi_l)$ the observer does not lapse and has a chance of $(1 - \pi_c)F(x, \boldsymbol{\theta}) + \pi_c$, which is $F(x, \boldsymbol{\theta})$ scaled to the $[\pi_c, 1]$ interval, to give the correct answer.

The psychophysical experiment can be seen as a sequence of such Bernoulli trials. Often only a small number $\{x_1, \ldots, x_k\}$ of distinct stimulus intensities are used in an experiment which allows a more compact representation. Aggregating the trials for identical stimulus intensities we compress the data to a set of triples $\mathcal{D} = \{(x, N, n)_i | i = 1, \ldots k\}$ such that at contrast x_i we conducted N_i trials and observed

n_i correct responses. Since n_i is a sum of Bernoulli variables it has a binomial distribution

$$p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c) = \prod_{i=1}^{k} p(n_i|N_i, x_i, \boldsymbol{\theta}, \pi_l, \pi_c) \tag{2a}$$

$$= \prod_{i=1}^{k} \text{Binomial}(n_i|N_i, \Psi(x_i, \boldsymbol{\theta}, \pi_l, \pi_c)) \tag{2b}$$

where Ψ is given by (1). Equations (2) describe the assumed generative model of the data, i.e. the *sampling distribution*. Furthermore, read as a function of $\boldsymbol{\theta}$ and π_l for observed $\mathcal{D}$ we refer to it as the *likelihood* of the binomial mixture model.

3 Bayesian Inference for Psychometric Functions

In this section we describe how the data collected in psychophysical experiments can be used to do Bayesian *inference* about the parameters $\boldsymbol{\theta}$ of a psychometric function $F(x, \boldsymbol{\theta})$ and the lapse probability π_l.

At first we give a general description how inference is performed in the Bayesian framework, using a simplified notation. Starting point is a model of the process of how the data that we can observe is generated. Let $p(\mathcal{D}|\boldsymbol{\phi})$ be a statistical description of this model where $\mathcal{D}$ denotes observable data and $\boldsymbol{\phi}$ are model parameters. In a nutshell, the problem is that the *true* generating parameter $\boldsymbol{\phi}^*$ is hidden, but by observing data we can reduce our uncertainty about its value. In the Bayesian framework probability distributions over parameter values are used to describe beliefs and uncertainties about the parameter value in the data generating process.

The *prior* distribution $p(\boldsymbol{\phi})$ represents beliefs about the value of the true parameter $\boldsymbol{\phi}^*$ previous to an inference step. By inference we refer to the process of integrating the information contained in observed data $\mathcal{D}$ and the prior $p(\boldsymbol{\phi})$ into a *posterior* distribution $p(\boldsymbol{\phi}|\mathcal{D})$. The posterior is obtained according to Bayes' rule

$$p(\boldsymbol{\phi}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\phi})p(\boldsymbol{\phi})}{p(\mathcal{D})} \ . \tag{3}$$

This can be understood as a weighting in which prior beliefs about $\boldsymbol{\phi}^*$ are weighted proportionally to their compatibility with the observed data. The weighting is given by the likelihood function, which is $p(\mathcal{D}|\boldsymbol{\phi})$ as a function of $\boldsymbol{\phi}$ for given $\mathcal{D}$. Prior and posterior are probability distributions describing two states relative to an inference step and correspond to potentially different beliefs about the value of the parameter that generated the data. For details the reader is referred to O'Hagan (1994) and Jaynes (2003), to mention only two textbooks on Bayesian statistics.

We now describe how this framework can be applied to infer something about the parameters of psychometric functions. In the following we assume the data is generated according to the binomial mixture model for some specific parametric type of $F(x, \boldsymbol{\theta})$. In psychometric studies data $\mathcal{D} = \{(x, N, n)_i | i = 1, \dots k\}$ is collected in order to learn about $\boldsymbol{\theta}$ and π_l. Again Bayes' rule describes how the observed data consistently reduces the uncertainty about the underlying value of $\boldsymbol{\theta}$ and π_l. Formally the posterior is obtained according to Bayes' rule

$$p(\boldsymbol{\theta}, \pi_l|\mathcal{D}, \pi_c) = \frac{p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c)p(\boldsymbol{\theta})p(\pi_l)}{\int p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c)p(\boldsymbol{\theta})p(\pi_l)d\boldsymbol{\theta}d\pi_l} \tag{4}$$

where $p(\boldsymbol{\theta})$ and $p(\pi_l)$ are prior distributions, $p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c)$ acts as the likelihood, and $p(\boldsymbol{\theta}, \pi_l|\mathcal{D}, \pi_c)$ is the posterior. The posterior distribution summarises all information contained in the observations and the prior about $\boldsymbol{\theta}$ and π_l. Unfortunately solving the integral in the denominator appears to be analytically

intractable such that the posterior cannot be computed in closed form. Even if we could compute the posterior, the distribution would be of a non-standard type and we would be unable to work with it analytically. We therefore have to use approximative techniques to describe the information presented by the posterior.

3.1 Point Estimates of Parameters and Confidence Intervals

The most simple approximation to the information represented by the posterior distribution is to state a single point estimate of the true parameter values. In the Bayesian framework choosing a point estimate is considered a *decision* problem in which the decision maker minimises an expected risk, where the expectation is taken with respect to the posterior distribution (Jaynes, 2003, ch. 13.9). The risk function characterises the *loss* associated with a discrepancy between the point estimate and the unknown true parameter value. For example, the expected absolute error is minimised by the median of the posterior distribution (MED). Likewise minimising the expected squared error leads to choosing the mean of the posterior (MEAN).

The mode of the posterior, which will be referred to as the *maximum a posteriori* (MAP) estimate, is obtained for a loss function which is zero if the estimate and the true value match exactly and 1 otherwise. In the binomial mixture model, assuming the psychometric function is differentiable, gradient based methods can be used to find the MAP point estimate

$$(\boldsymbol{\theta}, \pi_l)^{\text{MAP}} = \underset{\boldsymbol{\theta},\pi_l}{\operatorname{argmax}}\; p(\mathcal{D}|\boldsymbol{\theta}, \pi_l) p(\boldsymbol{\theta}) p(\pi_l)\ . \tag{5}$$

If the prior $p(\boldsymbol{\theta})p(\pi_l)$ is taken to be constant, i.e. a flat prior, the *maximum likelihood* (ML) estimator is derived as a special case.

Another simple approximation technique is *Laplace's* method by which the posterior is approximated by a Gaussian distribution which is found by a second order Taylor expansion around the mode. This method is applicable in the proposed setting but the approximation might be poor. An obvious drawback is that the approximation is symmetric and fitted locally around the mode but can be poor in approximating the tails of the posterior. We therefore do not consider this method in the following.

The posterior distribution represents the remaining uncertainty after having seen the data. An obvious problem is that any notion of uncertainty or confidence is lost when only point estimates are stated. A convenient way of expressing how narrowly a parameter is determined is to state *confidence regions*. Given a confidence level $\gamma \in (0, 1)$, the notion of a confidence region is conceptually different in frequentist and Bayesian statistics (see e.g. DeGroot and Schervish (2002, ch. 7)).

From a Bayesian perspective it is valid to define a γ confidence region simply as a region in which the true parameter values are believed to lie in with probability γ. This can be stated because the parameters are random variables and we can express our degree of belief for any statement regarding the parameters by evaluating the statement under the posterior distribution.

In frequentist statistics confidence regions are constructed and interpreted differently. In this setting the region itself is a random variable which contains the true parameter value with probability γ. This means that if the experiment is repeated infinitely many often γ percent of the computed regions would contain the true value. For a particular data set it is not possible to state a probability assignment that the true parameter lies in a computed confidence interval.

In case the distribution of an estimator cannot be computed analytically, a common strategy in frequentist statistics is to compute approximate confidence intervals using *bootstrap* methods (Efron and Tibshirani, 1993). The basic idea is to repeatedly generate artificial data sets. For each artificial data set the parameters are re-estimated and the variability of these estimates is used to estimate confidence intervals. In *parametric* bootstrap methods artificial data sets are generated from the model using maximum likelihood estimates of the parameters. For psychometric functions this approach has been described by Wichmann and Hill (2001b).

From a Bayesian perspective parametric bootstrap methods exhibit several conceptual flaws. The experimental data enters the bootstrap analysis only through the maximum likelihood estimator. How tightly the data determines the ML estimator is disregarded and instead it is subsequently taken as to be the *true* generating value, regardless of how informative the data was. Afterwards the bootstrap samples are used to regain a notion of variability, i.e. by how much the ML estimate could have been erroneous, given how the function was sampled. From a Bayesian point of view the posterior represents the uncertainty about the true parameter values and should therefore be used to make confidence statements. Reducing the information contained in the data to a ML estimate is counter-productive. The data is observed, therefore not a random variable, and generating artificial data samples from a model does not produce any new information about the parameters.

Approximations using sampling methods are also common in Bayesian statistics in situations in which the posterior cannot be computed analytically. The difference is that in the Bayesian framework samples are generated from the posterior over parameters. This can be implemented using *Markov chain Monte Carlo* techniques.

3.2 Approximate Inference by Markov Chain Monte Carlo Sampling

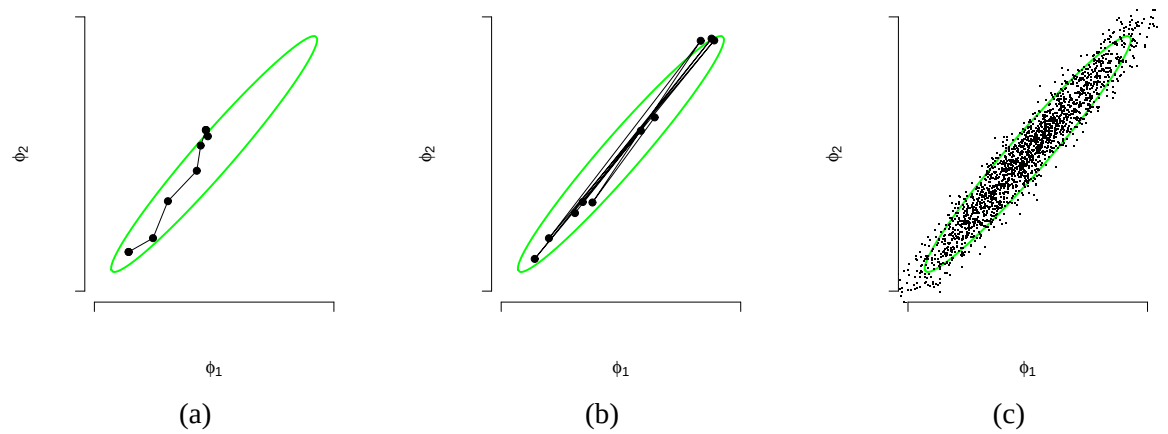
In this section we describe the basic idea of using Markov chain Monte Carlo (MCMC) methods for approximate Bayesian inference. For more technical introductions the reader is referred to MacKay (1999, 2003), while more comprehensive reviews can be found in Neal (1993) and Gilks and Richardson (1996).

Recall the simplified notation introduced in the previous section. Assume some data $\mathcal{D}$ has been observed and we want to compute the posterior according to Bayes' rule (3). A common situation is that we can evaluate the likelihood $p(\mathcal{D}|\boldsymbol{\phi})$ and the prior $p(\boldsymbol{\phi})$ for every possible value of $\boldsymbol{\phi}$ but we cannot compute or work with the posterior analytically. MCMC methods sidestep this problem by generating samples from the posterior $p(\boldsymbol{\phi}|\mathcal{D})$ using only evaluations of the *unnormalised* posterior $q(\boldsymbol{\phi}|\mathcal{D}) = p(\mathcal{D}|\boldsymbol{\phi})p(\boldsymbol{\phi})$. The idea behind is that the samples characterise the posterior distribution sufficiently well. In particular, statistics of the samples can be used to approximate properties of the posterior distribution. For example, the mean of the samples is an approximation to the mean of the posterior distribution.

In order to generate a sample from the posterior a random sequence of parameter values $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n$ is generated such that the distribution of $\boldsymbol{\phi}_n$ asymptotically becomes identical to the posterior as the length of the sequence n increases. In the MCMC terminology the sequence is called a *chain* and each element is referred to as a *state*. In practice the chain is generated for a finite length n and the state $\boldsymbol{\phi}_n$ is interpreted as a sample of the posterior. The procedure is repeated until enough samples are obtained such that the characteristics of the posterior distribution can be well approximated by statistics of the generated samples.

For this mechanism to work the sequence has to be constructed in a particular way following the *Metropolis-Hastings* method which describes how the consecutive state is found. Assume $\boldsymbol{\phi}_t$ is the current state. In order to find a valid consecutive state $\boldsymbol{\phi}_{t+1}$ a candidate value $\tilde{\boldsymbol{\phi}}$ is proposed from a *proposal distribution* $p(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t)$, for example a Gaussian distribution centred at $\boldsymbol{\phi}_t$. The decision whether $\tilde{\boldsymbol{\phi}}$ is accepted as consecutive state depends on the ratio of $q(\boldsymbol{\phi}|\mathcal{D})$ evaluated at $\tilde{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}_t$. Ignoring some further technicalities, the intuition is that $\tilde{\boldsymbol{\phi}}$ is accepted if it yields a higher value, i.e. $q(\tilde{\boldsymbol{\phi}}|\mathcal{D}) > q(\boldsymbol{\phi}_t|\mathcal{D})$. Otherwise the probability of acceptance is proportional to the ratio of values under the unnormalised posterior. Since $\boldsymbol{\phi}_{t+1}$ depends only on $\boldsymbol{\phi}_t$ and not on the history of states the resulting chain is called a *Markov* chain. See Figure 1(a) for an example.

The computational efficiency of MCMC sampling method depends on how the consecutive state is proposed. While simulating the Markov chain, states that occur close-by in the chain are dependent through the proposal distribution. Refinements of this scheme are directed towards improved proposal



(a) (b) (c)

Figure 1: Illustration of MCMC sampling: The ellipse represents the contour of a posterior $p(\boldsymbol{\phi}|\mathcal{D})$ we want to sample from, i.e. which we want to approximate. Figure (a) shows a chain (after 10 proposed states) generated by Metropolis-Hastings sampling using local moves (the proposal distribution is Gaussian $p(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t) = \mathcal{N}(\tilde{\boldsymbol{\phi}}|\boldsymbol{\phi}_t, \boldsymbol{I}\alpha)$). States are depicted by points and consecutive states are connected by lines. Figure (b) shows a chain (after 10 proposed states) obtained using hybrid MCMC sampling. Note that the states appear to be less dependent while the number of accepted states is larger. Figure (c) shows 2000 samples generated using hybrid MCMC.

distributions such that this dependence is reduced. As an effect this also reduces the length of the sequence n after which the state can be considered an approximately independent sample of the posterior.

In the following we use *hybrid* Monte Carlo sampling which is also known as *Hamiltonian* sampling, as described by Neal (1993) and MacKay (2003, ch. 30). New states are proposed using a procedure that can be understood as a discrete simulation of Hamiltonian dynamics. The sampling scheme requires to set additional parameters, namely the number of steps (so called *leapfrog* steps) and the step sizes used in the discrete simulation. In Appendix A.3 we discuss some heuristics how to find these parameters in practice. Figure 1(b) illustrates the potential improvement gained by using this method.

The main idea of this paper is to use hybrid MCMC sampling to generate samples from the posterior (4) over the parameters of psychometric functions. Once we are convinced that the generated MCMC samples are representative for the posterior, they can be used to estimate certain characteristics of the posterior distribution. The empirical mean of the samples can be used as an estimate of the expectation of the posterior distribution (MEAN). Likewise the sample median is an approximation to the median of the posterior distribution (MED). The empirical quantiles of the samples can be interpreted as estimates of the quantiles of the posterior distribution. We refer to the interval between the $(1-\gamma)/2$ and $(1+\gamma)/2$ empirical quantiles of the samples as an *approximate Bayesian* γ *confidence interval* .

Before we present examples of this approach, the following section describes parameterisations of psychometric functions and the role of prior distributions in the analysis.

4 Parameterisation and Prior Distributions

In psychophysical practice the experimentalist has certain beliefs about the mechanism of interest, otherwise the experiment could not be designed. Expressing prior beliefs and parameterisation of the model go hand in hand. It is therefore advantageous to parameterise the model close to the way the scientist thinks about the mechanism it describes. In the following section we describe a convenient parameterisation of psychometric functions, before we discuss various forms of prior distributions on their parameters.

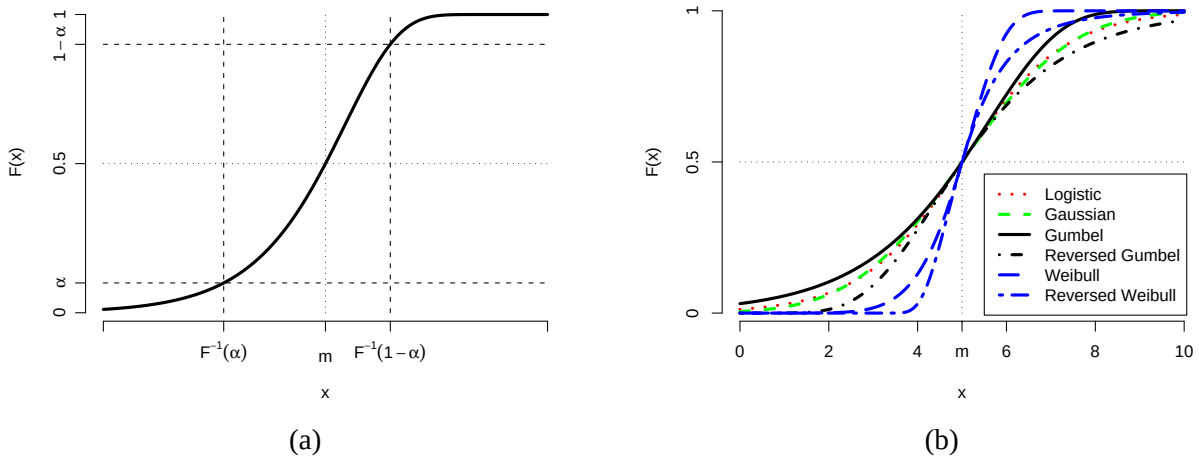


Figure 2: Types and parameterisation of psychometric functions: Figure (a) illustrates the parameterisation of psychometric functions in terms of threshold location m and width w between $F^{-1}(\alpha)$ and $F^{-1}(1-\alpha)$. The example shows a Gumbel function for $\alpha = 0.1$. Figure (b) exemplifies different types of psychometric function. The logistic, Gaussian and Gumbel functions are shown for $m = 5$ and $w = 5$. The Weibull functions are plotted for $m = 5$ and $s = 0.5$.

4.1 Parameterisation of the Psychometric Function

Let $F(x, \boldsymbol{\theta})$ be the psychometric function and F^{-1} its inverse. In the analysis of psychometric data a common interest is to locate the *threshold* $m = F^{-1}(0.5)$ and a range for which the detectability varies with the stimulus intensity. A common way of characterising the sensitivity of an observer is the slope of the psychometric function at the threshold location. Another way of describing the range of interest is the *width* w defined as $w = F^{-1}(1-\alpha) - F^{-1}(\alpha)$. This is the length of the interval between $F^{-1}(\alpha)$, the stimulus intensity at which $F(x, \boldsymbol{\theta}) = \alpha$, and $F^{-1}(1-\alpha)$ for some small α. As default we use $\alpha = 0.1$ (see Figure 2(a)). This parameterisation of the psychometric function in terms of threshold and width has been proposed by Alcalá-Quintana and Garcia-Pérez (2004). An advantage of this parameterisation is that w comes in the scale of the stimulus itself whereas the value of the slope is usually difficult to interpret. We now show how various common functions used to model F can be parameterised such that $\boldsymbol{\theta} = [m, w]$. Many of the function used to model F also appear in statistical Generalised Linear Models (GLMs) in which they are called response functions (Fahrmeier and Tutz, 2001; Collet, 1991).

The logistic function, which is called *logit* response function in GLMs, can be parameterised as

$$F_{\text{logistic}}(x, \boldsymbol{\theta}) = \left(1 + \exp\left(-\frac{z(\alpha)}{w}(x - m)\right)\right)^{-1} \tag{6}$$

where $z(\alpha) = 2\ln(\frac{1}{\alpha} - 1)$. The function is point symmetric around the threshold. If w is positive the functions have positive slope and negative slope if w is negative.

The cumulative density function (cdf) of the Normal distribution Φ, the *probit* response, can be parameterised as

$$F_{\text{gauss}}(x, \boldsymbol{\theta}) = \Phi\left(x \middle| m, \frac{w}{z(\alpha)}\right) \tag{7}$$

where $z(\alpha) = \Phi^{-1}(1-\alpha) - \Phi^{-1}(\alpha)$ using the quantile function Φ^{-1} (inverse of cdf) of the standard normal distribution. The resulting functions appear often very similar to the logistic (see Figure 2(b)).

The Gumbel function can be derived from the cdf of the Gumbel distribution and is known in GLMs as the *log-log* model. We use the parameterisation

$$F_{\text{gumbel}}(x, \boldsymbol{\theta}) = 1 - \exp\left(-\exp\left(\frac{z(\alpha) - z(1-\alpha)}{w}(x - m) + z(0.5)\right)\right) \tag{8a}$$

where $z(\alpha) = \ln(-\ln(\alpha))$. The Gumbel function is asymmetric. For small x the function is similar to the logistic function but approaches 1 faster as the stimulus intensity gets larger. The asymmetry can be reversed and we obtain the reversed Gumbel function

$$F_{\text{rgumbel}}(x, \boldsymbol{\theta}) = \exp\left(-\exp\left(\frac{z(1-\alpha) - z(\alpha)}{w}(x - m) + z(0.5)\right)\right) \tag{8b}$$

where again $z(\alpha) = \ln(-\ln(\alpha))$.

Another frequently found functional form is the cdf of the Weibull distribution which is also asymmetric. Using the Weibull function is equivalent to using a Gumbel function for log-transformed stimulus intensities. Unfortunately the Weibull function cannot be parameterised in terms of a width parameter w. Instead we parameterise the function by threshold location m and slope at threshold $s = \frac{\partial F}{\partial x}\big|_m$. So we use the following parametric form for the Weibull function

$$F_{\text{weibull}}(x, \boldsymbol{\theta}) = 1 - \exp\left(-\exp\left(\frac{2sm}{\ln(2)}\left(\ln(x) - \ln(m)\right) + \ln(\ln(2))\right)\right) \tag{9a}$$

and

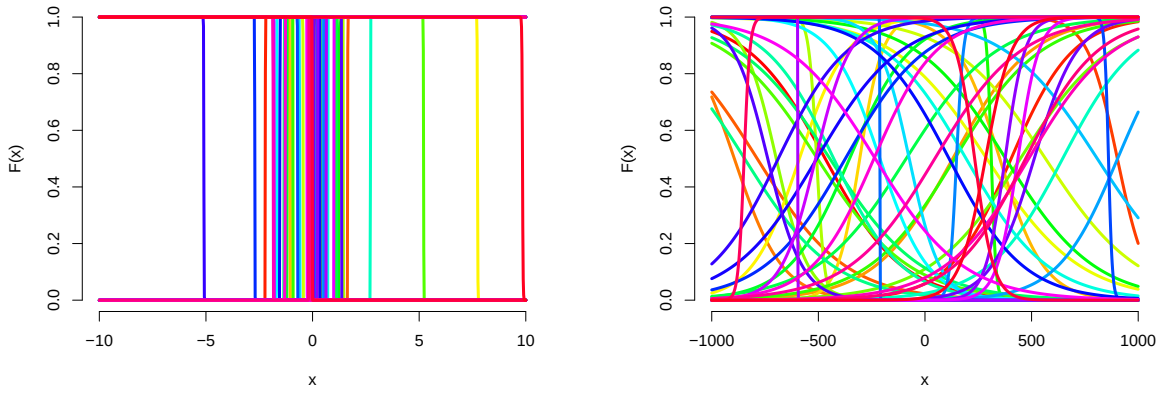
$$F_{\text{rweibull}}(x, \boldsymbol{\theta}) = \exp\left(-\exp\left(-\frac{2sm}{\ln(2)}\left(\ln(x) - \ln(m)\right) + \ln(\ln(2))\right)\right) \tag{9b}$$

for the reversed Weibull function. Both Weibull functions are defined for $x > 0$ and tend to 0 as $x \to 0$, which makes them conceptually appealing in many psychophysical settings.

4.2 Prior Distributions

Ideally a prior distribution describes the scientist's degree of belief for all hypotheses about the true model parameters. For continuous parameters one could ask to "draw" a curve over the parameter space representing the shape of the prior. The line would be at zero for parameter values which are believed to be absolutely impossible and otherwise proportional to the degree of belief in the hypothesis that the pen is over the true value. Using a prior from a parametric family of distributions can be seen as a convenient approximation to this "drawn prior" because it reduces the prior to a parametric form with a few parameters. In practice, a simple technique to find a parametric representation of prior beliefs is to plot probability densities functions from a convenient family of distributions. Varying the parameters one can often find a function that is close to the drawn prior. One should also sample from the prior and inspect whether the corresponding model is consistent with prior beliefs.

Often scientists unfamiliar with Bayesian data analysis feel that using informative priors—reflecting their understanding and uncertainties about the data generating process—somehow "distorts" the inference process. Expressing prior beliefs is certainly non-trivial, and requires great care. A common misconception is that using a flat, constant prior on model parameters is equivalent to expressing no prior information about the data generating process—from a Bayesian point of view this is exactly what non-Bayesians do when they do not specify any prior explicitly. In fact this prior describes the belief that *every* parameter value is equally likely to the scientist. However, this is typically *not* what the scientists intend: what they want is to allow every model or "shape of the model" to be equally likely, that is, they want a flat prior in "model space". Typically a flat prior on parameters is, unfortunately, not flat in model space. For psychometric functions this is illustrated in Figure 3. Here we show that using flat priors



(a) (b)

Figure 3: Simply changing the parameterisation of the psychometric function makes flat priors favour steep psychometric functions—the prior is flat on parameters but not in function space. We approximate a flat prior on the elements of $\boldsymbol{\theta}$ by uniform distributions on the interval $[-1000, 1000]$. For Figure (a) the logistic psychometric function was parameterised $F(x, \boldsymbol{\theta}) = (1 + \exp(-(\theta_1 x + \theta_2)))^{-1}$. We then sampled values of $\boldsymbol{\theta}$ from the "flat" prior and plotted the corresponding psychometric function. For Figure (b) the parameterisation as shown in equation 6 was used. Note the different x scales.

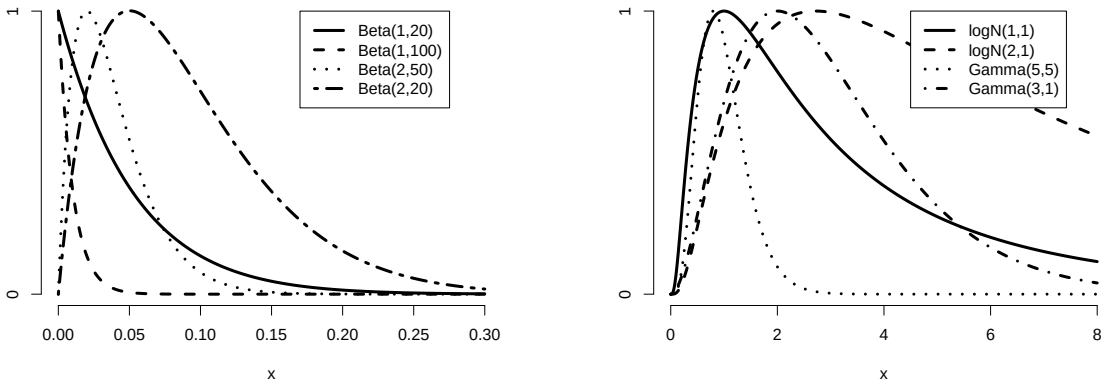
on the parameters $\boldsymbol{\theta}$ strongly favours very steep psychometric functions in Figure 3(a), whereas if we simply re-parameterise our psychometric function this tendency to favour steep psychometric functions disappears in Figure 3(b).

Using a flat prior for the lapse rate π_l, a uniform distribution on $[0, 1]$, indeed represents maximal uncertainty. The hypotheses that every experimental observation was independent of the stimulus or that no lapse occurred are equally likely. That might reflect the uncertainties of a scientist under certain circumstances, but in general the notion of a lapse implies a rare event. Note that a flat prior on the lapse probability allows the model to explain all the data as a sequence of lapses, which intuitively minimises the credibility of every observation. So if the scientist can safely assume that the lapse rate of an observer in a given task is small, the observations become more informative about the psychometric function and so its parameters can be better identified. On the other hand, excluding the potential existence of outliers forces the model to explain every single observation such that a single observation can become decisive. Note that also in the procedures described by Wichmann and Hill (2001a,b) the parameter similar to the lapse rate is constrained during the ML optimisation.

It can also be insightful to examine how sensitive the posterior reacts to changes of the prior. The more data is available and the more the data is informative about the parameters, the less influential the prior will be. Comparing posteriors and priors can illustrate how informative the experimental data is about the parameters. When the data does not reduce the uncertainty about a certain parameter then both distributions will be the same, expressing that the beliefs are unchanged. The only warning is not to put zero prior probability on potential parameter values unless one knows that they are impossible (Cromwell's dictum).

We now describe some families of distributions that will be used as priors for the parameters of psychometric functions in the experiments described in the following sections. For details on the distributions the reader is referred to any standard text book on statistics, e.g. DeGroot and Schervish (2002).

The lapse parameter π_l takes values in the unit interval and therefore the Beta distribution $p(\pi_l|\alpha, \beta) = \text{Beta}(\pi_l|\alpha, \beta)$ is a convenient choice (see Figure 4(a)). For $\alpha = 1$ and $\beta = 1$ the uniform distribution on $[0, 1]$ is a particular case.



(a) (b)

Figure 4: Illustration of the *form* of probability density functions of the beta, gamma, and log-normal distribution for different parameter values. Note that the probability density functions are scaled to the unit interval.

Considering priors for the elements of $\boldsymbol{\theta}$, parameterising the psychometric function as described above is advantageous because the parameters have a more intuitive interpretation. For convenience we specify the priors independently $p(\boldsymbol{\theta}) = p(\theta_1)\, p(\theta_2)$. Especially for the location parameter m a normal (Gaussian) distribution is a convenient choice if its value is unconstrained. By setting the standard deviation to increasingly large values the prior becomes more vague. For parameters that are known to be strictly positive, for example the width w or the slope s, the gamma or the log-normal distribution can be used. If x is log-normal distributed, $\log(x)$ follows a normal distribution. See Figure 4(b) for examples of gamma and log-normal probability density functions. This section sketched only a small selection of possible densities one can use for specifying priors on the parameters of psychometric functions. If common distributions are not sufficient to model the prior, also mixtures of distributions can be used.

5 Experiments

In this section we present and discuss simulations based on synthetic data and a case study in which we analyse real experimental data. Experiments in which the data is generated from the model can be useful for examining how well the true parameters can be identified dependent on the properties of the data. We do not aim at providing an exhaustive set of experiments for all possible data situations. Instead the focus will be on understanding the advantages and difficulties of the proposed method.

5.1 Synthetic Data

For illustration purposes, a data set from the binomial mixture model is generated, where F is a Gumbel function with threshold location $m = 5$, width $w = 3$ (for $\alpha = 0.1$), and lapse probability $\pi_l = 0.05$. For $k = 6$ stimulus intensities x_i corresponding to the F values equal to $[0.1, 0.3, 0.6, 0.74, 0.84, 0.94]$ we generate $N_i = 60$ samples respectively, which sums to 360 Bernoulli trials in total.

How to choose a prior in artificial experiments is a problematic issue. In the following examples the prior should be accepted as a toy-prior for demonstration purposes. For the lapse probability we use a Beta$(2, 50)$ prior (see Figure 4(a)). On the threshold location we put a wide normal prior with mean $\mu = 2$ and standard deviation $\sigma = 10$, which expresses very little information about m. On the width we put a log-normal prior distribution $\ln \mathcal{N}(1, 1)$ (see Figure 4(b)). Using hybrid MCMC sampling we simulate a Markov chain of 2000 samples from the posterior with 100 leapfrog steps and step sizes

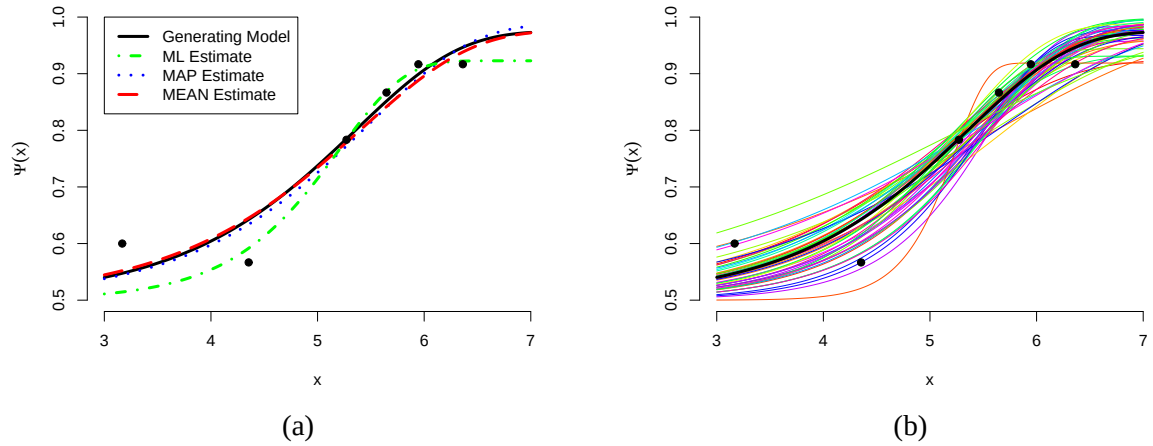


Figure 5: Synthetic data example: Figure (a) shows the generated data set (dots), the $\Psi(x, \boldsymbol{\theta}, \pi_l)$ that was used to generate the data and three estimates thereof. The ML clearly overestimates π_l and infers a too small width w. The estimate that appears closest to the generating $\Psi(x, \boldsymbol{\theta}, \pi_l)$ corresponds to the mean of MCMC samples and the MAP estimate. In Figure (b) we plot large number of "hypotheses", each corresponding to a MCMC sample.

$[0.5, 0.1, 0.2]$, which were chosen as to obtain an acceptance rate of approximately 80% and very little autocorrelation between samples.

Furthermore, we compute the posterior sample MEAN, MAP, and ML point estimates, for which the corresponding $\Psi(x, \boldsymbol{\theta}, \pi_l)$ are depicted in Figure 5(a). Taking samples from the MCMC chain and plotting the corresponding $\Psi(x, \boldsymbol{\theta}, \pi_l)$ we obtain Figure 5(b). Each of the sampled functions represents a hypothesis about the underlying generative function valid under the posterior. The functions are relatively close for large values of x but show rather large differences for smaller stimulus intensities. This can be interpreted such that the experimental observations for low stimulus intensities do still support a rather wide range of hypotheses about the width of the psychometric function.

We can gain more insights into the posterior by inspecting the MCMC samples. To illustrate how much the data reduced the uncertainty about the parameters we graphically compare priors and posteriors, of which the posterior is approximated by a normalised histogram of MCMC samples (see Figure 6).

For the lapse rate shown in Figure 6(a) we observe that the posterior is very similar to the prior, which indicates that the data did not allow us to reduce our uncertainty about this parameter. In many experiments we observed that identifying the lapse rate is relatively difficult. Nevertheless, the posterior samples of m and w are sampled while the assumed π_l value varies according to the prior.

For the threshold location m the samples shown in Figure 6(b) suggest that the threshold location is well inferred from the data. The prior, which was a wide normal, is approximately constant in the plotted region and we observe that the data was very informative. The posterior samples of the width w illustrated in Figure 6(c) show that the data was informative about w but the remaining uncertainty is still relatively large. Note that the function samples given in 5(b) already indicated that the posterior still supports a rather wide range of hypothesis on w.

We can use the empirical quantiles of the MCMC samples to estimate the quantiles of the posterior distribution. We take the range between the 0.05 and 0.95 empirical quantile as an approximation for the Bayesian 90% confidence interval.

In order to examine the accuracy of point estimates and the approximated Bayesian confidence regions we conducted a large set of repeated experiments. We compare approximate Bayesian confidence inter-

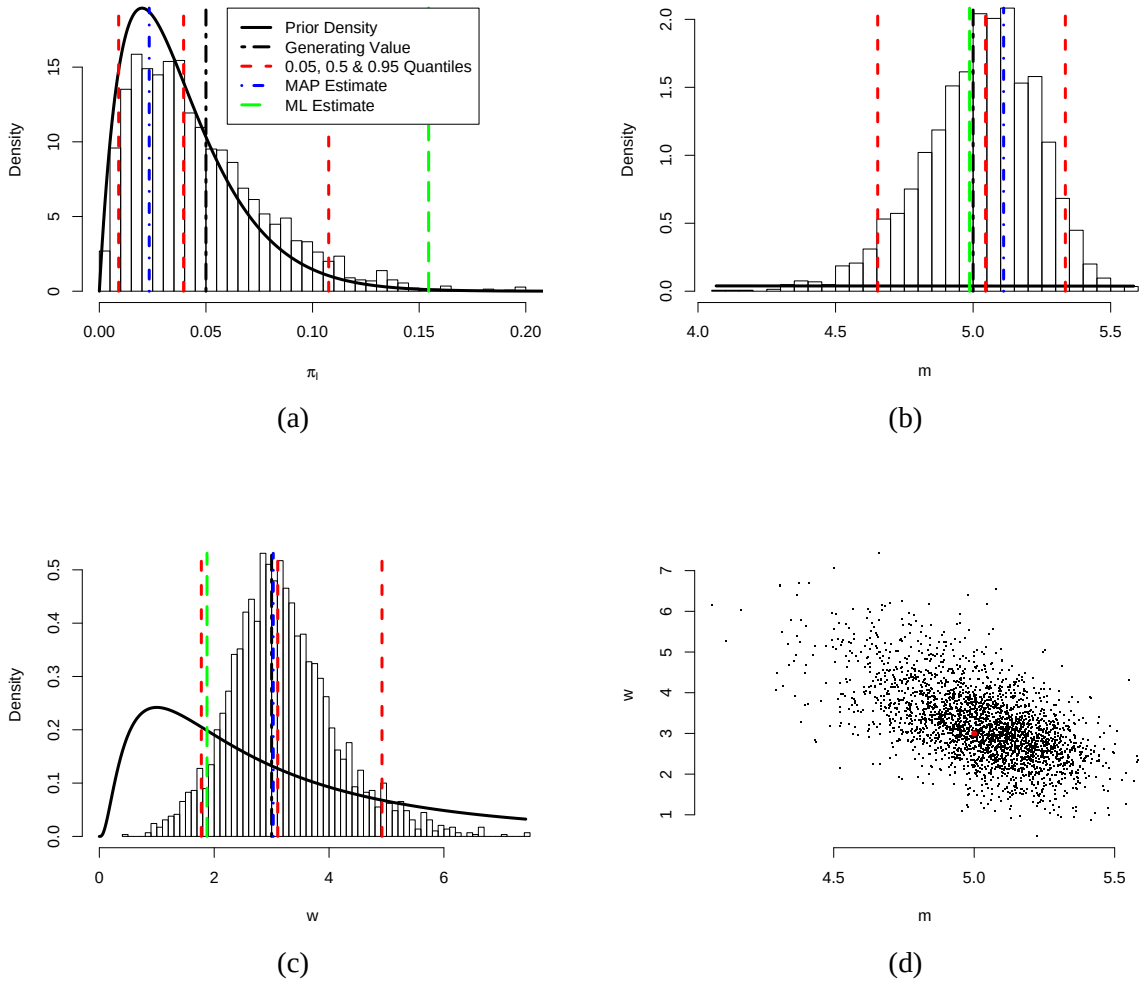


Figure 6: Synthetic data example: Plots of prior densities and histograms of posterior samples. Each plot (a–c) corresponds to one model parameter (π_l, m and w) and shows the normalised histogram of MCMC samples of the posterior distribution in comparison to the prior density. Vertical lines mark the value that was used to generate the data, the ML and MAP point estimates and the 0.05, 0.5, and 0.95 empirical quantiles of the MCMC samples. The interval between the 0.05 and 0.95 quantiles is the approximate Bayesian 90% confidence interval. Figure (d) shows a scatter plot of w and m parts of the samples. Note the negative correlation which corresponds to the necessity of steeper functions as the threshold location is moved to the right.

vals estimated from MCMC samples and the popular bootstrap[1] confidence intervals. In the experiments we varied the number of trials N and the lapse parameter π_l. The Gumbel function as described above with $m = 5$ and $w = 3$ and same sample locations were used. The data set size N takes the values $90, 360$, and 900 and lapse rate is set to either 0.05 or 0.15. For each of the six conditions we generated 1000 data sets.

Performing a large set of MCMC simulations is computationally demanding and we cannot inspect each individual chain. We used one set of parameters for hybrid MCMC sampling for each of the six conditions and later removed those rare runs in which the acceptance rate was lower than 50%. As above we used priors $p(m) = \mathcal{N}(2, 10)$ and $p(w) = \ln\mathcal{N}(1, 1)$. For data sets generated with a lapse rate $\pi_l = 0.05$ we used a Beta$(2, 50)$ prior for π_l in the MCMC sampling and a box constrained $[0, 0.1]$ on the corresponding parameter for bootstrap sampling. For data sets generated with $\pi_l = 0.15$ we used Beta$(2, 20)$ and $[0, 0.25]$ respectively.

At first we examine the accuracy of several point estimates for m and w. We compare the MAP estimate, the MCMC sample mean (MEAN) and median (MED), the maximum likelihood (ML) estimate, and the constrained ML (CML) estimate computed by `psignifit`. Each line in the following table states the median of the absolute errors of these point estimates in 1000 repeated experiments for the different values of N and π_l.

		$\lvert m - m^* \rvert$					$\lvert w - w^* \rvert$				
N	π_l	MAP	MEAN	MED	ML	CML	MAP	MEAN	MED	ML	CML
90	0.05	0.301	0.316	**0.289**	0.349	0.336	**0.906**	1.331	1.024	1.446	1.363
90	0.15	0.370	0.426	**0.353**	0.418	0.401	**0.905**	2.113	1.128	1.717	1.655
360	0.05	0.147	0.141	**0.136**	0.166	0.165	0.479	0.517	**0.478**	0.629	0.635
360	0.15	0.232	**0.175**	0.179	0.241	0.230	**0.559**	0.616	0.574	0.828	0.826
900	0.05	0.102	**0.088**	0.090	0.109	0.110	**0.314**	0.334	0.320	0.400	0.401
900	0.15	0.183	**0.131**	0.139	0.166	0.155	0.464	0.431	**0.405**	0.502	0.495

For estimating m the sample median MED consistently shows good accuracy. Note that the MED minimises the expected absolute error so this result conforms with the theory. The width w is best estimated by the MAP followed by the MED. The errors decrease with sample size and increase for the large lapse rate.

We now compare the reliability of bootstrapped and Bayesian confidence regions. We therefore compare the frequency at which the true generating value was included in the approximated 90% confidence interval. In theory this frequency should become exactly 90% for large numbers of repeated experiments. Larger values correspond to *over-conservative* statements while smaller values indicate *over-confidence*. With the frequency we also report the median width of the computed confidence intervals.

		Bayesian Confidence Intervals				Bootstrap Confidence Intervals			
		m		w		m		w	
N	π_l	Accuracy	Width	Accuracy	Width	Accuracy	Width	Accuracy	Width
90	0.05	0.911	1.765	0.933	8.103	0.783	1.422	0.903	9.049
90	0.15	0.922	2.340	0.981	11.882	0.707	1.558	0.911	18.258
360	0.05	0.918	0.750	0.931	2.959	0.863	0.757	0.883	3.184
360	0.15	0.926	0.884	0.937	3.745	0.795	0.915	0.865	4.148
900	0.05	0.919	0.457	0.916	1.694	0.848	0.488	0.859	1.934
900	0.15	0.901	0.585	0.916	2.209	0.818	0.682	0.867	2.573

For the threshold location m the approximated Bayesian confidence intervals exhibit accuracy close to the desired 90% for all six conditions. The bootstrap confidence intervals appear to be over-confident

[1]For the bootstrap experiments we use the `psignifit` software implementation of methods described by Wichmann and Hill (2001b) which can be obtained from `http://www.bootstrap-software.org`.

especially for small data sets and high lapse rates. For small data sets $N = 90$ the width of the Bayesian confidence regions is larger while for larger data set sizes the Bayesian confidence intervals exhibit higher accuracy but smaller interval width.

For the w parameter both the bootstrap and the Bayesian confidence regions show to be relatively accurate. The Bayesian confidence regions tend to be conservative, especially for $N = 90$ and $\pi_l = 0.15$, while the median width over the confidence regions is consistently smaller.

In the presented set of synthetic experiments the Bayesian MCMC sampling based estimators show to give more accurate point estimates and more accurate and tighter confidence regions.

5.2 A Case Study

The data for the case study are taken from a visual contrast discrimination task published by Henning et al. (2002). Observers either performed a sinusoidal contrast increment detection, or detected a contrast increment applied to a pulse train grating; both were two interval forced choice tasks. For both conditions the contrast of the added signal was varied using the method of constant stimuli.

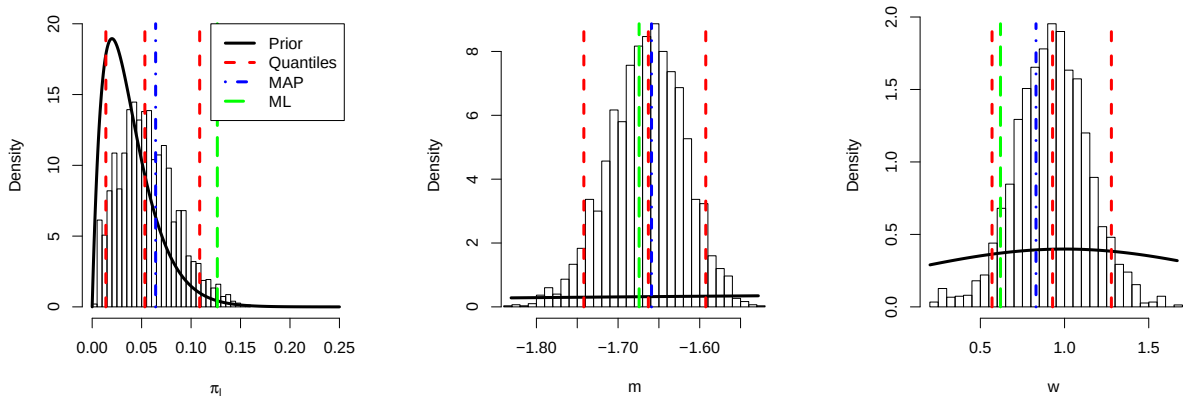
The aim of the experiment was to determine whether the two conditions yielded similar or different discrimination thresholds. Both stimuli—the sine wave and the pulse train—have the same fundamental frequency but the pulse train has additional higher frequency components. Hence, one might expect that these facilitate discrimination and therefore the threshold for the pulse condition might be lower.

First we analyse the data from the sine wave condition. The data come from one of the observers and consists of 13 blocks with 50 trials each. Each block was measured at a different contrast between 0.5% and 7.5%. Using a Weibull function to explain the data is a common choice for contrast experiments. Instead of directly fitting a Weibull we have found it more convenient to log-transform the contrast and use a Gumbel function instead. As pointed out before these two possibilities are equivalent but the Gumbel function allows a more intuitive parameterisation in terms of the width.

Next, we have to specify our prior beliefs about the parameter values. For the lapse rate, a convenient choice for the prior is the beta distribution. We expect from our experience with observers that some of the trials are lapses. A reasonable choice that makes small lapse rates more likely than big lapse rates is $\alpha = 2$ and $\beta = 50$ (see Figure 4(a)). This prior also expresses our belief that observers are unlikely to perform errorlessly. The mean of the prior distribution is given by $\alpha/(\alpha + \beta) \approx 4\%$ and the mode is $(\alpha - 1)/(\alpha + \beta - 2) = 2\%$. Thinking about the stimulus, we can derive a conservative prior on the threshold location. At 100% contrast a sine wave can clearly be seen but at about 10% the task becomes difficult. This is the range that we would expect the threshold to be in. At 1% the task seems almost impossible. A reasonable prior on log contrast therefore has a maximum at -1 (10%). We can take a Gaussian with this mean. Even though -2 corresponding to (1%) and 0 corresponding to (100%) seem to be unlikely threshold values we do not want to rule out these hypotheses *a priori*. Therefore, a standard deviation of 1 seems to be a conservative choice.

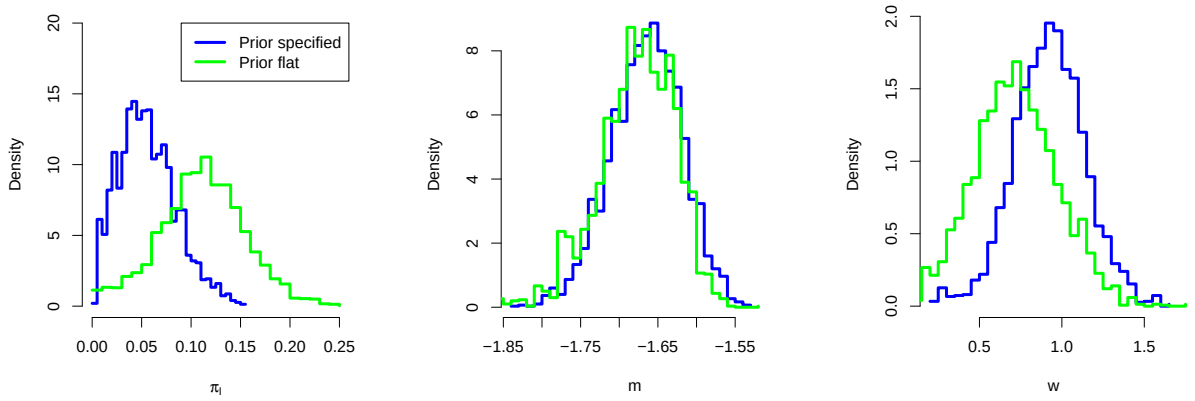
A width smaller than zero would correspond to a psychometric function for which performance increases with lower contrasts, so we constrained the prior to positive values. A width of 2 log-units is highly unlikely because it would mean that the psychometric function potentially ranges from 1% to 100% contrast. Therefore, a width between 0.5 and 1.5 log-units seems to be a reasonable range. For positive parameters the gamma distribution is a common choice for the prior. By plotting a gamma distribution for $\alpha = 2$ and $\beta = 1.5$ we found it to be a good description of our beliefs. The mean is given by $\alpha/\beta = 1.33$ and the mode by $(\alpha - 1)/\beta = 0.66$. The standard deviation that is given by $\sqrt{\alpha}/\beta = 0.94$ is large enough to even support values bigger than 2.

Once we have specified the prior we can sample from the posterior. For this we have to set values for the number and sizes of the leap-frog steps. We emphasise that this stage does need some experience and understanding of the MCMC procedure to ensure that enough approximately independent samples are generated in reasonable time. The code for reproducing this case study can be found in Appendix A.4



(a) (b) (c)

Figure 7: Sine condition: The estimated posterior distributions for the lapse parameter (a), the threshold (b) and the width (c) of the psychometric function. Vertical lines depict MAP estimates, ML estimates and quantiles at 5%, 50% and 95%. The solid black line shows the prior distribution. For the threshold and width the prior is relatively flat compared to the posterior.

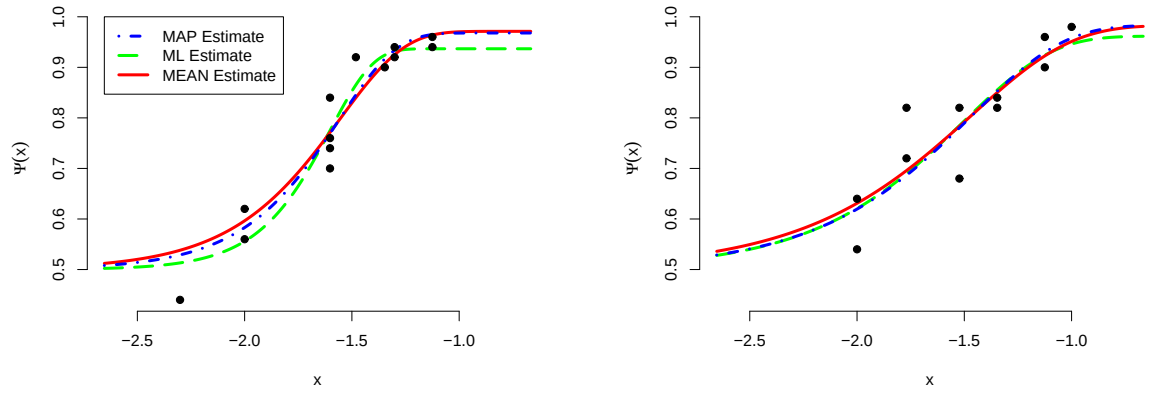


(a) (b) (c)

Figure 8: Sine condition: The estimated posterior distributions for the lapse parameter (a), the threshold (b) and the width (c) of the psychometric function. The posterior was computed with the same prior as in Figure 7 and for a flat prior for comparison. For the threshold parameter there is hardly any difference between the two posteriors. For the lapse parameter the influence of the prior is substantial.

and Appendix A.3 contains further comments about how the parameters of the MCMC sampling can be set.

Figure 7 presents histograms of the posterior samples generated by hybrid MCMC sampling. First we inspect the samples corresponding to the threshold parameter. The MAP, ML and MEAN point estimate for the threshold all lie between -1.70 and -1.65 log-units, i.e. at a contrast between 2.0% and 2.2%. Furthermore we compute the approximate Bayesian 90% confidence region from empirical quantiles of the samples. The Bayesian confidence interval ranges from -1.74 to -1.59 log-units or 1.8% and 2.6% contrast (the outermost dashed lines in Figure 7). Information like this is necessary if one wants to compare thresholds from different conditions.



(a) (b)

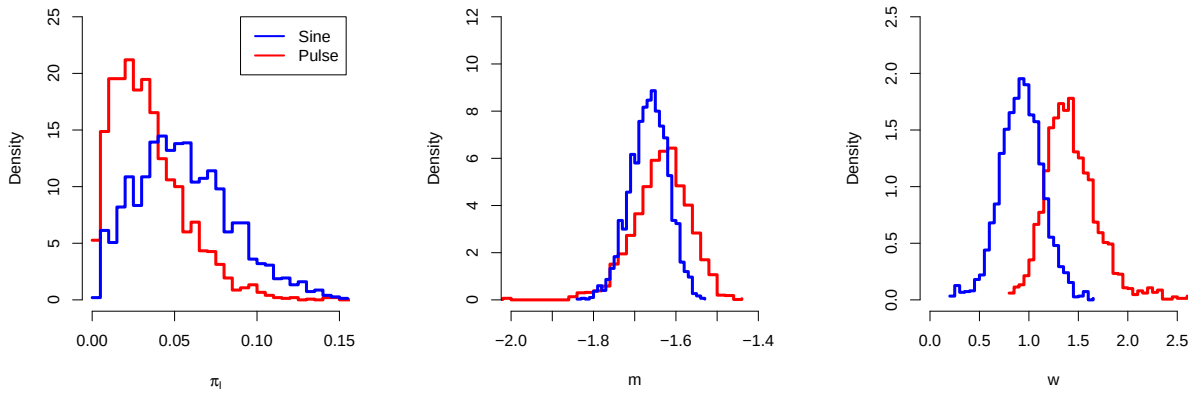
Figure 9: Point estimates for both conditions: Figure (a) shows data from the sine condition and three estimates of the psychometric function (MAP, ML and MEAN). Each data point represents 50 trials. In Figure (b) we show the same for the pulse condition. For both conditions the three estimates are very similar.

At this point it may be interesting to examine the influence of the prior in the analysis. At first sight, it can be noted that the prior densities for the threshold and the width parameter were flat relative to the posterior distributions. The posterior of the lapse rate parameter is similar to the prior which shows that the data did not allow a reduction in uncertainty about this parameter. Figure 7 also depicts the maximum likelihood (ML) and the maximum a posteriori (MAP) estimates. The difference between the two indicates how much the prior has influenced the MAP estimate. The difference for the lapse rate is substantial which again emphasises the importance of the prior on this parameters. The ML estimate suggests that far more than 10% of the observations were lapses, which also explains why the width is estimated to be relatively small.

Furthermore, we sample from a posterior distribution where the prior has been chosen to be flat. This allows us to examine how sensitive the posterior is to the choice of the prior. Note again that a flat prior is not uninformative. Figure 8 compares the posterior distributions that result from using either flat priors (on log-contrast) or the priors as specified above. It reveals that the choice of prior matters for the lapse parameter and the width but in this case not so much for the threshold.

Figure 9(a) shows the data and three point estimates of the psychometric function for the sine condition. We have also plotted approximated Bayesian 95% confidence intervals for the threshold parameter as given by the posterior (see Figure 7 (b)).

The second condition in the experiment was a pulse train instead of a sine wave discrimination task. The fundamental frequency of the pulse train was identical to the frequency of the sine wave. As the pulse train has additional higher frequency components one may expect that these facilitate discrimination. For the second condition the data consists of 11 blocks with 50 trials each. The stimuli varied between 10% contrast and 1% contrast. We used the same priors as we used for the first condition. The data can be seen in Figure 9(b) along with various estimates. The psychometric functions for the pulse and the sine condition look similar. Figure 10 compares the posteriors of both conditions, confirming that the approximated posterior distributions over the parameters are highly overlapping. Especially, the parameters for the thresholds are very close. Comparing the psychometric functions one might suspect a difference in the their width. The posteriors over the width parameters are shown in the right panel of Figure 10. It has to be stressed that the lapse parameter and the width cannot be interpreted independent of each other, since a negative correlation between the two parameters can be seen in the MCMC samples.



(a) (b) (c)

Figure 10: A comparison of the sine and the pulse condition. Panel (a) shows the posterior distributions for the lapse parameter. In panel (b) the posteriors for the thresholds are similar. The uncertainty is a bit bigger in the pulse condition. (c) It cannot be concluded that the psychometric functions have a different width. The respective posteriors are not different enough. Usually, the lapse parameter and the width show a negative correlation. Hence, they need to be compared carefully. The smaller width of the sine condition goes along with a higher lapse rate.

Intuitively, a higher lapse rate squeezes Ψ down which is compensated by a smaller width. In conclusion, we did not find evidence that pulse trains lead to a different discrimination performance than sine waves.

6 Conclusions

In this article we presented a Bayesian approach to inference about the parameters of psychometric functions. Since computing the density of the posterior distribution is analytically intractable we described how Markov chain Monte Carlo techniques can be used instead to generate samples from the posterior.

We exemplified that the proposed Bayesian method can produce more exact point estimates and confidence intervals than the popular frequentist bootstrap technique. Although we cannot prove that this observation generalises to all possible data situations, there is no reason to believe that the Bayesian approach should do worse on other datasets. Furthermore, the Bayesian approach exhibits several conceptual advantages. Yet another advantage is that by inspecting the MCMC samples and observing correlations and dependencies we gain a deeper understanding of the process at hand. A difficulty of the proposed method is that using Markov chain Monte Carlo methods is non-trivial and requires the Markov chains to be inspected by the user.

We discussed the role of prior distributions in the analysis of experimental data and the difficulties of avoiding informative priors. We observed that especially the prior on the lapse parameter can be influential. For $\boldsymbol{\theta}$ we found that even relatively small data sets are often informative enough to overrule the prior. However, a Bayesian analysis should always report prior and posterior distributions and the latter should always be interpreted relative to the prior given the model.

The described binomial mixture model for parametric psychometric functions is easy to analyse and efficient to implement but is based on perhaps overly simplistic assumptions. Assuming a particular parametric form of F might be difficult and opens the problem of model selection. The assumption that for a given stimulus intensity the Bernoulli trials all have the same probability of success ignores adaptation processes, learning, and other forms of non-stationarity.

Future work will be devoted to study Bayesian inference for models making less restrictive assumptions. Possible directions include *over-dispersion* models using the beta-binomial model developed in

Generalised Linear Models as for example described by Williams (1982) and Prentice (1986). Another simple extension is to use *neural networks* to implement the psychometric function $F(x, \boldsymbol{\theta})$. The potential advantage is that the class of functions implemented by neural networks is much richer than the functions described in Section 4 such that the risk of model-mismatch can be reduced.

As a companion to this report we release a software implementation in form of a package named `PsychoFun` for the (free) R environment for statistical computing[2].

Acknowledgements

We would like to thank Jeremy Hill and Carl Edward Rasmussen for helpful comments and discussions. MK was supported by the German Research Council (DFG) through grant RA 1030/1.

References

R. Alcalá-Quintana and M. A. Garcia-Pérez. The role of parametric assumptions in adaptive Bayesian estimation. *Psychological Methods*, 9(2):250–271, 2004.

D. Collet. *Modelling Binary Data*. Chapman & Hall, Boca Raton, 1991.

M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison-Wesley, 2002, third edition, 2002.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

L. Fahrmeier and G. Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer, New York, second edition, 2001.

D. J. Finney. *Probit Analysis*. Cambridge University Press, Cambridge, 3rd edition, 1971.

D. H. Foster and W. F. Bischof. Thresholds from psychometric functions: superiority of bootstrap to incremental and probit variance estimators. *Psychological Bulletin*, 109:152–159, 1991.

D. H. Foster and W. F. Bischof. Bootstrap estimates of the statistical accuracy of thresholds obtained from psychometric functions. *Spatial Vision*, 11(1):135–139, 1997.

M. A. Garcia-Perez. Forced-choice staircases with fixed step sizes: Asymptotic and small-sample properties. *Vision Research*, 38(12):1861–1881, 1998.

A. Gelman. Inference and monitoring convergence. In Gilks and Richardson (1996), pages 131–144.

W. R. Gilks and S. Richardson, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, 1996.

D. M. Green and J. A. Swets. *Signal Detection Theory and Psychophysics*. John Wiley & Sons, Inc, New York, 1966.

G. B. Henning, C. M. Bird, and F. A. Wichmann. Contrast discrimination with pulse trains in pink noise. *Journal of the Optical Society of America A*, 19(7):1259–1266, 2002.

E. T. Jaynes. *Probability Theory*. Cambridge University Press, Cambridge, 2003.

C. Kaernbach. Slope bias of psychometric functions derived from adaptive data. *Perception & Psychophysics*, 63(8):1389–1398, 2001.

[2]The software and further information can be obtained from `http://www.R-project.org`

P. E. King-Smith and D. Rose. Principles of an adaptive method for measuring the slope of the psychometric function. *Vision Research*, 37(12):1595–1604, 1997.

S. A. Klein. Measuring, estimating, and understanding the psychometric function: A commentary. *Perception & Psychophysics*, 2001.

L. L. Kontsevich and C. W. Tyler. Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, 39(16):2729–2737, 1999.

D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 175–204. MIT Press, Cambridge, MA, 1999.

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

R. Madigan and D. R. Williams. Maximum-likelihood psychometric procedures in two-alternative forced-choice: Evaluation and recommendations. *Perception & Psychophysics*, 42(3):240–249, 1987.

L. T. Maloney. Confidence intervals for the parameters of psychometric functions. *Perception & Psychophysics*, 47:127–134, 1990.

S. P. McKee, S. A. Klein, and D. Y. Teller. Statistical properties of forced-choice psychometric functions: Implications of probit analysis. *Perception & Psychophysics*, 37(4):286–298, 1985.

J. Miller and R. Ulrich. On the analysis of psychometric functions: The Spearman-Karber method. *Perception & Psychophysics*, 63(8):1399–1420, 2001.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

A. O'Hagan. *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics*. Arnold, London, 1994.

J. K. O'Regan and R. Humbert. Estimating psychometric functions in forced-choice situations: Significant biases found in threshold and slope estimation when small samples are used. *Perception & Psychophysics*, 45:434–442, 1989.

D. G. Pelli. The ideal psychometric procedure. *Investigative Ophthalmology and Visual Science (Supplement)*, 28:366, 1987.

A. Pentland. Maximum likelihood estimation: the best PEST. *Perception & Psychophysics*, 28(4): 377–379, 1980.

R. L. Prentice. Binary regression using an extended beta-binomial distribution, with discussion of correlation induced by covariate measurement errors. *Journal of the American Statistical Association*, 81 (394):321–327, 1986.

R. M. Rose, D. Y. Teller, and P. Rendleman. Statistical properties of staircase estimates. *Perception & Psychophysics*, 8(4):199–204, 1970.

P. R. Snoeren and M. J. H. Puts. Multiple parameter estimation in an adaptive psychometric method: MUEST, an extension of the QUEST method. *Journal of Mathematical Psychology*, 41(4):431–439, 1997.

M. M. Taylor. On the efficiency of psychophysical measurement. *Journal of the Acoustical Society of America*, 49(2, pt. 2):505–508, 1971.

B. Treutwein. Adaptive psychophysical procedures. *Vision Research*, 35(17):2503–2522, 1995.

B. Treutwein and H. Strasburger. Fitting the psychometric function. *Perception & Psychophysics*, 61(1): 87–106, 1999.

A. B. Watson and D. G. Pelli. QUEST: a Bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113–120, 1983.

R. J. Watt and D. P. Andrews. APE: adaptive probit estimation of psychometric functions. *Current Psychological Reviews*, 1(2):205–213, 1981.

G. B. Wetherill and H. Levitt. Sequential estimation of points on a psychometric function. *British Journal of Mathematical and Statistical Psychology*, 18(1):1–10, 1965.

F. A. Wichmann. *Some Aspects of Modelling Human Spatial Vision: Contrast Discrimination*. PhD thesis, Oxford University, 1999.

F. A. Wichmann and N. J. Hill. The psychometric function: I. fitting, sampling and goodness-of-fit. *Perception & Psychophysics*, 63(8):1293–1313, 2001a.

F. A. Wichmann and N. J. Hill. The psychometric function: II. bootstrap-based confidence intervals and sampling. *Perception & Psychophysics*, 63(8):1314–1329, 2001b.

D. A. Williams. Extra-binomial variation in logistic linear models. *Applied Statistics*, 31(2):144–148, 1982.

A The `PsychoFun` Software Package

In this section we describe some implementational details that might be useful for using the routines. This report covers `PsychoFun` version 0.5-0 which has been written entirely in R. The computational performance could be drastically improved by implementing the basic routines in C, which might be an issue for future releases. This section also provides some code that was used in the examples described in the main text.

A.1 Implementational Details

As described above, we are interested in sampling from the posterior

$$p(\boldsymbol{\theta}, \pi_l|\mathcal{D}, \pi_c) \propto p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c)\, p(\boldsymbol{\theta})\, p(\pi_l) \tag{10}$$

which we cannot compute in closed form. For hybrid MCMC sampling and for computing MAP estimates it is sufficient to be able to compute the logarithmic value of the product of likelihood and prior

$$\ln p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c) + \ln p(\boldsymbol{\theta}) + \ln p(\pi_l) \tag{11}$$

and the partial derivatives thereof with respect to π_l and $\boldsymbol{\theta}$. We use the gradient based optimisation routine "BFGS" provided by R to maximise this function to obtain MAP estimates. Note that numerical optimisation methods in general converge to a local mode of the posterior which must not necessarily be the global maximum. It is therefore recommendable to repeat the optimisation from different initialisations. The implementation of hybrid MCMC sampling is very similar to Algorithm 1.

The lapse parameter π_l is by definition constrained to the unit interval. In order to use hybrid MCMC and unconstrained optimisation routines the model can be reparameterised in terms of a real valued unconstrained parameter a such that

$$a(\pi_l) = -\ln\left(\frac{1}{\pi_l} - 1\right) \text{ and } \pi_l(a) = \frac{1}{1+\exp(-a)} \tag{12}$$

for which we have to make a Jacobian correction (for MCMC only)

$$p(\boldsymbol{\theta}, a|\mathcal{D}, \pi_c) = p(\boldsymbol{\theta}, \pi_l|\mathcal{D}, \pi_c)\frac{\partial \pi_l}{\partial a} \tag{13}$$

where $\frac{\partial \pi_l}{\partial a}$ is the so called Jacobian of the transformation (see for example DeGroot and Schervish (2002, ch 3.8)). Let

$$E(\boldsymbol{\theta}, \pi_l) = -\ln p(\mathcal{D}|\boldsymbol{\theta}, \pi_l(a), \pi_c) - \ln p(\boldsymbol{\theta}) - \ln p(\pi_l(a)) - \ln \frac{\partial \pi_l}{\partial a} \tag{14}$$

denote the negative value of the logarithm of the product of likelihood, priors and Jacobian. This function (and its derivatives) are used for hybrid MCMC sampling.

Let $\mathcal{D} = \{(x, N, n)_i | i = 1, \ldots k\}$ be the experimental data, then the log-likelihood for the Binomial model described in Section 2 can be computed as

$$\begin{aligned} \ln p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c) &= \ln \prod_{i=1}^{k} \binom{N_i}{n_i} \Psi(x_i, \boldsymbol{\theta}, \pi_l, \pi_c)^{n_i} (1 - \Psi(x_i, \boldsymbol{\theta}, \pi_l, \pi_c))^{N_i - n_i} && \text{(15a)} \\ &= \sum_{i=1}^{k} \left[\ln \binom{N_i}{n_i} + n_i \ln \Psi + (N_i - n_i) \ln(1 - \Psi) \right] . && \text{(15b)} \end{aligned}$$

The software implementation allows various forms of prior distributions. For π_l usually a Beta prior will be used and the flat prior (a uniform distribution on $[0, 1]$) corresponds to a Beta$(1, 1)$ prior. For the

elements of $\boldsymbol{\theta}$ the user can choose between normal, log-normal, gamma and flat (constant) priors. Since the software calls the R routines to evaluate the respective probability densities the parameterisation and implementation of these routines can be obtained from the respective help pages.

Furthermore, the package provides some useful macros which should cover the standard use-cases.

A.2 Hybrid Monte Carlo Sampling

Let $\boldsymbol{\phi} = [\boldsymbol{\theta}, \pi_l]$ be the shorthand notation for the vector of parameters of the model. In MCMC methods a Markov chain in the parameter space $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots$ is generated such that the distribution of the state is asymptotically identical to the posterior distribution. The state explores the parameter space in a random walk. The random walk is constructed according to the Metropolis-Hastings method such that in the long run the probability that the state is at a certain position is identical to the posterior distribution. The movement of the state is simulated and its states are interpreted as samples from the posterior. The challenge is to construct a Markov chain properly such that it explores the whole posterior distribution efficiently, in order to obtain a number of approximately independent samples in reasonable time. Different techniques exist to construct such a Markov chain. We use *hybrid* Monte Carlo sampling which is also known as *Hamiltonian* sampling, as described by Neal (1993) and MacKay (2003, ch. 30).

Hybrid MCMC sampling is a computationally efficient method that can be used in situations in which the parameters are continuous and the derivative of the likelihood and the priors can be computed. New states are proposed using a procedure that can be understood as a discrete simulation of Hamiltonian dynamics, hence the name Hamiltonian sampling. The intuition behind Hamiltonian sampling is that the states of a chain are interpreted as the positions of a particle that moves through the parameter space. In the Metropolis-Hastings algorithm the particle moves through the parameter space by a purely random walk. The next position of the particle is found by taking a random step from the old position. In Hamiltonian sampling the particle also takes random steps but additionally it has a momentum and follows the gradient. Because of the momentum the particle can travel a greater distance in the parameter space than would be possible by a purely random walk. The parameter space is explored faster and nearly independent samples are obtained quicker.

The algorithm consists of three steps. The first step is a random perturbation of the particle. The second step can be interpreted as a discrete simulation of Hamiltonian dynamics. In Hamiltonian dynamics the potential energy of a particle is transformed into kinetic energy following the gradient, and vice versa. By setting the potential energy equal to the negative logarithmic value of the unnormalised posterior it is ensured that asymptotically the states of the particle can be taken as samples from the posterior. Low probability regions have a high potential energy and therefore a particle will spend more time in the regions with high probability mass. In our case the potential function is

$$\begin{aligned} E(\boldsymbol{\phi}) &= -\ln\left[p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c)p(\boldsymbol{\theta})p(\pi_l)\right] && \text{(16a)} \\ &= -\ln p(\mathcal{D}|\boldsymbol{\theta}, \pi_l, \pi_c) - \ln p(\boldsymbol{\theta}) - \ln p(\pi_l)\ . && \text{(16b)} \end{aligned}$$

In the third step, after simulating the Hamiltonian dynamics for a certain time, the resulting position of the particle is a proposal state that is either accepted or rejected according to the Metropolis-Hastings rule.

The discrete simulation of the dynamics is implemented using the so-called leapfrog method. Algorithm 1 provides a schematic overview of the sampling scheme. The accuracy of the physical simulation is given by a parameter that is called leapfrog step size. It is better to have one step size for each parameter separately because the parameters will have different scales. The accuracy of the physical simulation is not crucial. By using the Metropolis-Hastings rule for the acceptance of proposal states the validity of the chain is assured irrespective of the accuracy of the physical simulation. Big leaps are possible and are actually desirable because they are computationally more efficient. Another parameter of the algorithm

Algorithmus 1 Hybrid MCMC Sampling

Given: Initial state $\boldsymbol{\phi}_0$, length of simulation T, number of leapfrog steps, vector of leapfrog step sizes $\boldsymbol{\epsilon}$, energy function E
Initialise $e \leftarrow E(\boldsymbol{\phi}_0)$ and $\mathbf{g} \leftarrow \nabla E(\boldsymbol{\phi}_0)$
for $t = 0, \ldots, T$ **do**
  Sample initial momentum $\mathbf{m}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
  $H \leftarrow \frac{\mathbf{m}^\top \mathbf{m}}{2} + e$
  Set $\tilde{\boldsymbol{\phi}} \leftarrow \boldsymbol{\phi}_t$ and $\tilde{\mathbf{g}} \leftarrow \mathbf{g}$
  for all leapfrog steps **do**
    $\mathbf{m} \leftarrow \mathbf{m} - \frac{1}{2}\boldsymbol{\epsilon} \odot \tilde{\mathbf{g}}$ {where $\odot$ denotes the element-wise product}
    $\tilde{\boldsymbol{\phi}} \leftarrow \tilde{\boldsymbol{\phi}} + \boldsymbol{\epsilon} \odot \mathbf{m}$
    $\tilde{\mathbf{g}} \leftarrow \nabla E(\tilde{\boldsymbol{\phi}})$
    $\mathbf{m} \leftarrow \mathbf{m} - \frac{1}{2}\boldsymbol{\epsilon} \odot \tilde{\mathbf{g}}$
  end for
  $\tilde{e} \leftarrow E(\tilde{\boldsymbol{\phi}})$
  $\tilde{H} \leftarrow \frac{\mathbf{m}^\top \mathbf{m}}{2} + \tilde{e}$
  Draw u from a uniform distribution on $[0, 1]$
  if $\ln u > \tilde{H} - H$ **then** {Proposed state is accepted}
    $\boldsymbol{\phi}_{t+1} \leftarrow \tilde{\boldsymbol{\phi}}$
    $\mathbf{g} \leftarrow \tilde{\mathbf{g}}$
    $e \leftarrow E(\boldsymbol{\phi}_{t+1})$
  else {Proposed state is rejected}
    $\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t$
  end if
  Store $\boldsymbol{\phi}_{t+1}$ and $E(\boldsymbol{\phi}_{t+1})$
end for
Return: $\boldsymbol{\phi}_t$ and $E(\boldsymbol{\phi}_t)$ for $t = 1, \ldots, T$

is the number of leapfrog steps between the random perturbations. The number of leapfrog steps and the leapfrog step size determine how far the particle can travel before being proposed as the consecutive state.

The algorithm requires that for given parameter values ϕ the logarithmic value of the likelihood and the prior can be evaluated. Furthermore derivatives of these quantities with respect to the parameters ϕ must be computed. Note that the lapse parameter π_l is constrained to the $[0, 1]$ interval, but hybrid MCMC is suitable only for unconstrained continuous variables. As described in Appendix A.1 using an invertible mapping from the real line to the $[0, 1]$ interval this problem can be avoided.

In the following section we discuss some heuristics how to set the leapfrog step size and the number of leap frogs in practice. After simulating the Markov chain using Algorithm 1 the chains have to be inspected for convergence and mixing. The chains should travel the support of the posterior quickly and states close in the chain should be as uncorrelated as possible. More on analysing convergence can be found in Gelman (1996).

A.3 Some Hints Concerning the Parameterisation of hybrid MCMC

Using Markov chain Monte Carlo methods in practice requires some experience and manual fine-tuning. For a comprehensive overview of practical issues of MCMC sampling the reader is referred to Gilks and Richardson (1996).

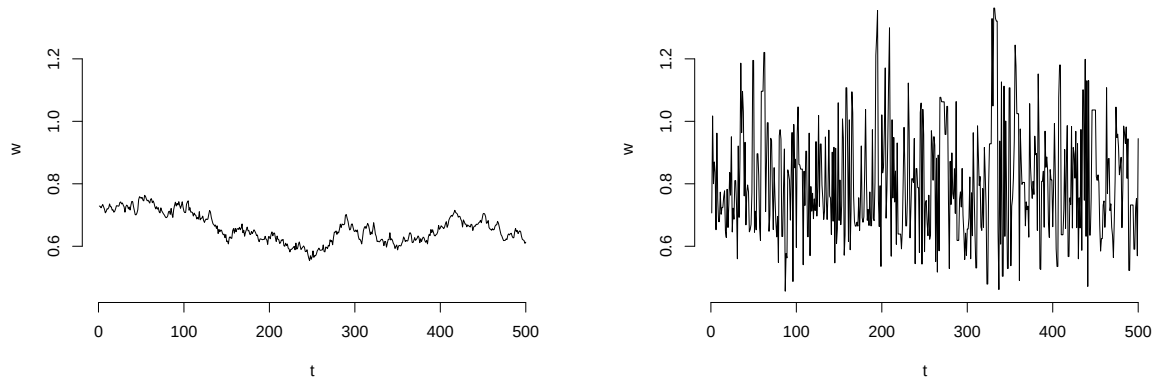
In theory, hybrid MCMC sampling can be shown to converge to the correct answer but the speed at which this happens and therefore the computational efficiency can be controlled by a few parameters. The parameters to tune are the number of leapfrog steps and the respective step sizes. The aim is to obtain a sufficient number of samples in reasonable time, such that the statistics of the samples are a good approximation to the corresponding properties of the posterior distribution.

Certain characteristics of the simulated chains should be inspected for each simulation:

- Plots of the samples, e.g. using the routine `PlotMCMCSamples` provided by `PsychoFun`. The chains should not show any obvious trends, shifts, steps etc.
- The autocorrelation of samples, e.g. using the function `acf` provided by R. The less correlated the samples are the more efficient the sampling is. Sometimes autocorrelation between samples can be reduced by increasing the corresponding leapfrog step sizes.
- The acceptance rate, as returned by `MCMCSampling`, is the fraction of proposed states that were accepted. As a rule of thumb: the acceptance rate should be between 60% and 90% and can be influenced by the number and size of leapfrog steps.

We now describe a heuristic procedure how to proceed in order to find good parameters for hybrid MCMC as implemented in `PsychoFun`. For a fixed number of leapfrog steps, e.g. 50, we set the step sizes to very small values, such that all proposed states in a simulation are accepted. We then vary one step-size parameter at a time to localise the region in which the acceptance rate reacts sensitively. Afterwards we increase—a trial-and-error procedure—all step size parameters such that the acceptance rate balances between 60% and 90% and the autocorrelation is acceptable (decreases rapidly). For each simulation the chains are plotted and autocorrelation coefficients are computed. Note that having rejections indicates that the chain also reaches areas of lower posterior density. If the samples show large autocorrelation we can a) increase the number of leapfrog steps and b) down-sample the chains. Increasing the number of leapfrog steps decreases the dependency between samples but is computationally expensive. By down-sampling we refer to the procedure of simulating a longer chain and to pick only every nth state of the chain as a sample.

We now give a concrete example of how this procedure might work in practice. At first we load some toy-data that comes with `PsychoFun` and describe the priors:



(a) (b)

Figure 11: Plots illustrating bad and good behaviour of simulated MCMC chains. Both figures show the part of a MCMC chain corresponding to the width parameter for two differently parameterised runs of hybrid MCMC. In Figure (a) a very small step size and a small number of leapfrog steps were used (acceptance rate was 1). The effect is that the states in the chain are highly correlated and the chain does not explore the posterior high density region. In Figure (b) the step size and the number of leapfrog steps were increased as described in the example resulting in an acceptance rate of 87%. The samples are much less correlated and the chain travels the whole range of the high posterior density region.

```
library(PsychoFun)              # Load routines
data(PsychoFunToyData)          # Load some toy data

SetUp <- PsychoSetUp(PsychoFunToyData,
                     plPrior="beta",
                     plPriorParameters=c(2,50),
                     t1Prior="normal",
                     t1PriorParameters=c(2,10),
                     t2Prior="lognormal",
                     t2PriorParameters=c(2,1),
                     type = "gumbel")

PlotPsychoPriors(SetUp)
```

We now simulate MCMC chains for several values of the leapfrog step sizes. We start of with very small values such that we get an acceptance rate of 100%:

```
MCMCOutput <- MCMCSampling(SetUp, 500, 10, c(0.001,0.001,0.001))
```

But as expected, plotting the chains it becomes obvious that the chains are static and the states are highly autocorrelated, as exemplified in Figure 11(a). We then proceed by testing separately for each parameter in which range the acceptance rate reacts sensitively to changes in the leapfrog step size. We then increase all parameters until the acceptance rate drops to values between 60% and 90%. We repeatedly use

```
PlotMCMCSamples(MCMCOutput)
acf(MCMCOutput$Samples)
```

to inspect the chains. If the autocorrelation of samples is unsatisfying we increase the number of leapfrog steps or subsample the chains. For the example above we might end up with

```
MCMCOutput <- MCMCSampling(SetUp, 500, 100, c(0.1,0.03,0.08))
```

for which an illustration is given in Figure 11(b).

A.4 Code Used In Experiments and Case Study

Finally we describe some of the code that was used in the reported experiments and the case study.

A.4.1 Synthetic Experiments

As described in Section 5.1 we conducted repeated experiments on artificially generated data. We illustrate the code for a simulation with $N = 90$ binomial trials and lapse rate $\pi_l = 0.05$. In each simulation first a data set is simulated:

```
nSamples <- 15
chance <- 0.5
alpha <- 0.1
type <- "gumbel"
trueTheta <- c(5,3)
truePlapse <- 0.05

# Find x values by looking at inverse of F
FLevels <- 2 * (c(0.97, 0.92, 0.87, 0.80, 0.65, 0.55) - 0.5)
z1 <- log(-log(0.5))
z2 <- log(-log(alpha))
z3 <- log(-log(1-alpha))
xValues <- (trueTheta[2]/(z2-z3)) * (log(-log(1-FLevels)) - z1) + trueTheta[1]

BinomialData <- matrix(data = NA, nrow = length(xValues), ncol = 3)
BinomialData[,1] <- xValues
BinomialData[,3] <- nSamples

# Sample the data from binomial distribution
for(i in 1:length(xValues)){
    F <-  PsychoFun(trueTheta, xValues[i], type=type, alpha=alpha)
    Psi <- (1-truePlapse) * ((1-chance) * F + chance) + truePlapse * chance
    BinomialData[i,2] <- rbinom(1,nSamples,Psi)
}
```

We then describe the priors, sample from the posterior and compute MAP and ML estimates:

```
SetUp <- PsychoSetUp(BinomialData,
                     plPrior="beta",
                     plPriorParameters=c(2,50),
                     t1Prior="normal",
                     t1PriorParameters=c(2,10),
                     t2Prior="lognormal",
                     t2PriorParameters=c(1,1),
                     type = type,
                     chance = chance,
                     alpha = alpha)

# Generate Samples from the posterior
MCMCOutput <- MCMCSampling(SetUp, 2000, 100, c(0.5,0.1,0.15), loud = FALSE)

# Compute quantiles of samples
MCMC <- SummaryPsychoSamples(MCMCOutput, SetUp, makePlots = FALSE)

# Maximum likelihood and MAP estimates
ML <- MLEstimation(SetUp)
MAP <- MAPEstimation(SetUp)
```

A.4.2 Case Study

The code in this section was used for the case study described in Section 5.2. First we analysed the sine condition.

```
library(PsychoFun)
set.seed(123)      # Set random seed for reproducibility

# load data from file
```

```
data(PsychoFunSineData)
PsychoFunSineData[,1] <- log10(PsychoFunSineData[,1])

# set up parameters
SineSetUp <- PsychoSetUp(PsychoFunSineData,
                         plPrior="beta",
                         plPriorParameters=c(2,50)
                         t1Prior="normal",
                         t1PriorParameters=c(-1,1),
                         t2Prior="normal",
                         t2PriorParameters=c(1,1),
                         type = "gumbel",
                         chance = 0.5,
                         alpha = 0.1)

# look at priors
PlotPsychoPriors(SineSetUp)

# have a first look at the data and the MAP and ML estimate
PlotData(SineSetUp)
MAP <- MAPEstimation(SineSetUp)
PlotPsi(MAP,SineSetUp,col='red',hold=TRUE)
ML <- MLEstimation(SineSetUp)
PlotPsi(ML,SineSetUp,col='blue',hold=TRUE)

# Sample
SineMCMC <- MCMCSampling(SineSetUp, 3000, 100, c(0.2,0.02,0.1))

# and check the Samples again
acf(SineMCMC$Samples)
PlotMCMCSamples(SineMCMC)

# this looks good, so here is the posterior distribution
# the first 100 Samples are discarded and we want to calculate
# some confidence intervals
SineQuantiles <- SummaryPsychoSamples(SineMCMC, SineSetUp,
                   burnIn = 100, quantiles = c(0.05, 0.25, 0.5, 0.75, 0.95))

# possible estimators for the function (the last column of
# SineQuantiles is the mean)
SineMean <- c()
SineMean$pLapse <- SineQuantiles[1,6]
SineMean$theta <- SineQuantiles[c(2,3),6]
SineMAP <- MAPEstimation(SineSetUp)
SineML <- MLEstimation(SineSetUp)

# look at the posterior distributions and priors
PlotPsychoPosterior(SineMCMC,SineSetUp,which=1,ylim=c(0,20),xlim=c(0,0.25))
PlotPsychoPosterior(SineMCMC,SineSetUp,which=2)
PlotPsychoPosterior(SineMCMC,SineSetUp,which=3)

# plots of the psychometric functions
PlotPsi(SineML,SineSetUp,col='green',lty=5,lwd=2)
PlotPsi(SineMean,SineSetUp,col='red',lty=1,lwd=2,hold=TRUE)
PlotPsi(SineMAP,SineSetUp,col='blue',lty=4,lwd=2,hold=TRUE,
        MCMCOutput=SineMCMC)
PlotData(SineSetUp,hold=TRUE)
```

To see the difference that the priors made we did the same analysis again with flat priors and compared the results.

```
SineSetUp2 <- PsychoSetUp(PsychoFunSineData,
                          plPrior="flat",
                          t1Prior="flat",
```

```
                          t2Prior="flat",
                          type = "gumbel",
                          chance = 0.5,
                          alpha = 0.1)

SineMCMC2 <- MCMCSampling(SineSetUp2, 3000, 100, c(0.1,0.01,0.05))

# check sampling
acf(SineMCMC2$Samples)
PlotMCMCSamples(SineMCMC2)
SineQuantiles2 <- SummaryPsychoSamples(SineMCMC2, SineSetUp2,
                    burnIn = 100, quantiles = c(0.05, 0.5, 0.95))

# compare flat vs. nice priors (specified prior blue, flat prior green)
# The lapse parameter is the first column of SineMCMC$Samples
# -- we convert the samples into a density and plot the density
plot.density(density(SineMCMC$Samples[,1]),
             col="blue",xlim=c(0,0.25),ylim=c(0,20),xlab=expression(pi[l]))
d <- density(SineMCMC2$Samples[,1])
lines(d$x,d$y,col="green")

# the threshold
plot.density(density(SineMCMC$Samples[,2]),col="blue",xlab='m')
d <- density(SineMCMC2$Samples[,2])
lines(d$x,d$y,col="green")

# and the width
plot.density(density(SineMCMC$Samples[,3]),col="blue",xlab='w')
d <- density(SineMCMC2$Samples[,3])
lines(d$x,d$y,col="green")
```

Finally, we analysed the pulse condition and compared it to the sine condition.

```
data(PsychoFunPulseData)
PsychoFunPulseData[,1] <- log10(PsychoFunPulseData[,1])
PulseSetUp <- PsychoSetUp(PsychoFunPulseData,
                          plPrior="beta",
                          plPriorParameters=c(2,50),
                          t1Prior="normal",
                          t1PriorParameters=c(-1,1),
                          t2Prior="normal",
                          t2PriorParameters=c(1,1),
                          type = "gumbel",
                          chance = 0.5,
                          alpha = 0.1)

PulseMCMC <- MCMCSampling(PulseSetUp, 3000, 100, c(0.2,0.05,0.1))
acf(PulseMCMC$Samples)
PlotMCMCSamples(PulseMCMC)
PulseQuantiles <- SummaryPsychoSamples(PulseMCMC, PulseSetUp,
                    burnIn = 100, quantiles = c(0.05, 0.25, 0.5, 0.75, 0.95))

# estimators for the function
PulseMean <- c()
PulseMean$pLapse <- PulseQuantiles[1,6]
PulseMean$theta <- PulseQuantiles[c(2,3),6]
PulseMAP <- MAPEstimation(PulseSetUp)
PulseML <- MLEstimation(PulseSetUp)

# plots of the psychometric functions
PlotPsi(PulseML,PulseSetUp,col='green',lty=5,lwd=2)
PlotPsi(PulseMean,PulseSetUp,col='red',lty=1,lwd=2,hold=TRUE)
PlotPsi(PulseMAP,PulseSetUp,col='blue',lty=4,lwd=2,hold=TRUE,
        MCMCOutput=PulseMCMC)
```

```
PlotData(PulseSetUp,hold=TRUE)

# Compare conditions -- first the lapse posteriors
plot.density(density(PulseMCMC$Samples[,1]),
             col="red",xlim=c(0,0.15),ylim=c(0,25),xlab=expression(pi[l]))
d <- density(SineMCMC$Samples[,1])
lines(d$x,d$y,col="blue")

# then the thresholds
plot.density(density(PulseMCMC$Samples[,2]),
             col="red",xlim=c(-2,-1.4),ylim=c(0,12),xlab='m')
d <- density(SineMCMC$Samples[,2])
lines(d$x,d$y,col="blue")

# then the widths
plot.density(density(PulseMCMC$Samples[,3]),
             col="red",xlim=c(0,2.5),ylim=c(0,2.5),xlab='w')
d <- density(SineMCMC$Samples[,3])
lines(d$x,d$y,col="blue")
```