Effective Sample Size Using RBesT

Methodology and Code Walkthrough

James Normington

August 27, 2019

In this document, I outline the underlying methodology in the "effective sample size" functionality of Sebastian Weber's RBesT package.

Throughout, we assume a likelihood for the current trial data y^*: $y^* \sim f(y^*|\theta^*)$, where θ^* is the treatment effect of interest and $f()$ is the binomial, Gaussian, or Poisson probability (density function depending on the endpoint type). We'll further assume the historical trial data y_h follows the hierarchical model

$$\begin{aligned} &y_h|\theta_h \sim f(y_h|\theta_h) \\ &g(\theta_h|\beta) = \beta + \epsilon_h, \epsilon_h \sim N(0, \tau^2), \text{where} \end{aligned} \tag{1}$$

y_h is the summary statistic (sample proportion or sample mean) yielded by the control set in trial h, θ_h is the treatment effect in trial h, $g()$ is a link function, β is the common mean of $g(\theta_1), ..., g(\theta_H)$, and ϵ_h is a normally distributed mean 0 error with variance τ^2. The prior distribution for β is Gaussian, with a default mean of 0 and a standard deviation specified by the user. The prior distribution for τ is specified by the user.

Alternatively, if one wishes trial h to ignore historical information, a non-informative prior independent of $\theta_1, ..., \theta_{h-1}, \theta_{h+1}, ..., \theta_H$ could be specified:

$$\begin{aligned} &y_h|\theta_h \sim f(y_h|\theta_h) \\ &g(\theta_h) \propto 1 \end{aligned} \tag{2}$$

In RBesT, the forest plot functionality fits both hierarchical models; (1) is labeled "meta" and (2) is labeled "stratified".

Below, I list the steps necessary to incorporate historical trial data in estimating the probability of success at interim, at each step elaborating on the underlying mathematics while completing an example in R. The example uses historical meta-data from placebo arms in relevant studies for the treatment of Crohn's disease. The primary outcome y here is the change from baseline in Crohn's Disease Activity Index over 6 weeks, and is assumed to be normally distributed. We assume an identity link; that is, $g(\theta_h) = \theta_h$. The standard deviation of y is estimated to be 88. Suppose the hypothesis set is

$H_0 : \theta^* = -50$

$H_1 : \theta^* < -50$

Some preliminaries in R:

```
#install.packages("RBesT")
library(RBesT)
options(RBesT.MC.control=list(adapt_delta=0.999))
crohn$y.se = 88 / sqrt(crohn$n) # "y" is a column of means; this line computes SE
```

1 Compute MAP prior.

The MAP prior $\pi(\theta^*|y_1, ..., y_h)$ is obtained through MCMC sampling using the hierarchical model in (1). In R,

```
set.seed(2019)
base.MAP.mc = gMAP(cbind(y, y.se) ~ 1| study, family = gaussian,
                   data = crohn, weights = n, beta.prior = 100*sd(crohn$y),
                   tau.dist = "HalfNormal", tau.prior = 1)
```

2 Compute parametric approximation of MAP MCMC samples.

The MAP prior as computed by the gMAP() function consists of thousands of posterior draws. To allow for the computation to proceed quickly upon introducing data at interim, one would want to approximate these raw posterior draws with parametric densities that are conjugate to the likelihood.

The software fits $K = 1$ component, $K = 2$ component, $K = 3$ component, and $K = 4$ component mixture distributions, computing each mixture's AIC:

$$AIC_K = p\lambda - 2ln(\hat{L}_K), \text{ where}$$

p is the number of parameters estimated by the model, λ is the penalty parameter and $\hat{L}_K$ is the maximum value of the likelihood where K components are used. By definition, AIC uses $\lambda = 2$ but $\lambda = 6$ is used here to more heavily penalize complex models, favoring mixtures with fewer components.

To obtain $\hat{L}_K$, the software uses an E-M algorithm to approximate the MAP MCMC samples as the mixture of K parametric distributions:

$$log(\pi(\theta^*|y_1, ..., y_h, a, b)) = \sum_{h=1}^{H} log\left[\sum_{k=1}^{K} w_k \pi_k(\theta_h|a_k, b_k)\right], \text{ where}$$

$k = 1, ..., K$ indexes the mixture components, w_k is the relative weight of mixture component k, and a_k and b_k are the parameters of mixture component k.

The E-M algorithm then extends the MAP prior to

$$log(\pi(\theta^*|y_1, ..., y_h, a, b)) = \int log(\pi(\theta^*, Z|y_1, ..., y_h, a, b))dZ$$

where Z represent the component indicators. The E-M algorithm then iteratively maximizes

$$\mathbb{E}_{Z|\theta^*, \boldsymbol{a}, \boldsymbol{b}}\left[log(\pi(\theta^*|y_1, ..., y_h, a, b, Z))\right]$$

until convergence. The mixture prior with the lowest AIC_K is then selected to represent the MAP MCMC samples.

This is all done in R by simply running

```
base.MAP = automixfit(base.MAP.mc)
```

3 Robustify the mixture prior by adding non-informative component.

This step adds a non-informative component with a user-specified weight to the mixture prior in case of data-prior disagreement. In the "Binary" endpoint case, this is a $U(0,1)$ prior. In the "Normal" endpoint case, this is a diffuse Gaussian prior with a user-specified mean. In the "Poisson" endpoint case, this is a diffuse Gamma prior. In the latter two cases, the default mean is the mean of the current prior mixture, so the user is *strongly* recommended to specify a mean (the null hypothesized mean would make the most sense). In R,

```
MAP.robust = robustify(base.MAP, weight = 0.2, mean = -50)
```

4 Compute ESS

There are many approaches to compute the effective sample size. Here, I outline the three approaches offered in RBesT.

4.1 ELIR

The recommended approach to compute ESS is the *expected local information ratio:*

$$ESS_{ELIR} = \mathbb{E}_{\theta^*}\left[\frac{i(\pi(\theta^*|y_1,...,y_H))}{\mathbb{E}_{y_1^*|\theta^*} i(f(y_1|\theta^*))}\right],$$

where $i(f(x)) = -\frac{d^2 log f(x)}{dx^2}$. Intuitively, ESS_{ELIR} is the ratio of prior precision to likelihood precision. In our context, this means if the MAP prior is very precise relative to the likelihood, we effectively have many patients of information. If the MAP prior is imprecise relative to the likelihood, we don't!

ELIR is the preferred ESS approach because it has *predictive consistency*: for a sample size N, the expected posterior ESS is the sum of the prior ESS and N. This again matches our intuition; for example, if I have $ESS = 50$, this should mean my historical data affords me to enroll 50 fewer patients.

In R,

```
ess(base.MAP, "elir") # ESS of MAP without robust component
ess(MAP.robust, "elir") # ESS of MAP with robust component
```

4.2 Moment Matching

Another approach to computing ESS is by matching the moments of the mixture representation of the MAP prior. The mean and standard deviation of the mixture distribution are computed, then approximated by a conjugate distribution, in which ESS is well defined. The author of this document believes the Moment method is too conservative; that is, it usually underestimates the information in the historical data.

In R,

```
ess(base.MAP, "moment") # ESS of MAP without robust component
ess(MAP.robust, "moment") # ESS of MAP with robust component
```

4.3 Morita-Thall-Müller

A final, more information-based approach to computing ESS is proposed by Morita, Thall, and Müller:

$$ESS_{MTM} = \text{argmin}_{m\in\mathbb{Z}} |i(\pi_0(\bar{\theta})) + \mathbb{E}_{y_m}\{\mathbb{E}_{y_1|\bar{\theta}}[y_m|\bar{\theta}]\} - i(\pi(\bar{\theta}))|$$

where $\pi_0(\theta^*)$ is a high-variance prior with the same mean $\bar{\theta}$ as the MAP prior $\pi(\theta^*)$. Here, the expectation of y_m is taken over the prior predictive distribution under $\pi(\theta^*)$. The author of this document believes the MTM method is not conservative enough; that is, it usually overestimates the information in the historical data. In R,

```
ess(base.MAP, "morita") # ESS of MAP without robust component
ess(MAP.robust, "morita") # ESS of MAP with robust component
```