Journal of the Korean Statistical Society (2023) 52:60–82
https://doi.org/10.1007/s42952-022-00189-x

Online ISSN 2005-2863
Print ISSN 1226-3192

RESEARCH ARTICLE

The phoropter method: a stochastic graphical procedure for prior elicitation in univariate data models

Christopher J. Casement[1,2] · David J. Kahle[2]

Received: 31 March 2022 / Accepted: 26 August 2022 / Published online: 19 September 2022

Abstract

Common methods for Bayesian prior elicitation call for expert belief in the form of numerical summaries. However, certain challenges remain with such strategies. Drawing on recent advances made in graphical inference, we propose an interactive method and tool for prior elicitation in which experts express their belief through a sequence of selections between pairs of graphics, reminiscent of the common procedure used during eye examinations. The graphics are based on synthetic datasets generated from underlying prior models with carefully chosen parameters, instead of the parameters themselves. At each step of the process, the expert is presented with two familiar graphics based on these datasets, billed as hypothetical future datasets, and makes a selection regarding their relative likelihood. Underneath, the parameters that are used to generate the datasets are generated in a way that mimics the Metropolis algorithm, with the experts' responses forming transition probabilities. Using the general method, we develop procedures for data models used regularly in practice: Bernoulli, Poisson, and Normal, though it extends to additional univariate data models as well. A free, open-source Shiny application designed for these procedures is also available online, helping promote best practice recommendations in myriad ways. The method is supported by simulation.

Keywords Bayesian statistics · Prior elicitation · Graphical inference · Markov chain Monte Carlo · Shiny

✉ Christopher J. Casement
casementc@gmail.com

[1] Department of Mathematics, Fairfield University, Fairfield, CT, USA

[2] Department of Statistical Science, Baylor University, Waco, TX, USA

1 Introduction

Eliciting a prior for a Bayesian analysis demands special care due to a prior's ability to strongly impact results. To this end, substantial literature has been written on prior elicitation. Garthwaite et al. (2005) provides an overarching summary of this literature while also detailing a four-stage process for prior elicitation that consists of four stages: (1) setup, (2) elicitation, (3) fitting, and (4) adequacy assessment. During the setup stage, the prior model, elicitation method to be used, and statistical summaries to be acquired from the expert are determined, and the expert is selected and trained. During the elicitation stage, the expert quantifies their knowledge in the form of the statistician's desired summaries, e.g. mean and percentiles. During the fitting stage, the statistician converts the summaries into the hyperparameters of the prior model. The exercise concludes with the adequacy assessment stage, in which the expert is asked to assess whether the resulting prior accurately reflects their beliefs.

While many methods and tools have been developed to assist this process (Morris et al., 2014; Oakley & O'Hagan, 2010; Su, 2006), certain drawbacks remain. One issue is that, despite training, it can be difficult for experts to provide reliable estimates of the quantities of interest. For example, it is not uncommon for the desired summaries to be convenient for the statistician but awkward for the expert due to a different cognitive framing of the phenomenon under study. Another, more serious challenge is the tendency to undervalue the elicitation process as a whole and focus resources almost exclusively on stages 2 and 3 (elicitation and fitting), which are the most mathematically challenging yet not necessarily the most important overall. These drawbacks undermine the process and threaten the quality of the analysis and its results.

In this article we use advances made in graphical inference to propose a novel graphical procedure for prior elicitation that tries to address these drawbacks. The resulting procedure promotes good practice by enforcing the elicitation process as a synergistic whole. Stage 1 of the Garthwaite program (setup) is done via the Rorschach procedure described in Sect. 3. For stage 2 (elicitation), rather than having the expert specify summaries directly, they make a series of specifications concerning the relative likelihood of observing two datasets illustrated by graphics chosen to be familiar to the expert, e.g. bar charts, histograms, or Kaplan–Meier curves. In stage 3 (fitting), the expert's specifications are converted into the parameter(s) of the prior model by fitting the model to the parameters used to create the datasets the expert selected in stage 2, which were proposed in such a way to enable such estimation. Stage 4 (assessment) is enforced throughout by the selection of the graphics; it is also supported through a graphical review of the process at the end. We call the resulting method the phoropter method, a nod to the procedure commonly used by eye doctors to assess visual acuity and determine corrective lens power that procedes through a sequence of binary comparisons ("one or two? two or three? one or three?...").

The process we outline is quite general for univariate data models, but we focus our attention on the four classic cases: (1) a Bernoulli data model with a beta prior,

(2) a Poisson data model with a gamma prior, (3) a normal data model (σ^2 known) with a normal prior, and (4) a normal (σ^2 unknown) data model with a normal-inverse-gamma prior. The descriptions used for these can be adapted for more complex modeling scenarios, e.g. regressions. Free web-based Shiny implementations for each of the scenarios are provided for free at https://ccasement.shinyapps.io/graphicalElicitationMCMC.

The article proceeds as follows. In Sect. 2 we discuss the prior elicitation process in practice: existing methods, advantages and disadvantages, and tools. In Sect. 3 we present two graphical procedures developed for statistical inference, the Rorschach and line-up, which we draw on for the proposed elicitation method. The proposed method follows in Sect. 4, followed by a demonstration of the Shiny app in Section 5. Sect. 6 provides probabilistic justification for the inner workings of the procedure. We conclude by summarizing the article in Sect. 7.

2 Prior elicitation in practice

Many prior elicitation methods have been developed. Nearly all of these are analytical in the sense that experts are asked to provide summaries of a parameter(s), and the summaries are converted into the hyperparameters of a predetermined prior model. Examples of such methods, which include the mode and percentile method, the probability density function method, the cumulative distribution function method, and the equivalent prior sample method, among others, are detailed in Garthwaite et al. (2005), Kahle et al. (2016), and O'Hagan et al. (2006).

To assist with the elicitation process, some of the methods have been implemented in software packages. These tools may perform a range of functions, from facilitating the interaction with the expert, e.g. stage 1 or stage 4 components, to performing the conversion in stage 3. Examples include SHELF, the Sheffield Elicitation Framework, which runs in R (Oakley & O'Hagan, 2010); MATCH, the Multidisciplinary Assessment of Technology Centre for Healthcare Uncertainty Elicitation Tool, a more comprehensive browser-based tool and set of materials from the same general team (Morris et al., 2014); BetaBuster, a Java applet that uses the mode/percentile method for eliciting a beta prior (Su, 2006); and Wolfram beta elicitation tool (Kahle et al., 2016), a Mathematica based tool that performs a similar function. Two additional tools, both interactive and graphical, operate in a web-based Shiny app (Casement & Kahle, 2018) and Microsoft Excel (Jones & Johnson, 2014).

All but one of these methods and tools have the experts speak directly to the value of the parameter. The others, developed by Casement and Kahle (2018), have the experts work directly with data. One concern with experts working with the value of the parameter is the expert's ability to accurately quantify summaries that represent their belief in a parameter. The elicitation method employed may be sensitive to changes in the summaries, which could, in turn, undermine the accuracy of an analysis, as prior distributions play an important role in determining posteriors. For instance, when using the mode and percentile method, small changes in the value of a percentile can dramatically alter how informative the resulting prior is (Blair, 2017).

Undervaluing the elicitation process as a whole is a second and perhaps more important concern. The statistician must always ensure the expert is properly trained prior to the elicitation stage and that they engage in an observational level with the implications of their specifications. Many studies have concluded that people often exert overconfidence in their beliefs that directly results in estimates that are overly precise (see, for example, Keren, 1991; Lichtenstein et al., 1977; Lichtenstein & Fischhoff, 1977; Oskamp, 1965). Such estimates jeopardize the accuracy of analyses if unduly informative priors are used, especially if the corresponding belief is incorrect. Of the elicitation tools discussed, only the Shiny app created by Casement and Kahle (2018) addresses the full elicitation process.

To address the concerns discussed in this section, we propose elicitation methods that not only simplify the process for the expert by allowing them to work with data rather than summaries of parameters, but that also treat the elicitation process holistically. While Casement and Kahle (2018) do both as well, their methods rely on algorithms that possess a deterministic foundation (as is discussed in Sect. 4), while the methods proposed in this article rely on a stochastic foundation, which we detail in Sects. 4, 5 and 6.

3 Graphical procedures for statistical inference

We now briefly turn our attention to two graphical procedures for inference: the Rorschach and line-up procedures (Buja et al., 2009; Wickham et al., 2010). After a discussion of both, we draw on them for the graphical elicitation methods proposed in Sect. 4.

3.1 Rorschach procedure

The Rorschach procedure is a graphical device used to train an individual to better understand the natural variability that exists in a stochastic model. When performing the procedure, G datasets of reasonable size are randomly generated from the same distribution, called the "null" distribution (Wickham et al., 2010), and graphs of the datasets are plotted next to one another. The individual examines and compares the graphs, focusing on features they do and do not expect given the particular null distribution. Those features, both expected and unexpected, can be attributed to the natural variability of the underlying null distribution for the sample size selected. For a fixed sample size, the data model induces a distribution on the set of graphics, and the Rorschach procedure allows an expert to learn this distribution through a series of observations. So long as the graphing method is sufficiently granular with respect to the sample space of the data, the two can be considered equivalent.

The Rorschach procedure is illustrated in Fig. 1 for a Beta$(0.5, 0.5)$ distribution and samples of size $n = 500$. Natural variability clearly appears in each graph, as

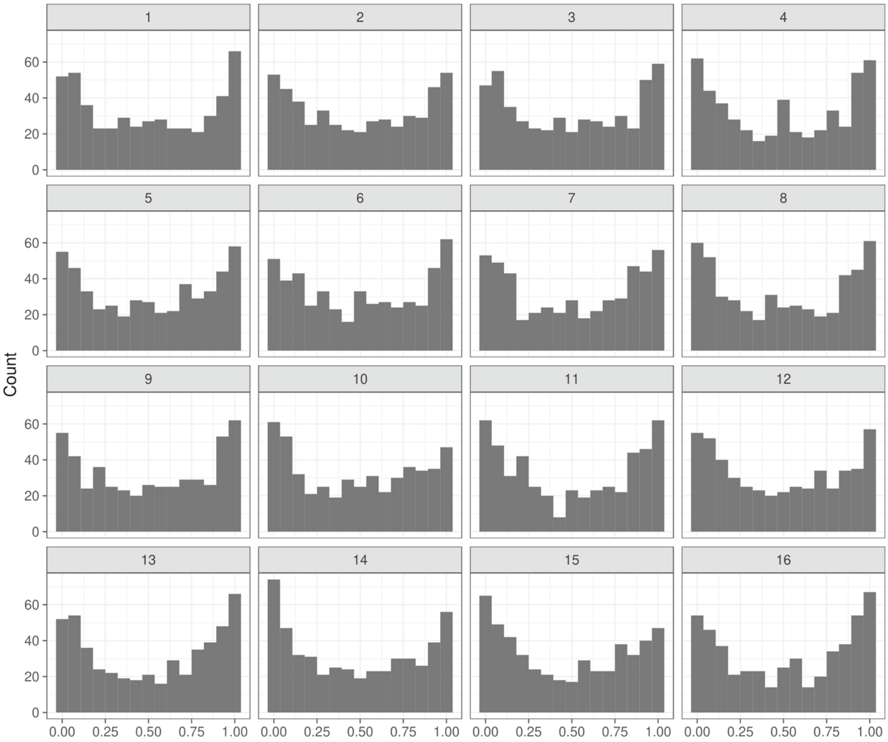


Fig. 1 Sixteen histograms, each of 500 random observations from the Beta(0.5, 0.5) distribution

none of the sixteen histograms shows a perfectly round bathtub shape. The Rorschach procedure thus improves an individual's ability to understand such variability for the Beta(0.5, 0.5) distribution and decreases their chance of concluding a dataset does not come from that distribution simply because its graph does not appear exactly as expected. While the Rorschach procedure involves the inspection of graphics without any selections made (e.g. best or worst), the next procedure described, the line-up, does. Thus, the Rorschach is a natural precursor to the line-up.

3.2 Line-up procedure

The line-up procedure is a graphical goodness-of-fit test. To test a model using the line-up procedure, the model is fit to the data and the resulting distribution treated as the null. Next, $G-1$ datasets, all of the same size as the initial dataset, are randomly generated from the null distribution, and graphs are plotted next to one another, with the plot of the real dataset randomly mixed in. The individual is then tasked with

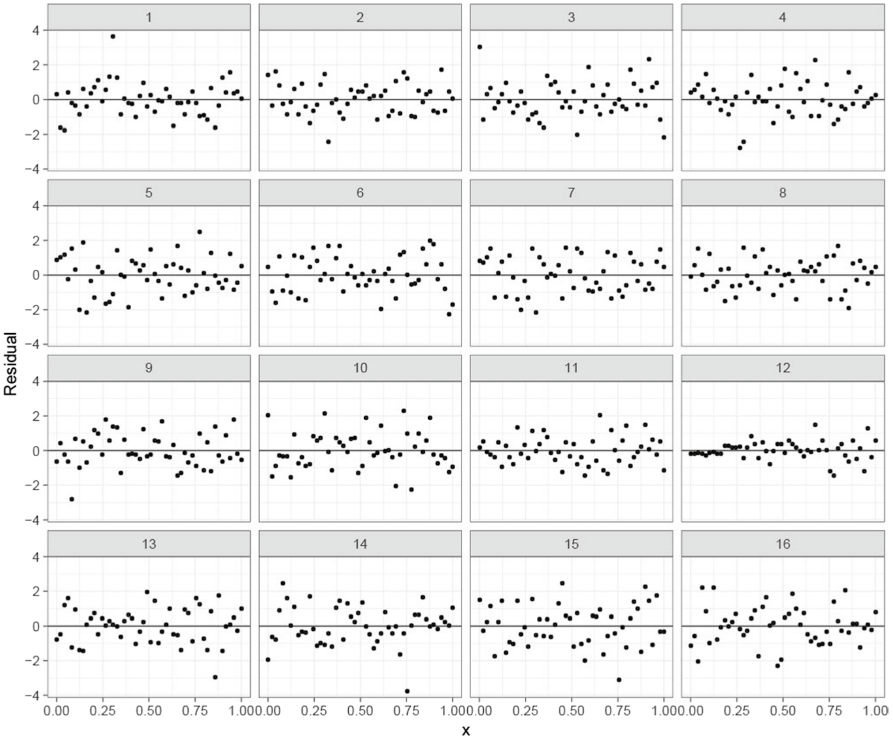


Fig. 2 Fifteen scatterplots of simulated residuals, one of real residuals randomly mixed in

selecting the graph of the real data. If they successfully choose the real dataset, the null hypothesis is rejected, suggesting the real data do not come from the null distribution. This is illustrated for assessing the homogeneity of variance assumption in a linear model in Fig. 2. For the solution to the example, see footnote.[1]

The two graphical inference procedures described here—the Rorschach and line-up—are well suited for the graphical elicitation methods described in the following section for two main reasons. First, the procedures are accessible to non-statisticians, as they do not require a strong statistical background. Generally experts are familiar with common statistical graphics, and these can be adapted to the context at hand. Second, the line-up—especially after the individual is trained using the Rorschach—allows them to visually assess whether a difference between graphics is meaningful based on their expertise.

[1] Graph 12 displays the real data.

4 Stochastic graphical elicitation

In this section we describe the proposed method in detail. Before proceeding to the full description, however, we provide a colloquial explanation of how the method works on a simple paradigmatic example, that of eliciting a beta prior for a binomial probability. We then build on this to provide a full description of the scheme.

To begin, it is easiest to first understand a similar deterministic method proposed by the present authors in Casement and Kahle (2018). To that end, suppose a data model f_θ is assumed and a prior $p_{\boldsymbol{\eta}}$ on θ is desired. In particular, following the paradigm suppose that f_θ is a binomial data model and a beta prior distribution $p_{\boldsymbol{\eta}}$ is desired, so $\boldsymbol{\eta} = (\eta_1, \eta_2) = (\alpha, \beta)$. Using the strategy of Casement and Kahle (2018), the expert is presented with $G = 5$ (say) bar charts based on synthetic datasets of a fixed sample size N, where each dataset is generated using a different θ value over a range of such values, and the expert is tasked with selecting the most realistic dataset. After that is selected, another collection of G datasets is generated and displayed, and the expert is again tasked with selecting the most realistic. At each stage of the procedure the range of θ values used is refined so as to hone in on a specific value the expert implicitly holds to be the most likely value of θ. This value is considered to be the mode of the beta prior. Next, the expert is asked to specify about how many observations their past experience is based upon; this is understood to be the effective sample size (ESS) of the beta prior (Morita et al., 2008). This pair of values can be then converted into the canonical (α, β) parameters of the beta model.

In this article we advance a similar method that does not require the expert's specification of the ESS. The basic motivation exchanges the deterministic "honing in" strategy for randomly proposed values of θ that are then accepted or rejected probabilistically in a manner similar to the Metropolis algorithm. Instead of being presented with $G = 5$ graphics, the expert is presented with $G = 2$ graphics, each with the same sample size but generated using a current value of θ and a proposed one. The expert specifies which is more likely to be observed in practice and roughly to what extent, and then the expert is presented with another such pair of graphics. After many such selections, the system is able to infer the expert's belief distribution from the prior model. Thus, the proposed method is a kind of stochastic version of the method described in Casement and Kahle (2018).

4.1 Uniparameter data models

Elaborating on the above description, in the first step of the procedure the expert is asked to provide a typical observation x for the hypothetical future dataset of size N. This value x is converted to $\theta^{(0)}$ to initialize the algorithm. For example, for a

Bernoulli data model the initial parameter $\theta^{(0)} = p^{(0)}$ is set at x/N, where x represents a typical number of successes in a trial of size N. For a Poisson or normal data model with σ^2 known, the initial parameter is set at $\lambda^{(0)} = x$ and $\mu^{(0)} = x$, respectively. The expert then goes through Rorschach training using the distribution determined from this value.

After the expert is comfortable, they move on to the graphical generation and selection process. A proposed parameter value θ^{prop} is drawn from a $\mathcal{N}\big(\theta^{(0)}, \sigma_\theta\big)$ proposal distribution; this proposal is used for all scenarios in this article but could be adapted for different ones. Next, N values $\boldsymbol{u}$ are independently drawn from a Unif(0, 1) distribution and converted to observations from the distribution of interest using inverse transform sampling. This results in two datasets: $\boldsymbol{x}^{\text{current}} = F^{-1}_{\theta^{(0)}}(\boldsymbol{u})$ and $\boldsymbol{x}^{\text{prop}} = F^{-1}_{\theta^{\text{prop}}}(\boldsymbol{u})$. The conversion is done in this way to allow for a direct comparison of the current and proposed parameter values based on a single original sample. Graphically, this amounts to the same dataset being either shifted or scaled (with possible re-binning for histograms) depending on the parameter of interest. Such a choice is used to best maintain the Metropolis foundation seen shortly.

Presented with graphical displays of $\boldsymbol{x}^{\text{current}}$ and $\boldsymbol{x}^{\text{prop}}$, the expert is tasked with a selection from a set of five options: the proposed plot is more likely than the current, both plots are equally likely, or three options representing how much more likely the current plot is than the proposed plot. If the expert states the proposed graphic is more or equally likely, the proposed step is accepted as the new state of the sampler. These mimic the transition probabilities of a Metropolis sampler. On the other hand, if the expert feels the current graphic is more likely than the proposed, then they must choose how much more likely the graphic is. In the Metropolis algorithm, this value is used to determine the transition probability. Since the specification of such a precise number by an expert seems both implausible and cumbersome, we restrict this number—the odds of the proposed plot relative to the current plot—to one of three values determined via simulation to provide optimal results (see Sect. 6): $\boldsymbol{O} = \big\{3, 25, 10^6\big\}$. The expert's choice of odds then determines the probability with which the proposed value will be accepted, as the odds and transition probability α are in one-to-one correspondence through the relationship

$$\alpha = \frac{1}{O_i^{\text{selected}}}.$$

The expert is asked to make a total of M selections, with M large enough such that the algorithm has had sufficient time to provide a stable estimate of $\boldsymbol{\eta}$, for instance via likelihood estimation. This procedure is summarized in Algorithm 1.

Algorithm 1 Graphical prior elicitation for uniparameter data models.

```
Inputs:
    $N$ – the number of observations in a hypothetical future dataset
    $x$ – a typical observation for the hypothetical future dataset
    $f_\theta, F_\theta^{-1}$ – the PDF/quantile function of the data model, $\theta \in \Theta$
    $p_{\boldsymbol{\eta}}$ – the prior model family parameterized by $\boldsymbol{\eta} \in \boldsymbol{H}$
    $\sigma_\theta$ – the proposal standard deviation for $\theta$
    $\boldsymbol{O}$ – the odds set, e.g. $(1, 3, 25, 10^6)$
    $M$ – the total number of selections made, e.g. 100

Output: $\boldsymbol{\eta}^\star$, the hyperparameters for the prior on $\theta$

$\theta^{(0)} \leftarrow$ the $\theta$ corresponding to the $x$ input by the expert
for $t = 0, \ldots, M-1$ do
    Sample $\boldsymbol{u} = (u_1, \ldots, u_N) \overset{iid}{\sim} \text{Unif}(0,1)$
    Sample $\theta^{\text{prop}} \sim \mathcal{N}\left(\theta^{(t)}, \sigma_\theta\right)$
    $\boldsymbol{x}^{\text{current}} \leftarrow F_{\theta^{(t)}}^{-1}(\boldsymbol{u})$ and $\boldsymbol{x}^{\text{proposed}} \leftarrow F_{\theta^{\text{prop}}}^{-1}(\boldsymbol{u})$
    Construct adjacent graphics with $\boldsymbol{x}^{\text{current}}$ and $\boldsymbol{x}^{\text{proposed}}$
    $O^\star \leftarrow$ the $O_i$ corresponding to the expert's selection
    $\alpha \leftarrow \min\left(1, 1/O^\star\right)$
    Sample $v \sim \text{Unif}(0,1)$
    if $v < \alpha$ then $\theta^{(t+1)} \leftarrow \theta^{\text{prop}}$ else $\theta^{(t+1)} \leftarrow \theta^{(t)}$
end for
$\boldsymbol{\eta}^\star \leftarrow \underset{\boldsymbol{\eta} \in \boldsymbol{H}}{\text{argmax}} \prod_{i=1}^{M} p_{\boldsymbol{\eta}}(\theta^{(i)})$
```

4.2 Multiparameter data models

The procedure as previously described naturally generalizes to any uniparameter data model; however, the scenario is more complicated when data models with more than one parameter are used. We now discuss the procedure for a multiparameter data model, the $\mathcal{N}\big(\mu, \sigma^2\big)$ with both μ and σ^2 unknown.

For data models with only one unknown parameter θ, the expert makes M total selections pertaining to θ. The $\mathcal{N}\big(\mu, \sigma^2\big)$ data model with σ^2 unknown has two unknown parameters $\boldsymbol{\theta} = (\mu, \sigma^2)$, and the expert must make M selections for each. We chose the conjugate prior for this model, a normal-inverse-gamma($\mu_0, \lambda, \alpha, \beta$) prior on $\boldsymbol{\theta}$, as is common practice and as was done for the models discussed previously. The MLE $\hat{\boldsymbol{\eta}} = (\hat{\mu}_0, \hat{\lambda}, \hat{\alpha}, \hat{\beta})$ of the hyperparameters $\boldsymbol{\eta} = (\mu_0, \lambda, \alpha, \beta)$ is then computed once the expert has completed their selections.

The first step of the procedure entails asking the expert for two quantities: a typical measurement value x for a hypothetical future dataset of size N, and the largest possible reasonable value x_u for the dataset. Again, these values are only intended to calibrate the algorithm; they need not be correct in any conventional sense, and their precision is irrelevant. We then obtain initial values for the sampler by setting $\theta_1^{(0)} = x$ and $\theta_2^{(0)} = \frac{x_u - x}{3}$, the latter of which is used as a rough application of the empirical rule.

A proposed mean θ_1^{prop} is then drawn from a $\mathcal{N}\left(\theta_1^{(0)}, \sigma_{\theta_1}\right)$ proposal distribution. Next, a sample $\boldsymbol{u}$ of size N is randomly drawn from a Unif(0,1) distribution and inverse-transformed using the normal quantile function with means of $\theta_1^{(0)}$ and θ_1^{prop} and a common variance of $\theta_2^{(0)}$: $\boldsymbol{x}^{\text{current}} = F^{-1}_{\theta_1^{(0)}, \theta_2^{(0)}}(\boldsymbol{u})$ and $\boldsymbol{x}^{\text{prop}} = F^{-1}_{\theta_1^{\text{prop}}, \theta_2^{(0)}}(\boldsymbol{u})$. Histograms of $\boldsymbol{x}^{\text{current}}$ and $\boldsymbol{x}^{\text{prop}}$ are then displayed side-by-side, and the expert must select one of five options, the same as those presented for the other data models. If the expert selects an option where the proposed and current plots are equally likely or where the proposed plot is more likely, then θ_1^{prop} becomes the new current mean step $\theta_1^{(1)}$ in the sampler, otherwise the proposal is accepted with probability $1/O_i^{\text{selected}}$. If θ_1^{prop} is rejected, $\theta_1^{(1)} = \theta_1^{(0)}$.

After one proposal in the mean parameter, the algorithm next turns its attention to the variance parameter, with the mean fixed at $\theta_1^{(1)}$. A proposed variance θ_2^{prop} is drawn from a $\mathcal{N}\left(\theta_2^{(0)}, \sigma_{\theta_2}\right)$ proposal distribution, and the previous process is repeated: $\boldsymbol{x}^{\text{current}} = F^{-1}_{\theta_1^{(1)}, \theta_2^{(0)}}(\boldsymbol{u})$ and $\boldsymbol{x}^{\text{prop}} = F^{-1}_{\theta_1^{(1)}, \theta_2^{\text{prop}}}(\boldsymbol{u})$. Histograms of $\boldsymbol{x}^{\text{current}}$ and $\boldsymbol{x}^{\text{prop}}$ are again displayed side-by-side; the expert selects one of the five options; and $\theta_2^{(1)}$ is determined based on the option selected. The algorithm then refocuses on the mean and the process continues.

The procedure continues to cycle between the mean and variance, allowing only one parameter to vary at a time while fixing the other at its current value. This kind of movement has one key advantage: it allows the expert to make judgments according to one parameter at a time. If the variables were both proposed simultaneously, one might envision a scenario where an expert likes one graphic's location more but another one's scale more, and then cannot decide which to select.

The algorithm terminates after the expert has made M selections for each parameter, at which point hyperparameters are found by computing the MLE of the joint prior, resulting in a normal-inverse-gamma($\hat{\mu}_0$, $\hat{\lambda}$, $\hat{\alpha}$, $\hat{\beta}$) prior on (μ, σ^2). The full procedure is presented in Algorithm 2.

Algorithm 2 Graphical prior elicitation for $\mathcal{N}\left(\mu, \sigma^2\right)$.

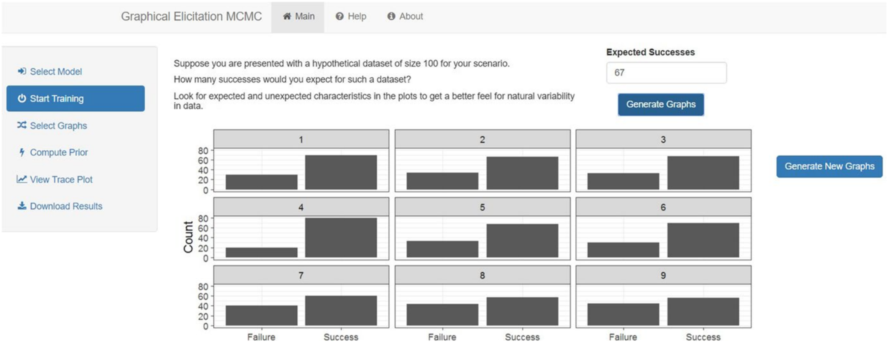
```
Inputs:
    $N$ – the number of observations in a hypothetical future dataset
    $x$ – a common measurement value for the hypothetical future dataset
    $x_u$ – the largest possible value for the hypothetical future dataset
    $f_{\boldsymbol{\theta}}, F_{\boldsymbol{\theta}}^{-1}$ – the PDF/quantile function of the data model, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$
    $p_{\boldsymbol{\eta}}$ – the normal-inv.-gamma prior parameterized by $\boldsymbol{\eta} = (\mu_0, \lambda, \alpha, \beta) \in \boldsymbol{H}$
    $\sigma_\theta$ – the proposal standard deviation for $\theta \in \Theta$
    $\boldsymbol{O}$ – the odds set, e.g. $(1, 3, 25, 10^6)$
    $M$ – the total number of selections made, e.g. 100

Output: $\boldsymbol{\eta}^\star$, the hyperparameters for the prior on $\boldsymbol{\theta}$

$\theta_1^{(0)} \leftarrow$ the $\theta_1$ corresponding to the $x$ input by the expert
$\theta_2^{(0)} \leftarrow (x_u - x)/3$
for $t = 0, \ldots, M-1$ do
    for $i = 1, 2$ do
        Sample $\boldsymbol{u} = [u_1 \cdots u_N] \overset{iid}{\sim}$ Unif(0, 1)
        Sample $\theta_i^{\text{prop}} \sim \mathcal{N}\left(\theta_i^{(t)}, \sigma_{\theta_i}\right)$
        if $i = 1$ then $\boldsymbol{x}^{\text{current}} = F^{-1}_{\theta_i^{(t)}, \theta_{i+1}^{(t)}}(\boldsymbol{u})$ and $\boldsymbol{x}^{\text{prop}} = F^{-1}_{\theta_i^{\text{prop}}, \theta_{i+1}^{(t)}}(\boldsymbol{u})$
        else $\boldsymbol{x}^{\text{current}} = F^{-1}_{\theta_{i-1}^{(t+1)}, \theta_i^{(t)}}(\boldsymbol{u})$ and $\boldsymbol{x}^{\text{prop}} = F^{-1}_{\theta_{i-1}^{(t+1)}, \theta_i^{\text{prop}}}(\boldsymbol{u})$
        Construct adjacent graphics with $\boldsymbol{x}^{\text{current}}$ and $\boldsymbol{x}^{\text{prop}}$
        $O^\star \leftarrow$ the $O_i$ corresponding to the expert's selection
        $\alpha \leftarrow \min\left(1, 1/O^\star\right)$
        Sample $v \sim$ Unif(0, 1)
        if $v < \alpha$ then $\theta_i^{(t+1)} \leftarrow \theta_i^{\text{prop}}$ else $\theta_i^{(t+1)} \leftarrow \theta_i^{(t)}$
    end for
end for
$\boldsymbol{\eta}^\star \leftarrow \underset{\boldsymbol{\eta} \in \boldsymbol{H}}{\operatorname{argmax}} \prod_{i=1}^{M} p_{\boldsymbol{\eta}}(\boldsymbol{\theta}^{(i)})$
```

5 Shiny app

A free Shiny app that implements the methods previously described in R can be found at https://ccasement.shinyapps.io/graphicalElicitationMCMC (Chang et al., 2021; R Core Team, 2021). Its source code, which can be used as a template for other such apps as well as to check the implementation, can be found on GitHub at https://github.com/ccasement/graphicalElicitationMCMC. In this section we



Suppose you are presented with a hypothetical dataset of size 100 for your scenario.

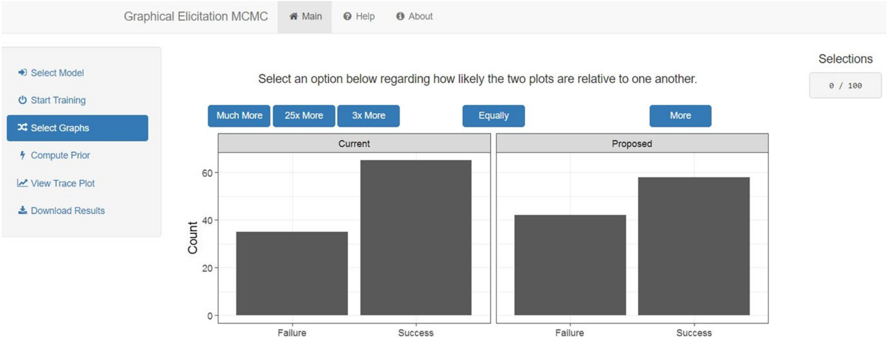
How many successes would you expect for such a dataset?

Look for expected and unexpected characteristics in the plots to get a better feel for natural variability in data.

Expected Successes

67

Fig. 3 Before the graphical selection process begins, the expert can go through Rorschach training. Each bar chart displays a random sample of size $n = 100$ from the Bernoulli($p = 0.67$) distribution



Select an option below regarding how likely the two plots are relative to one another.

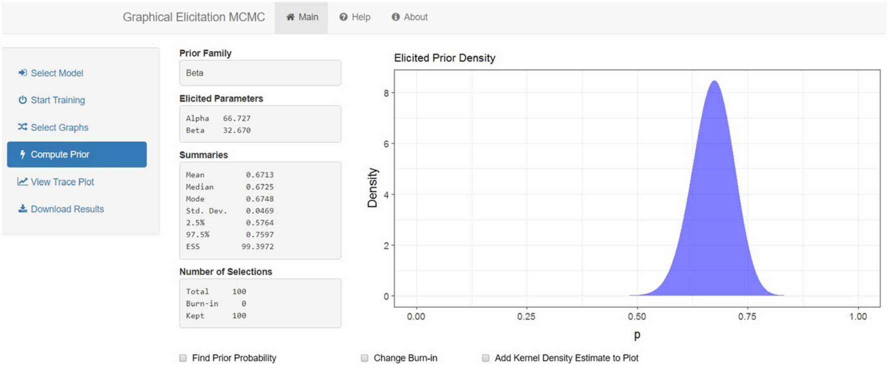
Selections

0 / 100

Fig. 4 At each step of the selection process, two graphics are presented to the expert, along with options for choosing between them

demonstrate the elicitation of a beta prior for a Bernoulli proportion through a series of screenshots of the app.

Suppose the expert expects $x = 67$ successes out of a hypothetical future dataset of size $N = 100$. They then proceed to the Rorschach training stage, where they are able to inspect randomly-generated datasets from Bernoulli distributions. Figure 3 displays bar charts of nine datasets of size $N = 100$ randomly generated from a Bernoulli($p = 0.67$) distribution, where the proportion of 0.67 corresponds to the $x/N = 67/100$ input by the expert. The expert is also able to go through Rorschach training for other Bernoulli distributions and can generate new sets of random samples for further training.



Prior Family

```
Beta
```

Elicited Parameters

```
Alpha   66.727
Beta    32.670
```

Summaries

```
Mean         0.6713
Median       0.6725
Mode         0.6748
Std. Dev.    0.0469
2.5%         0.5764
97.5%        0.7597
ESS         99.3972
```

Number of Selections

```
Total     100
Burn-in     0
Kept      100
```

Fig. 5 After the expert finishes the selection process, information regarding the elicited prior is presented

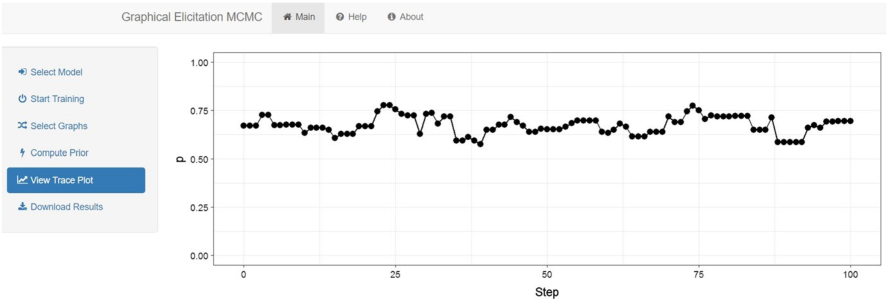


Fig. 6 A trace plot of the accepted proportions is also available

After finishing the Rorschach training the expert moves to the graphical elicitation procedure detailed in Sect. 4. Two bar charts—one for a current parameter value and another for a proposed parameter value—are presented to the expert as displayed in Fig. 4. The expert selects one of the five buttons in blue, and new current and proposed datasets are generated and plotted.

The selection process concludes and a prior is computed once the expert has made 100 selections (although this number is arbitrary). Figure 5 displays information about the prior that is provided to the expert: the elicited prior family and the estimated hyperparameters, summaries of the prior, and a density plot. These plots and summaries enable the facilitator and expert to assess the adequacy of the prior distribution. In fact, additional options are provided to aid in the assessment, including one that allows users to view a kernel density estimate of the selections and another that calculates the probability the proportion is between any two values they specify. The app also provides a trace plot of the chain, as shown in Fig. 6. Further, the expert can download a PDF report of the results as well as a CSV file containing the chain.

6 Technical considerations

In this section we turn to technical details validating the inferential process running under the hood of the procedure described above. In particular, we address two main concerns: (1) how do we know that the proposed process will produce draws from the desired prior?; and (2) why is the odds set selected as it is?

The underlying mechanism that the procedure follows is that of the classic Metropolis algorithm, a Markov chain Monte Carlo (MCMC) algorithm [for in-depth discussions on common MCMC methods, see Robert and Casella (2004)], using a Gaussian proposal with one assumption and one twist. The twist is that the transition probabilities α associated with the procedure are rounded to those dictated by the odds set. We call this algorithm—a Metropolis algorithm with rounded transition probabilities—a rigid Metropolis algorithm. Recall that the transition probability associated with the Metropolis algorithm is

$$\alpha = \min\left(1, \frac{p(\theta^{\text{prop}})}{p(\theta^{(t)})}\right). \tag{1}$$

Here, of course, $p(\theta)$ is the desired prior distribution representing the expert's knowledge, presumably acquired as the posterior of the expert's previous experience with an uninformative prior. If α is allowed to be as defined above, i.e. if the expert could specify their probabilities as precise real values, standard theory dictates that the draws we obtain from the process are in fact derived from $p(\theta)$.

The question we address in this section is how the rounding affects that standard theory, and in particular, how can the odds set, in this context called the rigid set, be selected so as to minimize the effect of the rounding? The answer is given by simulation: we simulate the rigid Metropolis algorithm using different odds sets and compare the distribution of those draws to the distribution of the draws we would have obtained had we not rounded the values. The comparison comes in the form of the total variation distance, described next. In brief, the answers to the above questions are that we have performed extensive simulations that demonstrate that so long as the odds set is chosen carefully, the difference between the rigid Metropolis algorithm used in the app and the standard Metropolis algorithm is quantifiably small.

Before proceeding, we also need to state the assumption: that the expert is able to determine the ratio of prior densities from the two graphics. That is to say: that the expert's specification of how much more likely the first plot is relative to the second in fact approximates the ratio $p(\theta^{\text{prop}})/p(\theta^{(t)})$. This can to some extent be made technical. For a fixed θ and graphical type (e.g. histogram with specified bin width and boundary), the data model induces a probability measure on the space of graphics. For the two graphics provided, the expert's judgment is therefore equivalent to the ratio of likelihoods with respect to that measure. In this work we assume that quantity is not materially different than the desired quantity $p(\theta^{\text{prop}})/p(\theta^{(t)})$.

6.1 Total variation distance

To measure the departure of the steady-state distribution from that of the Metropolis algorithm, we use the total variation distance (TVD). The TVD is a classic metric on the space of probability measures defined on the same measurable space $(\Omega, \mathcal{B})$. For probability measures P and Q on $(\Omega, \mathcal{B})$, the TVD is defined

$$\delta(P, Q) = \sup_{A \in \mathcal{B}} |P(A) - Q(A)|. \tag{2}$$

It is the maximal discrepancy the two measures can ascribe to an event. A remarkable fact about the TVD between probability distributions that admit densities with respect to the Lebesgue measure on $\mathbb{R}$ is that it can be easily computed via an integral involving those densities:

$$\delta(P, Q) = \frac{1}{2} \int_{\mathbb{R}} |p(x) - q(x)| dx, \tag{3}$$

a proof of which can be found in Resnick (2013).

To evaluate this integral, as well as to perform other TVD calculations in this article, we use `TotalVarDist()` from the **distrEx** R package (Ruckdeschel et al., 2006). In the cases described, `TotalVarDist()` typically uses R's `stats::integrate()` function for numerical integration, which uses an adaptive Gauss-Kronrod quadrature scheme.

6.2 Rigid Metropolis algorithm

To understand the simulations some notation is required first. To approximate the Metropolis algorithm with a rigid Metropolis one, we round the transition probability α to the nearest probability $\hat{\alpha}$ in a pre-specified vector of r probabilities $\mathcal{P} = \left\{p_1^{\text{rigid}}, \ldots, p_r^{\text{rigid}}\right\}$ that we call the rigid set, resulting in a $(100\hat{\alpha})\%$ chance the proposed step is accepted. The remainder of the algorithm follows that of standard Metropolis.

As in any Metropolis procedure, certain tuning parameters need to be set before implementing the algorithm. In this article we always use a normal distribution for generating proposals, and thus an appropriate standard deviation must be set. However, the rigid aspect of the Metropolis procedure entails two more selections: the number of probabilities in the rigid set and their distribution in (0, 1]. Ideally, for a given number of such probabilities they are determined in such a way that the asymptotic distribution of the rigid Metropolis procedure is as close as possible to the asymptotic distribution of the standard Metropolis procedure. In this article we propose a strategy to obtain near-optimal rigid sets of lengths $r \geq 2$ and discuss simulation results for scenarios where $r = 2$, 3, and 4.

6.2.1 TVD for rigid Metropolis

When using MCMC procedures in a Bayesian analysis, the target distribution is typically the joint posterior distribution of the parameters. To assess the ability of rigid Metropolis to accurately approximate these distributions, we consider the TVD between the target posterior and that found using the rigid Metropolis procedure.

Suppose a data model $f_{\boldsymbol{\theta}}$ is assumed with $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ for a dataset $\boldsymbol{y}$, and an uninformative conjugate prior $p_{\boldsymbol{\eta}}$ with $\boldsymbol{\eta} \in \boldsymbol{H}$ is assumed for $\boldsymbol{\theta}$. The dataset $\boldsymbol{y}$ is based on the expert's previous experiences, and the prior $p_{\boldsymbol{\eta}}$ represents the expert's beliefs about $\boldsymbol{\theta}$ before such experiences. We first find the target posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$, the asymptotic distribution of a chain using standard Metropolis, which is known for all distributions in this article but more generally could be estimated using the standard Metropolis sampler. We then compare this to the distribution $\hat{p}(\boldsymbol{\theta}|\boldsymbol{y})$ estimated from the draws resulting from the rigid Metropolis process.

In this article we consider three common data models—the Bernoulli(p), Poisson(λ), and $\mathcal{N}\left(\mu, \sigma^2\right)$ with σ^2 known—and conjugate priors Beta(α, β), Gamma(α, β), and $\mathcal{N}\left(\mu_0, \sigma_0^2\right)$. It is worth noting that in the Bernoulli case, for example, we assume that the prior is a Beta$(1, 1)$ so that the target posterior is also a member of the beta family by conjugacy. Under standard Metropolis, that is also the stationary distribution of the chain. Through simulation, this appears to be the case for the rigid Metropolis as well, and we consequently used its samples to fit a beta, but in general it need not be. This does not present a problem for the algorithm: one can simply estimate $p(\theta|\boldsymbol{y})$ directly from the data using any standard technique and then use $\hat{p}(\theta|\boldsymbol{y})$ in the integral formulation of the TVD.

With the posteriors $p(\theta|\boldsymbol{y})$ and $\hat{p}(\theta|\boldsymbol{y})$ in hand, we can compute the TVD between them:

$$\delta(p, \hat{p}) = \frac{1}{2} \int_{\Theta} |p(\theta|\boldsymbol{y}) - \hat{p}(\theta|\boldsymbol{y})| d\theta, \tag{4}$$

where $p(\theta|\boldsymbol{y})$ and $\hat{p}(\theta|\boldsymbol{y})$ are the PDFs corresponding to the target and rigid posteriors. We elected to use the form of the TVD specified in (4) due to the ease with which it can be computed numerically between continuous distributions, a characteristic of all posteriors covered in this article.

Working with rigid Metropolis, however, requires a pre-specified vector of probabilities $\mathcal{P} = \left\{p_1^{\text{rigid}}, ..., p_r^{\text{rigid}}\right\}$ of length r. We now formulate a procedure for determining an optimal $\mathcal{P}$.

6.2.2 Rigid Metropolis transition probabilities

When determining appropriate rigid probabilities for $\mathcal{P} = \left\{p_1^{\text{rigid}}, ..., p_r^{\text{rigid}}\right\}$, clearly r should be at least two, and those should indicate roughly “do not move” or “move”: $p_1^{\text{rigid}} = 10^{-6}$ and $p_r^{\text{rigid}} = 1$, for all lengths r. This smallest probability virtually prevents the rigid Metropolis algorithm from getting stuck in an undesirable part of the parameter space. Next, the largest probability in $\mathcal{P}$ was fixed at 1. Such is the case when comparing proposed and current parameters in standard Metropolis – if the proposed step is at least as equally likely as the current step, the proposed is accepted with probability one.

With the lowest and highest values in $\mathcal{P}$ set, we now turn to finding additional probabilities for $\mathcal{P}$. For data model f_θ, fixed number of probabilities r, and underlying parameter value θ^* of θ, our initial goal is to find the set of probabilities that minimizes the expected total variation distance between $p(\theta|\boldsymbol{y})$ and $\hat{p}(\theta|\boldsymbol{y})$:

$$\mathcal{P}^{\star}\left(\theta^*\right) = \underset{\mathcal{P}\subset(0,1],|\mathcal{P}|=r}{\operatorname{argmin}} \; E\left[\delta\big(p(\theta|\boldsymbol{y}), \hat{p}(\theta|\boldsymbol{y})\big)\,\middle|\,\theta^*, \mathcal{P}\right], \tag{5}$$

where the expectation is taken with respect to f_{θ^*}. Solving this optimization problem exactly for a given θ^*, however, is analytically intractable in general, so computational methods are used. Accounting for the complexity of the process discussed later in this section, while simultaneously ensuring the theoretical MCMC foundation of the process is maintained, we considered $p_i^{\text{rigid}} \in \{0.02, 0.04, 0.06, ..., 0.98\}$ as candidate probabilities for $p_2^{\text{rigid}}, ..., p_{r-1}^{\text{rigid}} \in \mathcal{P}$, when $r > 2$.

To minimize (5) for data model f_θ, and for a fixed θ^* and rigid set of size $r \in \{2, 3, 4, ...\}$, we run a grid-search style process as follows. First randomly generate a sample $\boldsymbol{y}$ of size n from f_{θ^*}. Second, find the target posterior $p(\theta|\boldsymbol{y})$ for θ; this is a simple task since each of the priors used is conjugate. Next, approximate the posterior $\hat{p}(\theta|\boldsymbol{y})$ for θ based on M iterations of the rigid Metropolis algorithm. In our simulations, this is done in a two-stage process: first generate the posterior values with the rigid Metropolis algorithm, rounding the true Metropolis transition probability to the closest in the rigid set, and then use those to fit parameter values $\hat{\boldsymbol{\eta}}$ (e.g. α and β of the beta distribution if the posterior is known to be beta) using maximum likelihood. After approximating $\hat{p}(\theta|\boldsymbol{y})$, calculate the TVD $\delta\big(p(\theta|\boldsymbol{y}), \hat{p}(\theta|\boldsymbol{y})\big)$ between the two posteriors according to (4). This is actually quite simple, since the true posterior $p(\theta|\boldsymbol{y})$ and $\hat{p}(\theta|\boldsymbol{y})$ are both members of the same conjugate family, so they simply correspond to two distributions with different parameter values. One is known directly from the simulation setup and conjugacy, $p(\theta|\boldsymbol{y}) = p(\theta|\boldsymbol{y}, \boldsymbol{\eta}^*)$, and one is determined through estimation, $\hat{p}(\theta|\boldsymbol{y}) = p(\theta|\boldsymbol{y}, \hat{\boldsymbol{\eta}})$. Perform this process T

times and average the total variation distances. Then repeat this entire procedure for all c possible combinations of r candidate probabilities in the rigid set, resulting in c average total variation distances. With the goal being to minimize $E\left[\delta\left(p(\theta|\mathbf{y}),\hat{p}(\theta|\mathbf{y})\right)\middle|\theta^{*},\mathcal{P}\right]$, select the $\mathcal{P}$ that results in the smallest average total variation distance $\bar{\delta}\left(p(\theta|\mathbf{y}),\hat{p}(\theta|\mathbf{y})\right)$.

With a procedure in place to obtain an optimal $\mathcal{P}$ for a given data model f_{θ} with a fixed parameter value θ^{*}, we now turn to our ultimate goal of obtaining an optimal $\mathcal{P}$ for a given f_{θ}. To do so, we explore how $\mathcal{P}^{\star}$ varies across Θ for a given f_{θ} by employing the procedure described above for various values of $\theta^{*} \in \Theta$. We analyze the results of this full process for three data models in Section 6.3.

6.2.3 Rigid Metropolis proposal standard deviations

The choice of a proposal standard deviation σ_{θ} with data model f_{θ} is another important consideration when working with MCMC methods. If σ_{θ} is too small, then the sampler will not be very efficient; its values will be highly autocorrelated. As a consequence for our application, the expert will have difficulty distinguishing between the proposed and current plots, as the proposed and current parameter values will often be close in magnitude. Additionally, far too many selections will be demanded of the expert. On the other hand, if σ_{θ} is too large, many proposed values will be rejected, as they represent unrealistic scenarios or, in some cases, will be outside the parameter space.

As the expert makes a number of selections in the app that is small relative to the number of iterations typically used in practice, the proposal standard deviation used is an important choice for the successful elicitation of a prior that represents the expert's beliefs. We found proposal standard deviations of $\sigma_{p} = 0.05$, $\sigma_{\lambda} = \sqrt{\lambda}$, and $\sigma_{\mu} = \sqrt{\sigma}$ to be reasonable for Bernoulli (p), Poisson (λ), and $\mathcal{N}\left(\mu,\sigma^{2}\right)$ with σ^{2} known data models, respectively. These standard deviations allow for efficient exploration of the parameter space when utilizing rigid Metropolis as well as in the graphical elicitation procedures, while being sufficiently large that they enable the expert to distinguish between plots in the Shiny app and reduce the need for thinning, maximizing the information gained from every selection.

6.3 Results

We now assess the accuracy of the rigid Metropolis algorithm when used to draw samples from the posterior distribution for three data models in this article: Bernoulli(p), Poisson(λ), and $\mathcal{N}\left(\mu,\sigma^{2}\right)$ with σ^{2} known and priors Beta(α,β), Gamma(α,β), and $\mathcal{N}\left(\mu_{0},\sigma_{0}^{2}\right)$. Since each of these priors is conjugate, the target posterior distribution is known exactly. For each model we computed the average total variation distance between $p(\theta|\mathbf{y})$ and $\hat{p}(\theta|\mathbf{y})$ for various distributions: Bernoulli

Table 1 Total variation distance summaries for Bernoulli(p), Poisson(λ), and $\mathcal{N}(\mu, \sigma^2 = 100)$ data models based on $\mathcal{P} = \{10^{-6}, 1\}$, indicating that merely rounding the transition probability in the Metropolis algorithm yields an average error on the order of 10%

Data Model	θ^*	Mean δ	SD of δ	95% CI for δ
Bernoulli(p)	0.1	0.1213	0.0083	0.105–0.138
	0.2	0.1138	0.0076	0.100–0.128
	0.3	0.1094	0.0074	0.096–0.123
	0.4	0.1055	0.0081	0.090–0.122
	0.5	0.1060	0.0073	0.091–0.119
	0.6	0.1064	0.0074	0.091–0.121
	0.7	0.1092	0.0081	0.093–0.124
	0.8	0.1134	0.0080	0.098–0.129
	0.9	0.1202	0.0082	0.104–0.137
Poisson(λ)	1	0.1243	0.0146	0.096–0.151
	5	0.1270	0.0158	0.096–0.159
	10	0.1268	0.0186	0.092–0.158
	25	0.1266	0.0164	0.097–0.159
	50	0.1261	0.0163	0.092–0.157
	100	0.1246	0.0158	0.094–0.158
$\mathcal{N}(\mu, \sigma^2 = 100)$	0	0.1245	0.0093	0.107–0.143
	10	0.1249	0.0098	0.107–0.145
	25	0.1242	0.0085	0.109–0.143
	50	0.1240	0.0094	0.104–0.143
	100	0.1246	0.0092	0.104–0.141
	500	0.1238	0.0091	0.105–0.140

with probabilities of 0.1 through 0.9, in increments of 0.1; Poisson with rates of 1, 5, 10, 25, 50, 100; and $\mathcal{N}(\mu, \sigma^2)$ with means of 0, 10, 25, 50, 100, and 500, all with a variance of 100.[2] When doing so, we assumed relatively uninformative priors: Beta(1, 1), Gamma(1, 1), and $\mathcal{N}(0, 10^6)$, respectively. Additionally, for each scenario above, we used acceptance probability vectors of lengths $r = 2$, 3, and 4. Further, for each case, we used $T = 250$ total iterations of $n = 100$ observations, each with $M =$ 5,000 MCMC iterations.

We first examine the case where $r = 2$, for which the resulting vector of rigid MCMC transition probabilities is $\mathcal{P} = \{10^{-6}, 1\}$. Summaries of the resulting total variation distances between $p(\theta|\boldsymbol{y})$ and $\hat{p}(\theta|\boldsymbol{y})$ for all cases considered can be found in Table 1, with 95% equal-tailed credible intervals displayed in the last column. Although the summaries of the total variation distances are reasonably small, we consider scenarios with $r = 3$ transition probabilities with the hope of obtaining smaller values.

For the case where $r = 3$, $\mathcal{P} = \{10^{-6}, p_2^{\text{rigid}}, 1\}$. For each distribution and value of θ^*, plots were made that illustrated how the average TVD varied over the range

[2] Additional variances were considered, and the conclusions discussed in the following sections (as well as the proposal standard deviation used) are robust across reasonable ranges of σ^2.

of the new transition probability $p_2^{\text{rigid}} \in \{0.02, ..., 0.98\}$. In every case, these illustrated the same general pattern: a smooth, near parabolic curve with a clear global minimum. More, these minima are constant across the different values of θ^* used. These minima are taken to be p_2^{rigid} in their respective cases. Summaries of the total variation distances for $r = 3$ are presented in Table 2, indicating substantial improvements from rigid sets of size 2: adding a third probability significantly improved performance.

For the case where $r = 4$, $\mathcal{P} = \left\{10^{-6}, p_2^{\text{rigid}}, p_3^{\text{rigid}}, 1\right\}$, a similar picture was observed: regardless of θ^*, the same or substantially similar values of p_2^{rigid} and p_3^{rigid} were found to be optimal, and when these are used we find another significant reduction in the average TVD, to around 2 to 3%. This is illustrated in Table 3.

Obviously, as r increases, the results of the rigid Metropolis algorithm must converge to that of standard Metropolis. Seen another way: computer

Table 2 Total variation distance summaries for Bernoulli(p), Poisson(λ), and $\mathcal{N}\left(\mu, \sigma^2 = 100\right)$ data models based on $\mathcal{P} = \left\{10^{-6}, p_{2\star}^{\text{rigid}}, 1\right\}$, indicating that three rigid probabilities improve on two (simple rounding), leaving an average error of about 4–5%

Data Model	θ^*	$p_{2\star}^{\text{rigid}}$	Mean δ	SD of δ	95% CI for δ
Bernoulli(p)	0.1	0.28	0.0450	0.0094	0.027–0.063
	0.2	0.28	0.0389	0.0098	0.021–0.058
	0.3	0.30	0.0375	0.0107	0.017–0.057
	0.4	0.32	0.0357	0.0098	0.017–0.053
	0.5	0.34	0.0351	0.0101	0.016–0.054
	0.6	0.36	0.0359	0.0102	0.017–0.053
	0.7	0.36	0.0371	0.0097	0.018–0.055
	0.8	0.32	0.0390	0.0097	0.019–0.055
	0.9	0.28	0.0449	0.0095	0.026–0.061
Poisson(λ)	1	0.26	0.0476	0.0163	0.014–0.077
	5	0.24	0.0480	0.0172	0.014–0.078
	10	0.26	0.0493	0.0175	0.017–0.086
	25	0.26	0.0475	0.0167	0.014–0.079
	50	0.24	0.0466	0.0185	0.014–0.083
	100	0.20	0.0479	0.0197	0.013–0.085
$\mathcal{N}(\mu, 100)$	0	0.26	0.0455	0.0104	0.024–0.064
	10	0.32	0.0457	0.0101	0.025–0.065
	25	0.28	0.0444	0.0104	0.025–0.065
	50	0.28	0.0455	0.0103	0.025–0.063
	100	0.26	0.0453	0.0112	0.024–0.068
	500	0.24	0.0459	0.0111	0.024–0.069

Table 3 Total variation distance summaries for Bernoulli(p), Poisson(λ), and $\mathcal{N}\left(\mu, \sigma^2 = 100\right)$ data models based on $\mathcal{P} = \left\{10^{-6}, 0.04, 0.34, 1\right\}$, indicating that four rigid probabilities improve on three, leaving an average error around 3%

Data Model	θ^*	Mean δ	SD of δ	95% CI for δ
Bernoulli(p)	0.1	0.0341	0.0099	0.014–0.052
	0.2	0.0284	0.0103	0.009–0.047
	0.3	0.0286	0.0110	0.006–0.046
	0.4	0.0286	0.0108	0.007–0.048
	0.5	0.0291	0.0114	0.007–0.050
	0.6	0.0281	0.0106	0.007–0.048
	0.7	0.0285	0.0110	0.006–0.051
	0.8	0.0290	0.0098	0.010–0.048
	0.9	0.0341	0.0101	0.014–0.052
Poisson(λ)	1	0.0344	0.0156	0.008–0.065
	5	0.0348	0.0163	0.006–0.072
	10	0.0332	0.0181	0.006–0.072
	25	0.0356	0.0155	0.007–0.067
	50	0.0362	0.0164	0.006–0.069
	100	0.0334	0.0159	0.008–0.066
$\mathcal{N}(\mu, 100)$	0	0.0294	0.0101	0.010–0.048
	10	0.0289	0.0104	0.007–0.051
	25	0.0304	0.0106	0.011–0.052
	50	0.0284	0.0098	0.010–0.045
	100	0.0292	0.0098	0.010–0.048
	500	0.0300	0.0107	0.006–0.049

implementations of the Metropolis algorithm are simply rigid Metropolis with a very high r. Having attained a desirably low average TVD, we conclude our simulations with $r = 4$. For the three uniparameter data models considered in this article, we found the rigid set $\mathcal{P} = \left\{10^{-6}, 0.04, 0.34, 1\right\}$ strikes a suitable balance for both the rigid procedure and the Shiny app in achieving the following goals: (1) keeping r at a small value, (2) selecting odds experts can reliably attest to, (3) maintaining the theoretical foundation of Metropolis, and (4) attaining a sufficiently small average total variation distance.

7 Conclusion

While methods and tools have been developed that enable experts to inject their beliefs into a Bayesian analysis through the use of an elicited prior, the methods possess certain drawbacks. To this end, we have proposed an interactive graphical procedure for prior elicitation that promotes best practice by allowing experts to work with data rather than parameters, as we believe experts can more reliably attest to the former than the latter. We also describe an accompanying free Shiny implementation which not only enables experts to use the proposed methods, but also leads them through the full elicitation process in a synergistic way, emphasizing often undervalued stages such as expert training and post-elicitation verification of the elicited prior, rather than focusing solely on the elicitation and fitting stages.

The proposed prior elicitation procedure, which treats an elicited prior as the posterior of an expert's previous experience, is based on rigid Metropolis, a variation on standard Metropolis where only a finite number of transition probabilities are used. To assess the accuracy of the rigid MCMC process in three cases of practical import, we use the total variation metric to measure the distance between the target posterior distribution and that found using a rigid MCMC. We have found the rigid set $\mathcal{P} = \left\{10^{-6}, 0.04, 0.34, 1\right\}$ strikes a sensible balance between simplicity for the expert in the elicitation process in the app and accuracy.

The procedure opens the door to a new line of research for eliciting priors. Additional considerations include developing similar procedures for multivariate data models and theoretically connecting the acceptance probability from the Metropolis algorithm to that used in the proposed procedures.

Author Contributions The authors report they each contributed substantially to this work.

Funding No funding was received to assist with the preparation of this manuscript.

Availability of data, material, and code Links to the Shiny app and its source code can be found in Sect. 5. The code used for the simulations in Sect. 6.3 can be provided.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Both authors agreed with the content and gave explicit consent to submit the manuscript.

References

Blair, S. (2017). Contributions to the theory and practice of prior elicitation in biopharmaceutical research (Unpublished doctoral dissertation). Baylor University.

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D. F., & Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 367*(1906), 4361–4383.

Casement, C. J., & Kahle, D. J. (2018). Graphical prior elicitation in univariate models. *Communications in Statistics-Simulation and Computation, 47*(10), 2906–2924.

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., & Borges, B. (2021). shiny: Web application framework for R. Retrieved from https://CRAN.R-project.org/package=shiny (R package version 1.6.0). Accessed 24 Sept 2021.

Garthwaite, P. H., Kadane, J. B., & O'Hagan, A. (2005). Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association, 100*(470), 680–701.

Jones, G., & Johnson, W. O. (2014). Prior elicitation: interactive spreadsheet graphics with sliders can be fun, and informative. *The American Statistician, 68*(1), 42–51.

Kahle, D., Stamey, J., Natanegara, F., Price, K., & Han, B. (2016). Facilitated prior elicitation with the wolfram CDF. *Biometrics & Biostatistics International Journal, 3*(6), 1–6.

Keren, G. (1991). Calibration and probability judgements: Conceptual and methodological issues. *Acta Psychologica, 77*(3), 217–273.

Lichtenstein, S., & Fischhoff, B. (1977). Do those who know more also know more about how much they know? *Organizational Behavior and Human Performance, 20*(2), 159–183.

Lichtenstein, S., Fischhoff, B., & Phillips, L. D. (1977). Calibration of probabilities. In H. Jungermann & G. De Zeeuw (Eds.), *The state of the art. In Decision making and change in human affairs* (pp. 275–324). Springer.

Morita, S., Thall, P. F., & Müller, P. (2008). Determining the effective sample size of a parametric prior. *Biometrics, 64*(2), 595–602.

Morris, D. E., Oakley, J. E., & Crowe, J. A. (2014). A web-based tool for eliciting probability distributions from experts. *Environmental Modelling & Software, 52*, 1–4.

Oakley, J., & O'Hagan, A. (2010). *SHELF: The Sheffield elicitation framework (version 2.0)*. School of Mathematics and Statistics, University of Sheffield.

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., & Rakow, T. (2006). *Uncertain judgements: Eliciting experts' probabilities*. Wiley.

Oskamp, S. (1965). Overconfidence in case-study judgments. *Journal of Consulting Psychology, 29*(3), 261.

R Core Team. (2021). R: A language and environment for statistical computing. Vienna, Austria. Retrieved from https://www.R-project.org/. Accessed 24 Sept 2021.

Resnick, S. I. (2013). *A probability path*. Springer Science & Business Media.

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods* (2nd ed.). Springer.

Ruckdeschel, P., Kohl, M., Stabla, T., & Camphausen, F. (2006). distrEx: S4 classes for distributions (Vol. 6) (No. 2).

Su, C.-L. (2006). BetaBuster. Retrieved from https://cadms.vetmed.ucdavis.edu/diagnostic/software (Center for Animal Disease Modeling and Surveillance, University of California, Davis). Accessed 24 Sept 2021.

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2010). Graphical inference for infovis. *IEEE Transactions on Visualization and Computer Graphics, 16*(6), 973–979.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.