COMPARISON ON DIFFERENT METHODS FOR NUMERICAL CALCULATION OF POSTERIOR MODEL PROBABILITY MID-TERM REPORT OF STAT6011

LIU ZHIZHOU, UID: 3036212817, DEGREE: MSTAT

Abstract. In this report, we would demonstrate five different methods of calculating posterior model probability. Three of them are based on prior samples; one of them is based on posterior samples; and the remaining one is calculated through direct calculation via Python package.

Contents

1. Setup

Let p_j denote the toxicity probability of dose level j, $j = 1, \ldots, J$, and the toxicity probability is assumed to increase monotonically with the dose level, i.e. $0 < p1 < \cdots < p_J < 1$. In a clinical trial with five dose levels, i.e. $J = 5$, the number of observed toxicity y_j and the number of treated patients n_j are

Dose Level j	1	2	3	4	5
y_j	0	1	4	3	0
n_j	3	6	12	6	0

Date: Nov. 11, 2023.

We consider three models (model M1 is the true model as it satisfies the monotonic constraint):

$$M_1 : p_1 < p_2 < p_3 < p_4 < p_5$$
$$M_2 : p_1 > p_2 > p_3 > p_4 > p_5$$
$$M_3 : p_1 < p_2 < p_3 > p_4 > p_5$$

The aim is to find the posterior model probability $\Pr(M_k|D)$, where D is the observed data. Assume each levels of trials are independent. Then the likelihood function is

$$L(D|p_1, \ldots, p_J) = \prod_{j=1}^{J} \binom{n_j}{y_j} p_j^{y_j} (1-p_j)^{n_j - y_j}. \tag{1}$$

Then given the observed data D, the marginal likelihood under model M_k ($k = 1, 2, 3$) is

$$\begin{aligned} \Pr(D|M_k) &= \int f(p_1, \ldots, p_J|M_k) L(D|p_1, \ldots, p_J) \mathrm{d}p_1 \cdots \mathrm{d}p_J \\ &= \int f(p_1, \ldots, p_J|M_k) \prod_{j=1}^{J} \binom{n_j}{y_j} p_j^{y_j} (1-p_j)^{n_j - y_j} \mathrm{d}p_1 \cdots \mathrm{d}p_J \end{aligned} \tag{2}$$

where $f(p_1, \ldots, p_J|M_k)$ is the joint prior distribution of $p_1, \ldots, p_J$ under M_k. The posterior probability of M_k is then given by

$$\Pr(M_k|D) = \frac{\Pr(D|M_k)\Pr(M_k)}{\sum_{k=1}^{K} \Pr(D|M_k)\Pr(M_k)}, \tag{3}$$

where $\Pr(M_k)$ is the prior probability of M_k.

Now we specify a discrete uniform distribution for the prior model probability; that is $\Pr(M_k) = 1/3$, $k = 1, \ldots, 3$. We assign the joint prior distribution $f(p_1, \ldots, p_5|M_k)$ to be multivariate uniform but with restriction on the domain such that $(p_1, \ldots, p_5)$ must satisfy the order constraint under each model M_k.

The keys for calculations of (3) are on (2).

2. Methods Based on Prior Samples

The idea of this section is that, once the prior samples are generated, then we could obtain (2) by Monte Carlo estimation

$$\Pr(D|M_k) \approx \frac{1}{I} \sum_{i=1}^{I} \prod_{j=1}^{J} \binom{n_j}{y_j} (p_j^{(i)})^{y_j} (1 - p_j^{(i)})^{n_j - y_j}.$$

There are three *possible* approaches to generate joint uniform prior of $(p_1, \ldots, p_5)$ under the model constraint.

- Sequential: generate $p_1, \ldots, p_5$ from the joint uniform prior in a sequential order, e.g. under M_3, simulate $p_1, \ldots, p_5$ as $p_1 \sim \text{Uni}(0, 1)$, $p_2 \sim \text{Uni}(p_1, 1)$, $p_3 \sim \text{Uni}(p_2, 1)$, $p_4 \sim \text{Uni}(0, p3)$ and $p_5 \sim \text{Uni}(0, p_4)$.
- Reorder: Generate $p_1, \ldots, p_5$ from the joint uniform prior by first simulating five uniform $(0, 1)$ variates, and then reordering them according to each model constraint. Note that model M_3 does not fully specify the order so that we seperate the $p_1 < p_2 < p_3 > p_4 > p_5$ into six equal-probability order-specified subcases:
  - (i) $p_3 > p_2 > p_1 > p_4 > p_5$,
  - (ii) $p_3 > p_2 > p_4 > p_1 > p_5$,
  - (iii) $p_3 > p_2 > p_4 > p_5 > p_1$,

(iv) $p_3 > p_4 > p_2 > p_1 > p_5$,
(v) $p_3 > p_4 > p_2 > p_5 > p_1$,
(vi) $p_3 > p_4 > p_5 > p_2 > p_1$,
to overcome.

- Gibbs: the full conditional distribution of joint (truncated [1]) uniform distribution can be calculated as

$$f(p_i|p_1,\dots,p_{i-1},p_{i+1},\dots,p_5) = \frac{f(p_1,\dots,p_5)}{f(p_1,\dots,p_{i-1},p_{i+1},\dots,p_5)}.$$

In particular, for each model, when $k=1$,

$$f(p_i|p_1,\dots,p_{i-1},p_{i+1},\dots,p_5) = \frac{1}{p_{i+1}-p_{i-1}}, \quad i=1,\dots,5$$

where we set $p_0 = 0, p_6 = 1$. For $k=2$,

$$f(p_i|p_1,\dots,p_{i-1},p_{i+1},\dots,p_5) = \frac{1}{p_{i-1}-p_{i+1}}, \quad i=1,\dots,5$$

where we set $p_0 = 1, p_6 = 0$. For $k=3$,

$$f(p_i|p_1,\dots,p_{i-1},p_{i+1},\dots,p_5) = \begin{cases} \frac{1}{p_{i+1}-p_{i-1}}, & i=1,2 \\ \frac{1}{1-\max\{p2,p4\}}, & i=3 \\ \frac{1}{p_{i-1}-p_{i+1}}, & i=4,5 \end{cases},$$

where we set $p_0 = 0, p_6 = 0$.

An alternative but easier method is that because of

$$f(p_i|p_1,\dots,p_{i-1},p_{i+1},\dots,p_5) \propto f(p_1,\dots,p_5) = \text{constant}$$

$p_i|p_1,\dots,p_{i-1},p_{i+1},\dots,p_5$ would follows a corresponding uniform distribution.

Therefore, we may generate the Gibbs samples easily by Python function `scipy.stats.uniform()`.

2.1. **Theoretical Comments.**

2.1.1. *On Sequential Method.*

In fact, the sequential method would not produce a joint uniform distribution. Take M_1 as an exmaple,

$$\begin{aligned} f(p_1,\dots,p_5) =& f(p_5|p_1,\dots,p_4)f(p_4|p_1,p_2,p_3)f(p_3|p_1,p_2)f(p_2|p_1)f(p_1) \\ =& \frac{1}{(1-p_4)(1-p_3)(1-p_2)(1-p_1)}. \end{aligned}$$

Although the support of density is indeed $p_1 < \dots < p_5$, the value is not 1. Therefore we conclude that the sequential method is incorrect theoretically.

[1] In the following, the "truncated" distribution means a distribution following a known distribution but with domain truncated by the corresponding model.

Methods	$-\log \Pr(D\|M_1)$	$-\log \Pr(D\|M_2)$	$-\log \Pr(D\|M_3)$	$\Pr(M_1\|D)$	$\Pr(M_2\|D)$	$\Pr(M_3\|D)$
Sequential	8.30	10.37	8.65	0.55	0.07	0.38
Reorder	5.24	10.16	8.02	0.93	0.01	0.06
Gibbs	5.13	11.11	9.05	0.98	0.00	0.01

Table 1. Summerized results of estimations based on prior samples. As the values of $\Pr(D|M_k)$s are very small, we present the $-\log$-value here. Results are rounded to two decimal places.

2.1.2. *On Reorder Method.* The technique of reordering is actually producing order statistics. Take M_1 as an example, we actually have $p_1 = X_{(1)}, \dots, p_5 = X_{(5)}$ where $X_1, \dots, X_5 \sim \text{Uni}(0,1)$ independently and $X_{(i)}$ is the i-th order statistics. Then by formula of the joint distribution of the order statistics of an absolutely continuous distribution, or directly stated in Wikipedia, we have

$$f(p_1, \dots, p_5) = 5! = 120, \quad 0 < p_1 < \cdots < p_5 < 1,$$

which is indeed the truncated uniform distribution.

2.1.3. *On Gibbs Sampler.* The Gibbs sampler would converge to the desired truncated uniform distribution by ergodic theorem, as illustrated in Lecture 3.

2.2. **Implementation in Python.** For codes, see `generatePriorSample()` in "`modelposterior.py`". Note that the samples of truncated unifrom distribution can be generated through a rejection process. The generated prior samples are visualized in Figure 1. These figues are plotted by `pltAllThree()` in "`modelposterior.py`".

2.3. **Results.** Presented in Table 1.

3. Method Based on Posterior Samples

Again, we aim to find $\Pr(D|M_k)$ for $k = 1, 2, 3$. We shall use Chib's method [1] to utlize posterior samples for the calculations of $\Pr(D|M_k)$. The idea is as follows. By Bayes' rule,

$$\Pr(D|M_k) = \frac{L(D|\mathbf{p}^*, M_k) f(\mathbf{p}^*|M_k)}{f(\mathbf{p}^*|D, M_k)}, \tag{4}$$

where $\mathbf{p}^* = (p_1^*, \dots, p_5^*)$ is an arbitrary parameter of the corresponding model M_k. As the likelihood function and prior distribution are both known, it remains to find the posterior $f(\mathbf{p}^*|D, M_k)$. Typically, $\mathbf{p}^*$ would be chosen as a point of high posterior probability to increase the numerical accuracy of the estimate. It is clear that

$$f(\mathbf{p}^*|D, M_k) = \prod_{j=1}^{n} f(p_j^*|p_{j-1}^*, \dots, p_1^*, D, M_k).$$

And each factor can be estimated from the Gibbs output of posterior by integrating out parameters $p_{j+1}^{(i)}, \dots, p_n^{(i)}$,

$$f(p_j^*|p_{j-1}^*, \dots, p_1^*, D, M_k) = \frac{1}{I} \sum_{i=1}^{I} f(p_j^*|p_5^{(i)}, \dots, p_{j+1}^{(i)}, p_{j-1}^*, \dots, p_1^*, D, M_k). \tag{5}$$

Therefore, it remains to generate the Gibbs samples of posterior and find the formula of full conditional distribution.

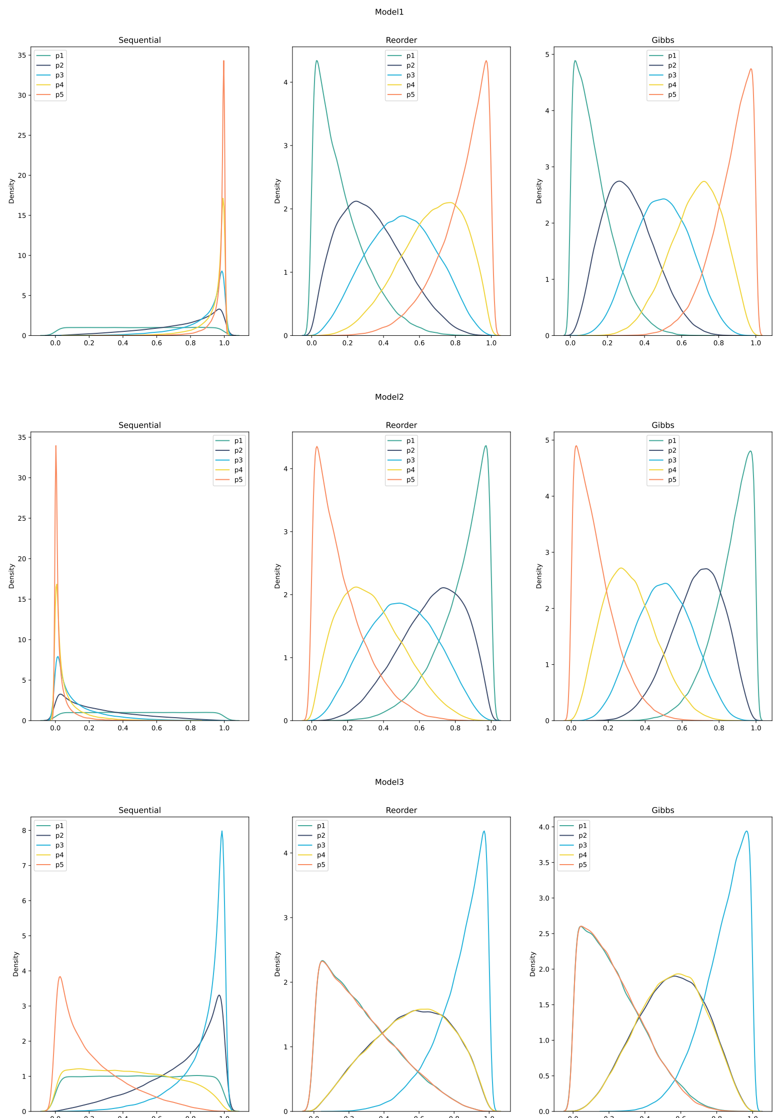

FIGURE 1. Margianl density plot of prior distribution $f(p_1, \ldots, p_5|M_k)$ generated by three different methods of model M_k, $k = 1, 2, 3$. Note that the prior samples generated by Sequential methods are totally different from the other two methods. See subsection 2.1 for theoretical illustrations.

3.1. **Full Conditional Distribution of Posterior.** It is clear that the posterior is proportional to the product of likelihood (1) and prior (constant), so

$$f(p_1, \ldots, p_5|D, M_k) \propto \prod_{j=1}^{5} p_j^{y_j}(1-p_j)^{n_j - y_j}. \tag{6}$$

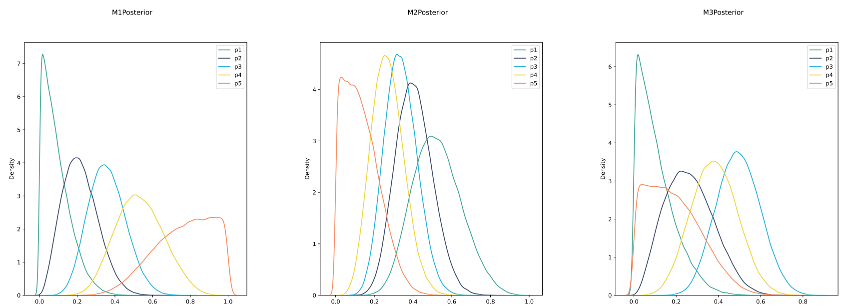

FIGURE 2. Margianl density plot of posterior distributions $f(p_1, \ldots, p_5|D, M_k)$ generated by Gibbs sampler for each $k = 1, 2, 3$.

From the expression, we could deduce that $p_1, \ldots, p_5|D, M_k$ follows a truncated Dirichlet distribution. Then take $k = 1$ as an example,

$$\begin{aligned} &f(p_j|p_1, \ldots, p_{j-1}, p_{j+1}, \ldots, p_5, D, M_k) \\ &\qquad = \frac{f(p_1, \ldots, p_5|D, M_k)}{f(p_1, \ldots, p_{j-1}, p_{j+1}, \ldots, p_5, D, M_k)} \\ &\qquad = \frac{p_j^{y_j}(1-p_j)^{n_j-y_j}}{\int_{p_{j-1}}^{p_{j+1}} p_j^{y_j}(1-p_j)^{n_j-y_j}}, \end{aligned}$$

which can be viewed as a truncated Beta distribution.

An alternative but easier way to see this fact is from the observation

$$f(p_j|p_1, \ldots, p_{j-1}, p_{j+1}, \ldots, p_5, D, M_k) \propto p_j^{y_j}(1-p_j)^{n_j-y_j},$$

so that it must follows a truncated Beta distribution.

Here the intepretation of p_0, p_6 is the same as Gibbs prior sampler in section 2 for each model k.

3.2. **Implementation in Python.** The samples of truncated Beta distribution can be generated through a rejection process. For codes, see `generatePosteriorSample()` in "`modelposterior.py`". Also, generated samples are plotted through `pltPosterior()`, see Figure 2. Then we use the Gibbs samples to find

$$f(p_j^*|p_{j-1}^*, \ldots, p_1^*, D, M_k)$$

through (5), taking $\mathbf{p}^*$ as the point with maximum density. The procedure of finding the point with maximum density is contained in the plotting process `pltPosterior()`. Finally we obtain $f(\mathbf{p}^*|D, M_k)$ by products (6). It remains using (4) to obtain $\Pr(D|M_k)$.

3.3. **Results.** Presented in Table 2.

4. DIRECT CALCULATION BASED ON PYTHON PACKAGE `SYMPY`

The direct calculations of $\Pr(D|M_k)$ through (2) by hands are quite involved multipule integral exercise. Thanks to Python package `Sympy`, they can be solved

$-\log \Pr(D\|M_1)$	$-\log \Pr(D\|M_2)$	$-\log \Pr(D\|M_3)$	$\Pr(M_1\|D)$	$\Pr(M_2\|D)$	$\Pr(M_3\|D)$
5.25	9.95	7.97	0.93	0.01	0.06

TABLE 2. Summerized results of estimations based on posterior samples. As the values of $\Pr(D|M_k)$s are very small, we present the $-\log$-value here. Results are rounded to two decimal places.

$-\log \Pr(D\|M_1)$	$-\log \Pr(D\|M_2)$	$-\log \Pr(D\|M_3)$	$\Pr(M_1\|D)$	$\Pr(M_2\|D)$	$\Pr(M_3\|D)$
5.25	10.15	8.00	0.93	0.01	0.06

TABLE 3. Summerized results of direct calculation based on Python package `Sympy`. As the values of $\Pr(D|M_k)$s are very small, we present the $-\log$-value here. Results are rounded to two decimal places.

by programing. Take $k = 1$ as an example, the integrand is

$$5! \cdot \prod_{j=1}^{5} \binom{n_j}{y_j} p_j^{y_j} (1 - p_j)^{n_j - y_j} 1_{0<p_1<\cdots<p_5<1},$$

where 1_A is the indicator function for some set A. Since the package is not able to solve for integrands containing indicator functions, it is better to write the integral as

$$\int 5! \cdot \prod_{j=1}^{5} \binom{n_j}{y_j} p_j^{y_j} (1 - p_j)^{n_j - y_j} 1_{0<p_1<\cdots<p_5<1} \mathrm{d}p_1 \cdots \mathrm{d}p_5$$
$$= \int_0^1 \mathrm{d}p_5 \int_0^{p_5} \mathrm{d}p_4 \cdots \int_0^{p_2} \mathrm{d}p_1 \prod_{j=1}^{5} \binom{n_j}{y_j} p_j^{y_j} (1 - p_j)^{n_j - y_j}.$$

4.1. **Results.** Presented in Table 3.

5. CAMPARISONS AND CONCLUSIONS

The final results are presented in Table 4. In terms of values $\Pr(M_k|D)$, we conclude that

- for methods based on prior samples, sequential is incorrect, both practically and theoretically.
- reorder is correct, both practically and theoretically.
- Gibbs is correct theoretically, but incorrect in practise. This may due to the reason that sample size is still not big enough, as the ergodic theorem only gives a qualitative conclusion.
- the method based on posterior samples is correct, both practically and theoretically.

See section 2.1 for the theoretical comments on the methods based on prior samples; [1] for the theoretical justifical on the methods based on posterior samples.

```
P(D|M1) using
prior samples generated by:
    - Sequential:  0.0002495301469676799 Take Log:  -8.295930820542221
    - Reorder:  0.005284676195702929 Take Log:  -5.242943930088884
    - Gibbs:  0.005900350497520215 Take Log:  -5.132743523475593
posterior samples generated by Gibbs:  0.005266292920716173 Take Log:
-5.246428594556399
Real value calculated by Sympy.integrate():  0.00526204857061652 Take Log:
-5.2472348659587995

P(D|M2) using
prior samples generated by:
    - Sequential:  3.13264162115432e-05 Take Log:  -10.371048847939349
    - Reorder:  3.883222660131004e-05 Take Log:  -10.156260073569829
    - Gibbs:  1.4957567875769999e-05 Take Log:  -11.110293173783779
posterior samples generated by Gibbs:  4.787782275247257e-05 Take Log:
-9.946858151314434
Real value calculated by Sympy.integrate():  3.92595954034856e-5 Take Log:
-10.14531467473489

P(D|M3) using
prior samples generated by:
    - Sequential:  0.0001746873227341847 Take Log:  -8.652512909374957
    - Reorder:  0.00033011143232042264 Take Log:  -8.01608028680443
    - Gibbs:  0.00011790271590876442 Take Log:  -9.045650714988518
posterior samples generated by Gibbs:  0.0003469696029222537 Take Log:
-7.966273381502172
Real value calculated by Sympy.integrate():  0.000335330507221715 Take Log:
-8.00039392377646

Model Posterior Probability for
prior samples generated by:
    - Sequential: P(M1|D)=0.5477631347578045, P(M2|D)=0.06876706543592573,
    P(M3|D)=0.3834697998062698
    - Reorder: P(M1|D)=0.9347420469701484, P(M2|D)=0.006868559896107142,
    P(M3|D)=0.05838939313374447
    - Gibbs: P(M1|D)=0.9779785111774594, P(M2|D)=0.0024792052553707163,
    P(M3|D)=0.019542283567169957
posterior samples generated by Gibbs: P(M1|D)=0.930253022975053,
P(M2|D)=0.008457275358487752,P(M3|D)=0.061289701666459195
Real value calculated by Sympy.integrate(): P(M1|D)=0.933543708521986,
P(M2|D)=0.00696507221402340, P(M3|D)=0.0594912192639910
```

FIGURE 3. An example output of `modelposterior.py` in Terminal.

Methods	$-\log\Pr(D\|M_1)$	$-\log\Pr(D\|M_2)$	$-\log\Pr(D\|M_3)$	$\Pr(M_1\|D)$	$\Pr(M_2\|D)$	$\Pr(M_3\|D)$
Prior (Sequential)	8.30	10.37	8.65	0.55	0.07	0.38
Prior (Reorder)	5.24	10.16	8.02	0.93	0.01	0.06
Prior (Gibbs)	5.13	11.11	9.05	0.98	0.00	0.01
Posterior	5.25	9.95	7.97	0.93	0.01	0.06
True	5.25	10.15	8.00	0.93	0.01	0.06

TABLE 4. Summerized results of ALL methods, combining Table 1, 2 and 3.

REFERENCES

[1] Siddhartha Chib. Marginal likelihood from the gibbs output. *Journal of the american statistical association*, 90(432):1313–1321, 1995.

Email address: `liuzz95@connect.hku.hk`