SBC on adjoint-differentiated Laplace approximation when prior, likelihood, data are changed

Hyunji Moon
Andrew Gelman LAB Intern
2020.09

This research is based on

Hamiltonian Monte Carlo using an adjoint-differentiated Laplace approximation
- Charles C. Margossian, Aki Vehtari, Daniel Simpson, Raj Agrawal

Validating Bayesian Inference Algorithms with Simulation-Based Calibration
- Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, Andrew Gelman

contents

0. Latent gaussian model, SBC
1. Prior
    a. Parameter
    b. Shape
2. Likelihood
    c. Poisson, Bernoulli
3. Data

Latent gaussian model

$$\eta_i = \alpha + \sum_{j=1}^{n_f} f^{(j)}(u_{ji}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \varepsilon_i$$

goal: structured additive regression models where the latent field is Gaussian, controlled by a few hyperparameters and with non-Gaussian response variables

Latent gaussian model

$$\theta \sim \text{Normal}(0, K(\phi))$$

$$K_{ij} = \alpha^2 \exp\left(-\frac{||x_i - x_j||^2}{\rho^2}\right)$$

$$\pi(y_i \mid \theta_i, \phi) = \text{Normal}(\theta_i, \sigma^2)$$
$$\pi(y_i \mid \theta_i, \phi) = \text{Poisson}\left(\exp \theta_i\right)$$
$$\pi(y_i \mid \theta_i, \phi) = \text{Bernoulli}(\text{logit}^{-1}\theta_i)$$

$$\pi(\theta \mid y, \phi) = \text{Normal}\left(\left(K^{-1} + \frac{n}{\sigma^2} I\right)^{-1} \frac{1}{\sigma^2} y, \left(K^{-1} + \frac{n}{\sigma^2} I\right)^{-1}\right)$$

No closed form for π(y | θ), π(θ | φ , y). SOS, MCMC!

	problem	solution
1	No closed form	MCMC

Latent gaussian model

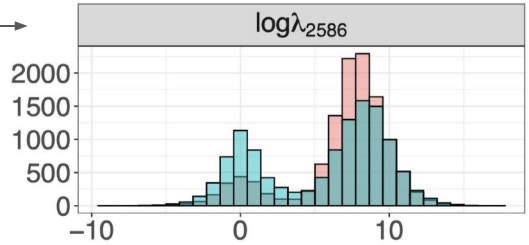
	problem	solution
2	Bad(θ, φ) joint posterior	Divide and conquer (integrate over lat.var)
3	high dimension + multimodal	Charles' idea on using reverse mode + use our model :)

1. run HMC on ϕ , by encoding π(ϕ) and π(y | ϕ) in the model block.
2. sample θ from π(θ | y, ϕ) in generated quantities block.

$$\beta_i \sim \text{Normal}(0, \tau^2 \tilde{\lambda}_i^2) \qquad y \sim \text{Bernoulli}\left(\text{logit}(\beta_0 + \mathrm{X}\beta)\right)$$
$$\theta \sim \text{Normal}(0, K(\alpha, \rho, x)), \qquad y_i \sim \text{Poisson}(y_e^i e^{\theta_i}),$$

$$\pi_{\mathcal{G}}(\phi \mid y) := \pi(\phi) \frac{\pi(\theta^* \mid \phi)\pi(y \mid \theta^*, \phi)}{\pi_{\mathcal{G}}(\theta^* \mid \phi, y)} \approx \pi(\phi \mid y)$$

normal



Can capture bimodality

Data and goal

Disease map $\pi(y_i \mid \theta_i, \phi) = \text{Poisson}\left(\exp \theta_i\right)$

- mortality count, from 100 or 20 location coordinates
- predict mortality or identify high risk locations

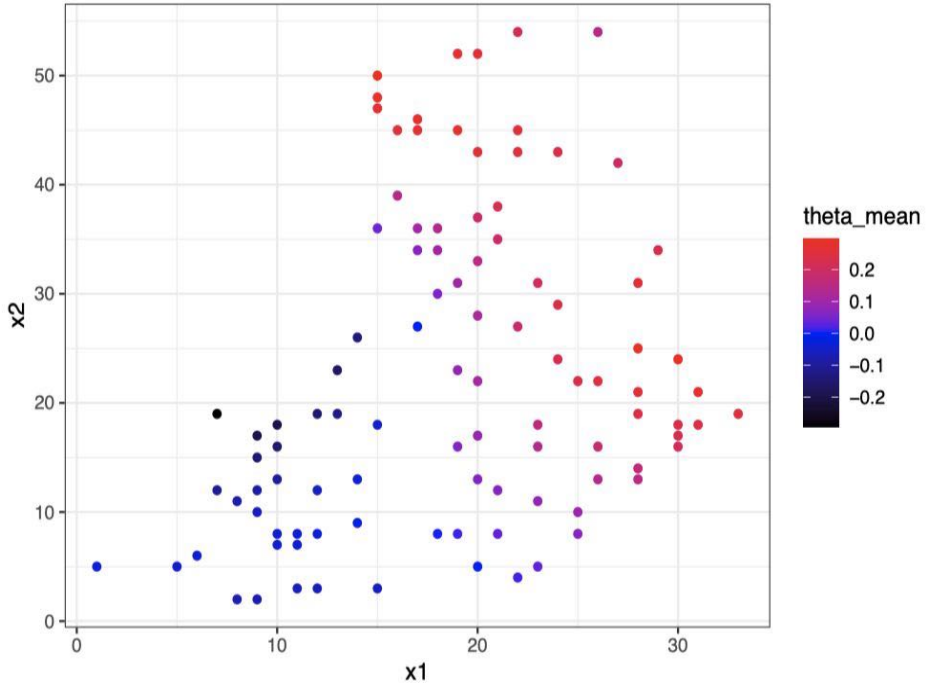
$$\theta = X\beta$$

Prostate cancer $\pi(y_i \mid \theta_i, \phi) = \text{Bernoulli}(\text{logit}^{-1}\theta_i)$

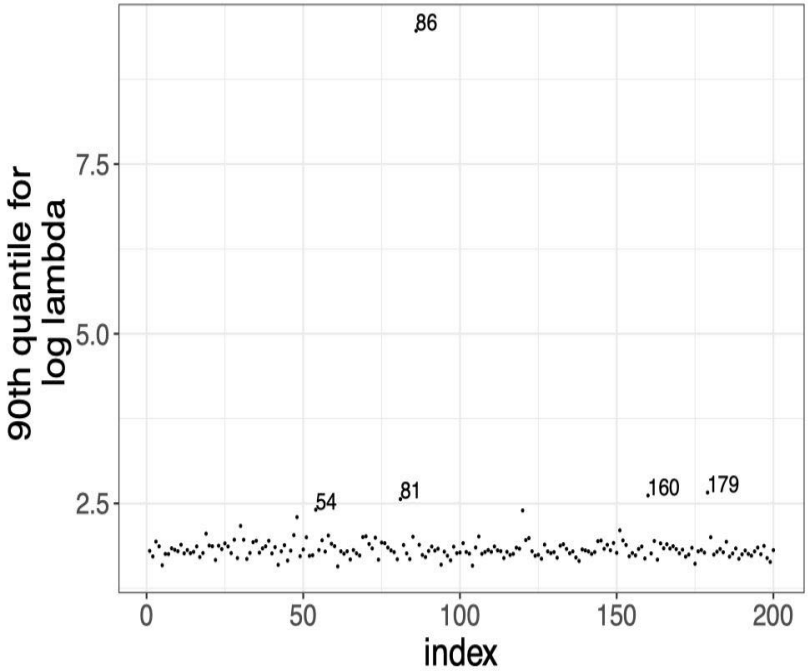
- cancer classification, 200 covariates from 102 patients
- predict probability of developing cancer for each patient or identify risk factors

Latent variable matters!

$$\theta_i = \sum_j x_{ij}\beta_j$$



Identify high risk locations



identify cancer factors

Simulated Based Calibration

1. L posterior samples from 1 fit

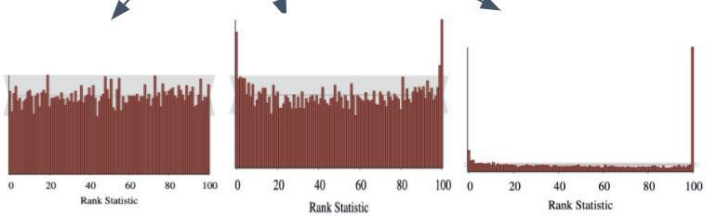
$$\tilde{\theta} \sim \pi_S(\theta)$$
$$\tilde{y} \sim \pi_S(y \mid \tilde{\theta})$$
$$\left(\tilde{\theta}'_1, \ldots, \tilde{\theta}'_L\right) \sim \pi(\theta \mid \tilde{y})$$

$$\rho = \#\left\{\tilde{\theta} < \tilde{\theta}'_l\right\}$$

Rank statistic

2. Repeat N times of fit

$$\rho_1, \ldots, \rho_N$$



uniform if posterior and prior samples are identically distributed

3. Global summary of rank uniformity

E.g. chi square goodness of fit

$$g(\theta) = \chi^2 test - pval$$

frequentist checks to validate Bayesian procedures

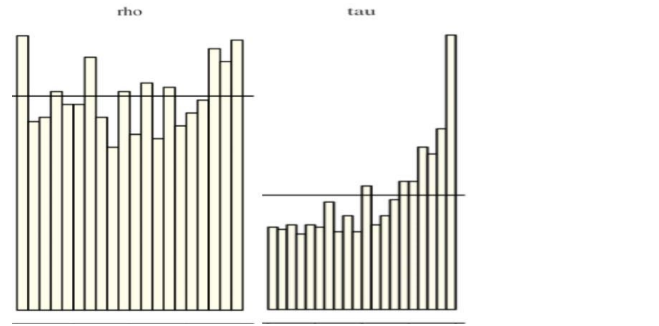
Three types of calibration

	Inf.Calib_env	Inf.Calib_self	Alg.Calib_self
use y_obs (data block)	O	X	X
dgp model (transf. data block)	X	standard model	standard model
fit model (model block)	standard model	standard model	approximate model

= adj-diff.Laplace

I used this one.

Need for different measure




```
chisq.test(c(51, 53, 52, 40, 44, 48, 46, 57, 51, 40, 55, 58, 48, 52, 38, 49, 51, 69, 57, 40))$p.value
] 0.31
chisq.test(c(46, 41, 42, 53,52, 44, 50, 50, 46, 49, 49, 45, 64, 55, 44, 50, 46, 50, 55, 69))$p.value
] 0.51
```

```
MW1 <- function(bin_count){
  bins <- length(bin_count)
  unif <- rep(1/bins, bins)
  M <- sum(bin_count)
  tempf <- Vectorize(function(i)  abs(bin_count[i]/M  - unif[i]))
  val <- integrate(tempf,1,bins, rel.tol=.Machine$double.eps^.05)$value
  return(val)
}
MKM <- function(bin_count){
  bins <- length(bin_count)
  diff <- abs(mean(bin_count) - bin_count)
  val <- diff[which.max(diff)] / mean(bin_count)
  return(val)
}
MChisq <- function(bin_count){
  return(chisq.test(bin_count)$p.value)
}
```

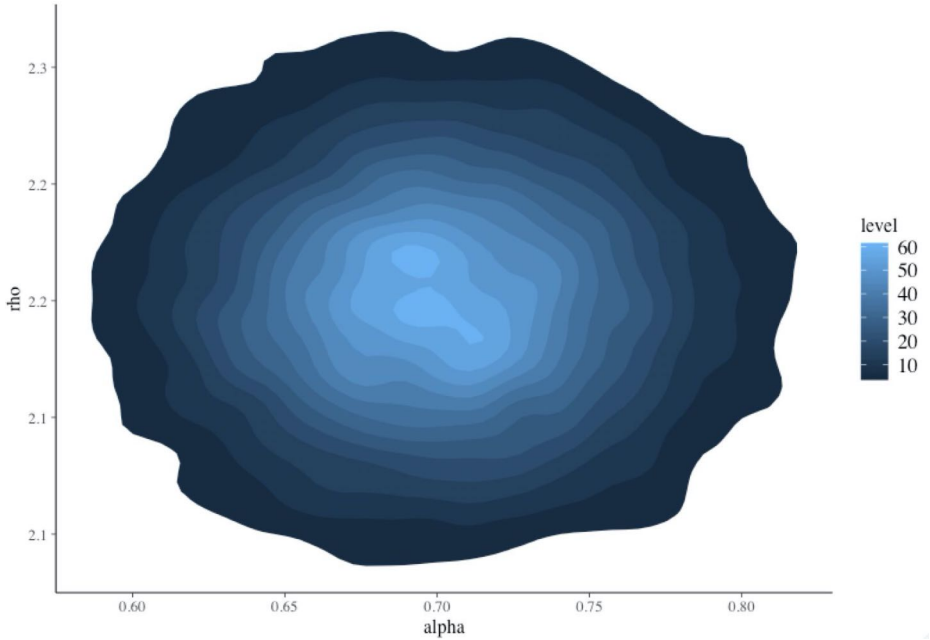
Rho seems less uniform-deviated but has lower pvalue.

Additional uniformity measures:

1. wasserstein distance from uniform distribution
2. Max deviation from uniform

1. Prior

Wild prior search



Based on SBC, our approximate model works well in this parameter (α, ρ) range

Prior range and its density for ideal alg. functionality

Procedure

1. Find the widest prior range validated by SBC
2. Variation in terms of density width(sd, fwhm, scale), tail behavior(df of t, gamma vs inv-gamma) specific focus on the causes of the transition from accurate to biased computation at the prior boundary

Analysis

Simulating data from certain model configurations range is unlikely to result in problematic posterior for approximate algorithm. More extreme model configurations may result in data and realized posterior densities that frustrate the accuracy of our approximate method.

Covariate dimension 20 vs 100

	20 covariates (avg. fitting time)	100 covariates (avg. fitting time)
α~N(0, 0.1) ρ~ N(0, 0.1)	0.87	15.32
α~N(0, 0.5) ρ~ N(0, 0.5)	1.06	21.51
α~N(0, 1) ρ~ N(0, 1)	1.11	X
α~N(0, 10) ρ~ N(0, 10)	X	X

$$\theta \sim \text{Normal}(0, K(\phi))$$

$$K_{ij} = \alpha^2 \exp\left(-\frac{||x_i - x_j||^2}{\rho^2}\right)$$

Average fitting time when half normal prior was given for parameter alpha and rho

More covariate dimension led to smaller range of parameters (without 'initialization failed no chain ended' error)

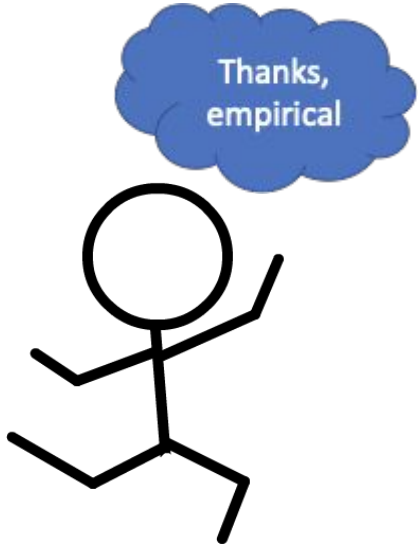
X: 'initialization failed no chain ended'
If 1000fits, .1 took 1.49 (due to extreme fitting time outliers)

Data = 20

Prior parameter: mean

	20 covariates (avg. fitting time)	100 covariates (avg. fitting time)
α~N(0, 1) ρ~ N(0, 1)	1.11	3 out of 3 initialization failed error
α~N(0.7, 0.1) ρ~ N(2.2, 0.1)	1.49	
α~N(0.7, 0.05) ρ~ N(2.2, 0.05)	1.28	

Adjusting the prior mean to empirical Bayes result, extended the scale boundary.

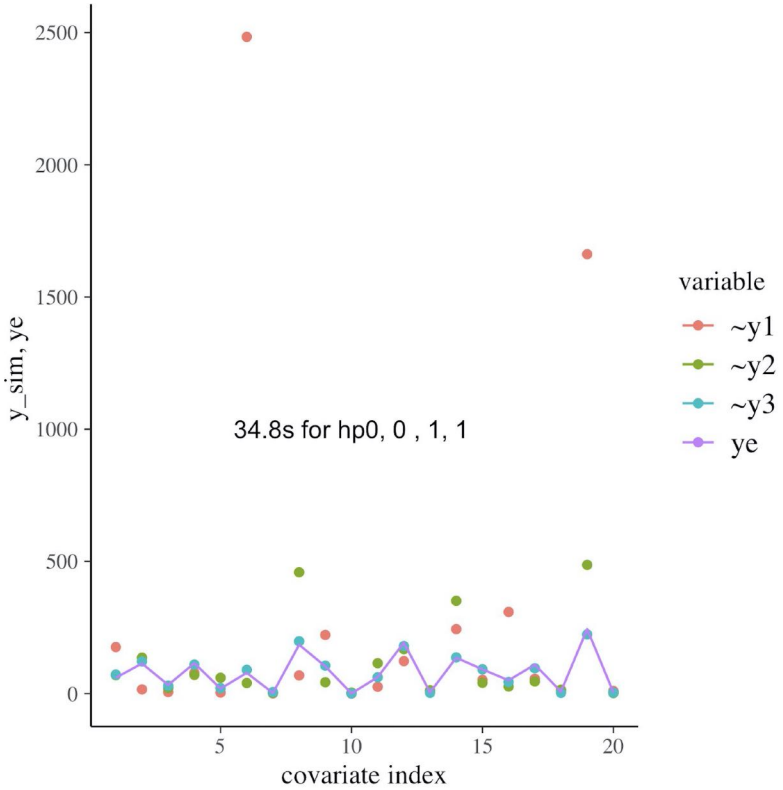


An alternative, but useful view is to understand an approximate algorithm as an *exact* algorithm for an approximate model. In this sense, a workflow is a sequence steps in an abstract computational scheme aiming to infer some ultimate, unstated model. More usefully, we can think of things like empirical Bayes approximations as replacing a model's prior distributions with a particular data-dependent point-mass prior. Similarly a Laplace approximation can be viewed as a data-dependent linearization of the true model, while a nested Laplace approximation (Rue et al., 2009, Margossian et al., 2020) uses a linearized conditional posterior in place of the true conditional posterior.

fitting time for y_sim with different prior mean, sd

	Mean 0	Empirical Mean
α~N(*, over 3) ρ~ N(*, 3)	X (no chain ended successfully)	X
α~N(*, 2) ρ~ N(*, 1)	X	18s
α~N(*, 1) ρ~ N(*, 1)	3.5s	6.3s
α~N(*, 0.1) ρ~ N(*, 0.1)	2s	1.5s
α~N(*, 0.01) ρ~ N(*, 0.01)	0.54s	0.64s

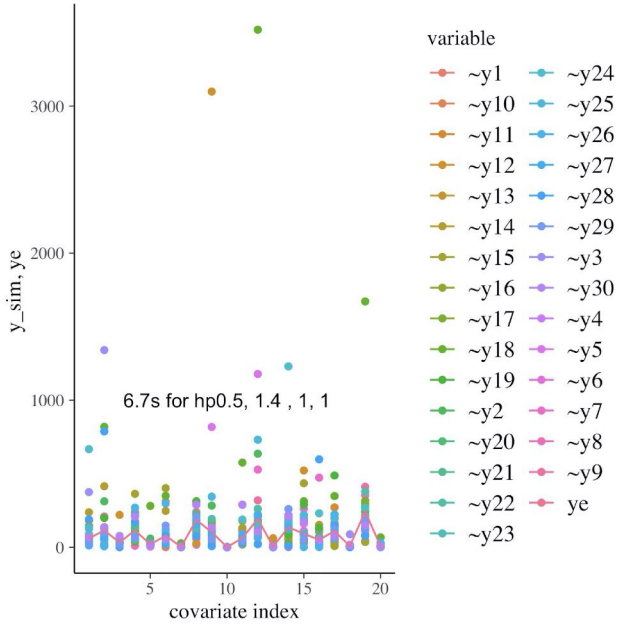
Extreme y_sim takes much longer to fit

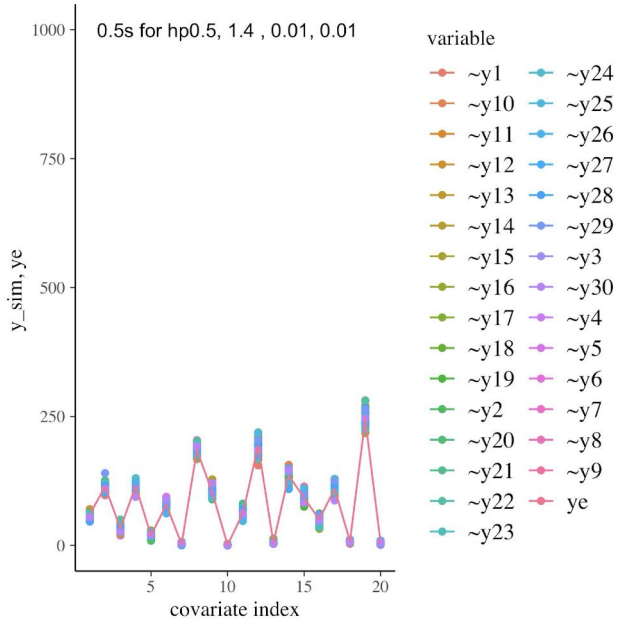


ye represents real data $y_i \sim \text{Poisson}(y_e^i e^{\theta_i})$

~y1 had extreme value(2500) and took 10 times longer to fit

Setting prior sd smaller prevents extreme y

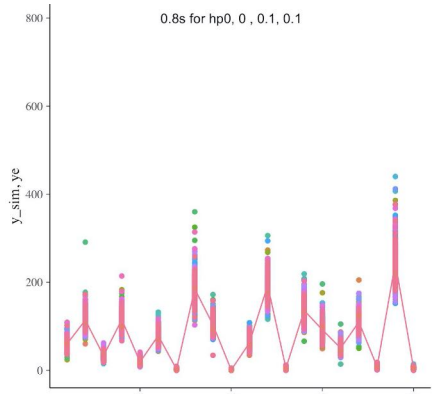


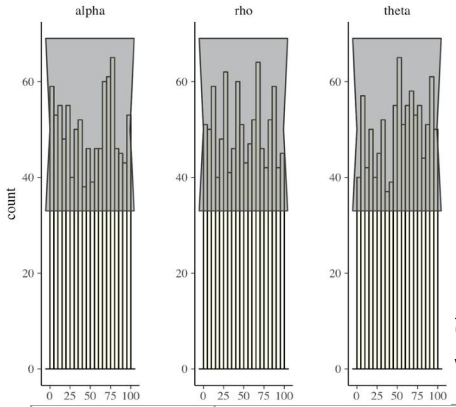


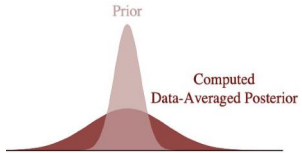
10times faster fitting when sd is 0.01 compare to 1

y_sim of sd1 much nearer to real data

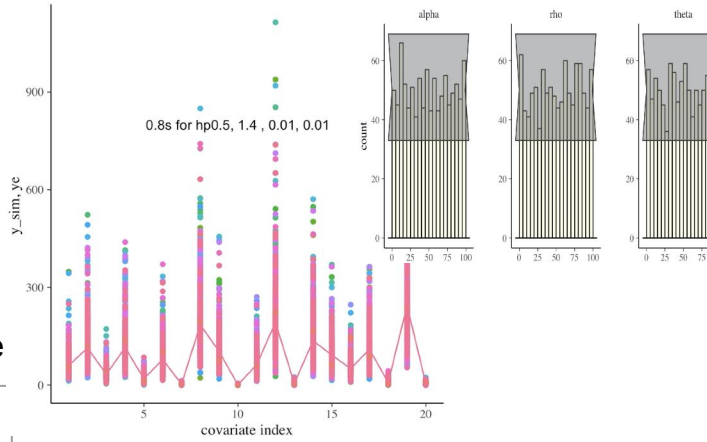
2*2 comparison of y_sim, SBC histogram pvalue for different priors are presented in the next slide





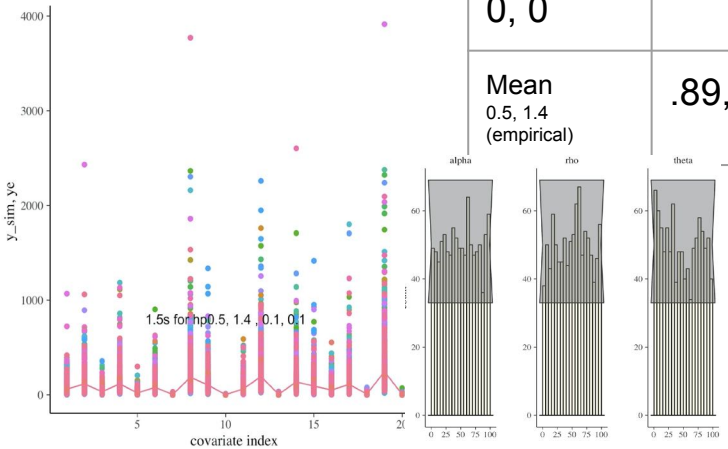


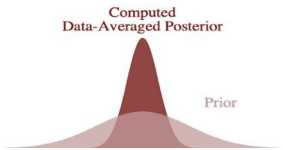
computed posterior
wider than true poste



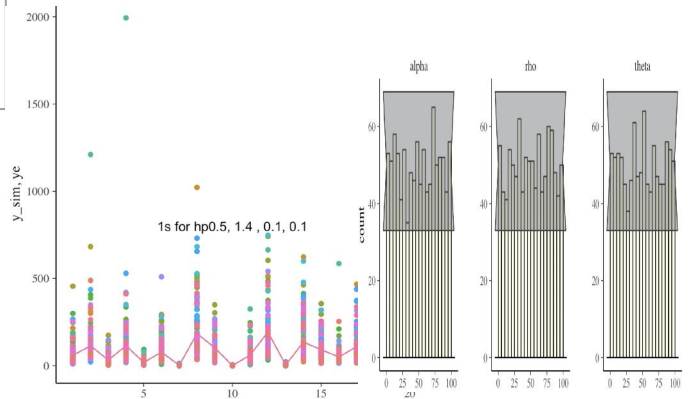
α~N(m, sd) ρ~ N(m, sd)	Sd 0.1	Sd 0.01
mean 0, 0	.27, .34, .22	.59, .46, .62
Mean 0.5, 1.4 (empirical)	.89, .37, .14	.63, .28, .33

(p-value)





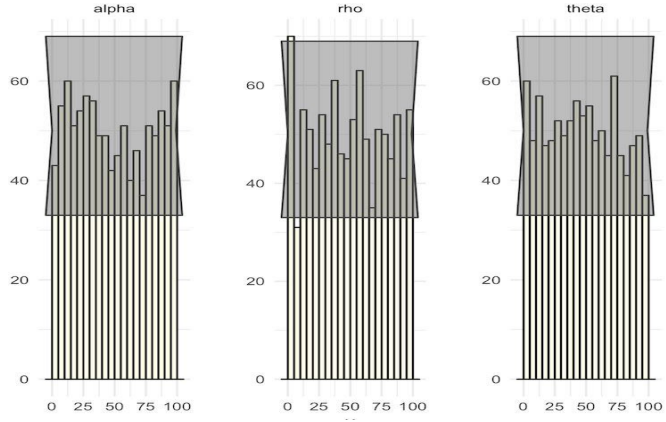
computed posterior
narrower than true posterior



Data = 20

Prior parameter: scale

	α, ρ, θ P-value SBC chi-sq test	Different measure
α~N(0.7, 0.01) ρ~ N(2.2, 0.01)	0.694 0.036 0.806	0.097 0.124 0.080
α~N(0.7, 0.05) ρ~ N(2.2, 0.05)	0.757 0.052 0.544	0.106 0.099 0.097
α~N(0.7, 0.1) ρ~ N(2.2, 0.1)	0.90 0.37 0.14	0.071 0.107 0.124




```
MW1 <- function(bin_count){
  bins <- length(bin_count)
  unif <- rep(1/bins, bins)
  M <- sum(bin_count)
  tempf <- Vectorize(function(i)  abs(bin_count[i]/M  - unif[i]))
  val <- integrate(tempf,1,bins, rel.tol=.Machine$double.eps^.05)$value
  return(val)
}
```

Data = 20, Prior parameter(.7,.05),(2.2,.05)

Prior shape

	α P-value	ρ P-value
α~N(0.7, 0.05) ρ~ N(2.2, 0.05)	0.69	0.31
α~inv-gamma ρ~ inv-gamma	0.75	0.29
α~t(4, 0.7, 0.05) ρ~ t(4, 2.2, 0.05)	0.64	0.25

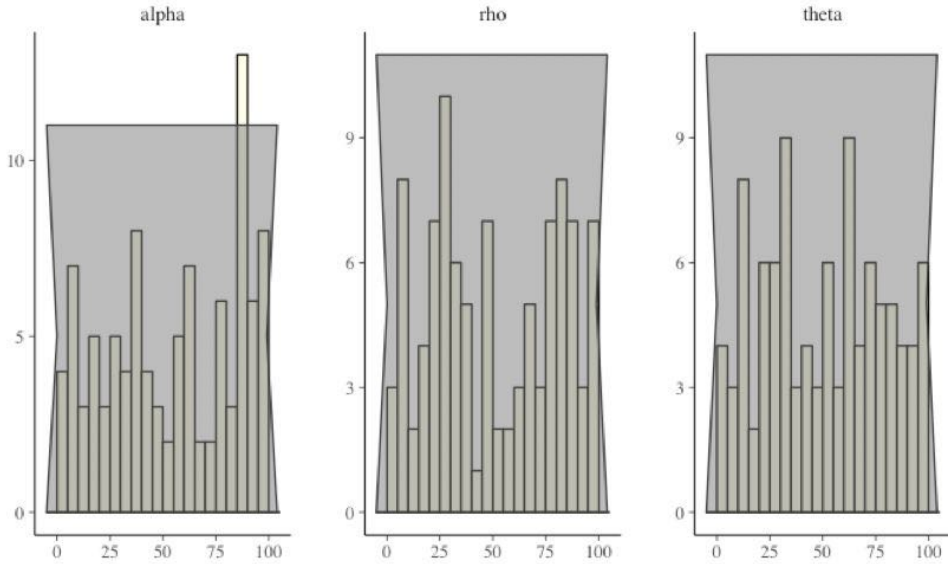
prior shapes affect SBC results as they have different tail, symmetry

2. Likelihood

Data = 20, Prior parameter(.7,.05),(2.2,.05), Prior shape: half-Normal

likelihood

Avg. time	poisson	bernoulli
α~N(0.7, 1) ρ~ N(2.2, 1)	fit failed	0.7
α~N(0.7, 10) ρ~ N(2.2, 10)	fit failed	1.28
α~N(0.7, 100) ρ~ N(2.2, 100)	fit failed	1.29

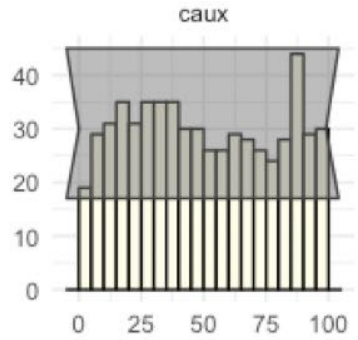


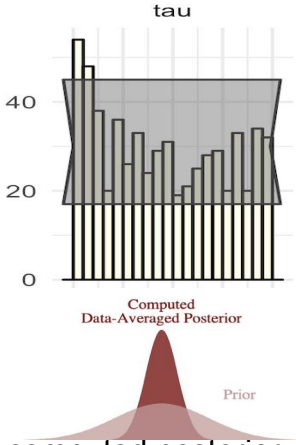
Pvalue: 0.091 0.167 0.710

In sbc world, any is possible!
Bernoulli likelihood can be used for poisson dgp data-> need care

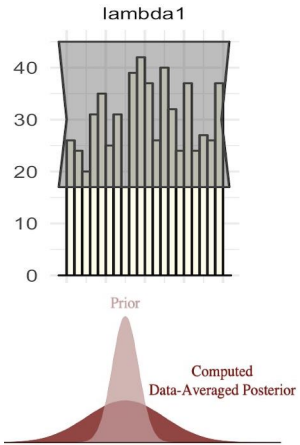
3. Data

Bernoulli-logit regression lgm. SBC results

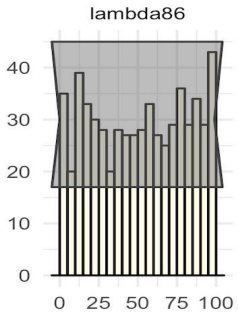


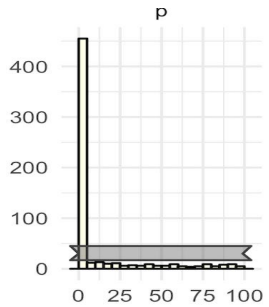


computed posterior
narrower than true posterior



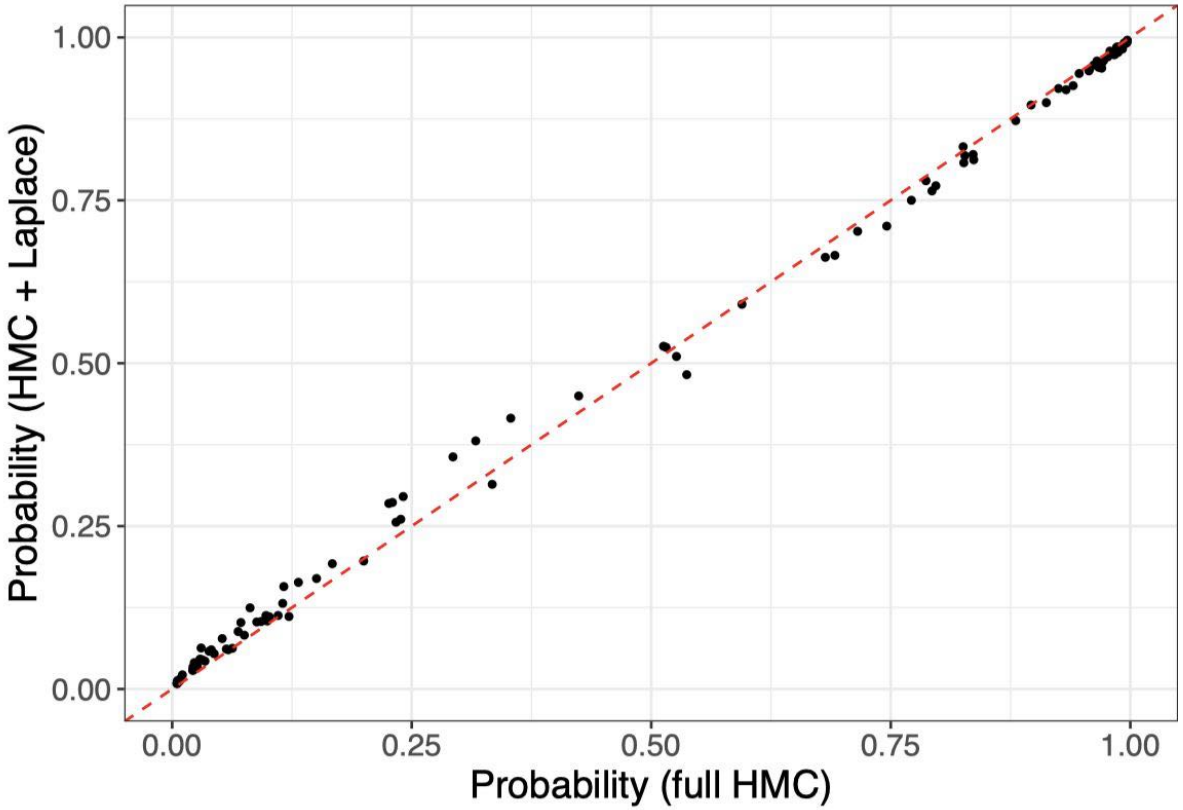
computed posterior
wider than true posterior

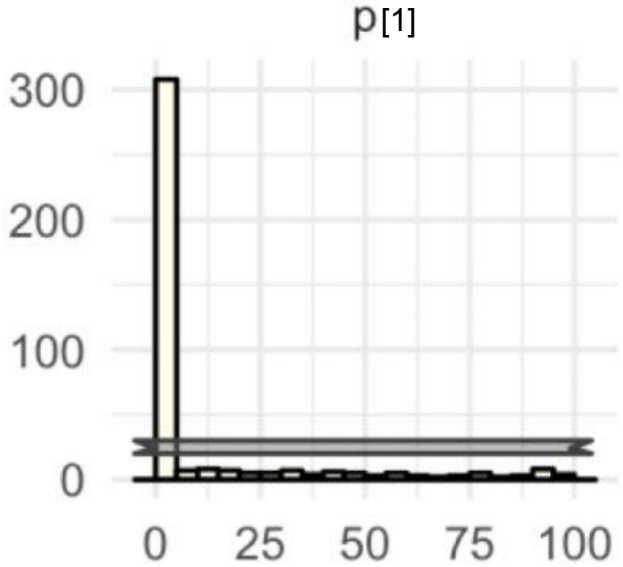




	caux	tau	lambda1	lambda86	p[1]
pval	0.58	3.2e-05	3.1e-02	3.7e-01	0
integ_msr	0.11	0.21	0.20	0.12	1.40
max_relative_diff	0.47	0.80	0.43	0.43	14.17

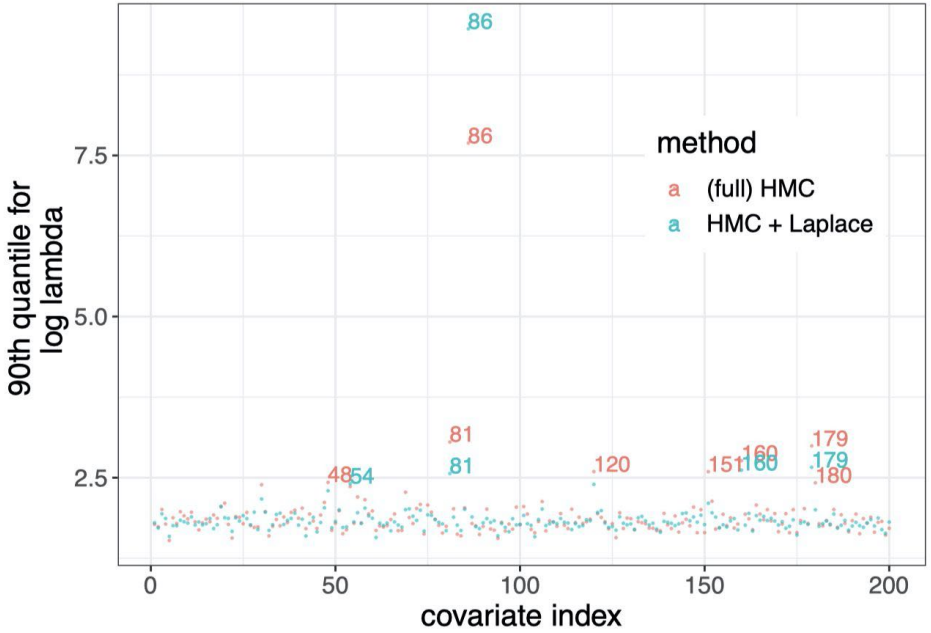
More overestimating than under

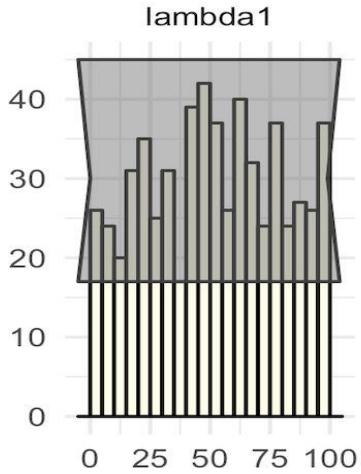


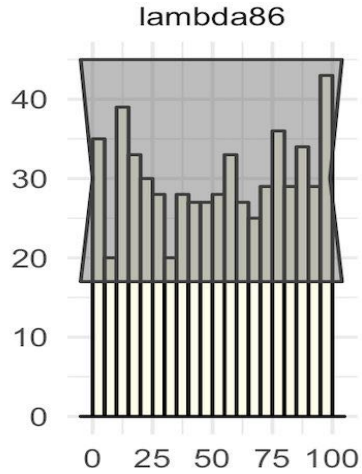


algorithm is strongly biased towards larger values of p in the true posterior.

Lambda interpretation: different bias







comments

Models simulate the world.

Even self-consistent validation needs external reference - at the very least for efficiency

How should we choose SBC prior and likelihood? - especially prior parameters which currently depends on modeler's choice.

Stan Development Team

2020-07-26

Here is a Stan program for a beta-binomial model

```
data {
  int<lower = 1> N;
  real<lower = 0> a;
  real<lower = 0> b;
}
transformed data { // these adhere to the conventions above
  real pi_ = beta_rng(a, b);
  int y = binomial_rng(N, pi_);
}
parameters {
  real<lower = 0, upper = 1> pi;
}
model {
  target += beta_lpdf(pi | a, b);
  target += binomial_lpmf(y | N, pi);
}
generated quantities { // these adhere to the conventions above
  int y_ = y;
  vector[1] pars_;
  int ranks_[1] = {pi > pi_};
  vector[N] log_lik;
  pars_[1] = pi_;
  for (n in 1:y) log_lik[n] = bernoulli_lpmf(1 | pi);
```

Contrary to real modeling context where cutting off certain prior domain based on observed data, SBC has no external reference which makes its prior parameter determination hard.

Prior to SBC prior parameter setting

1. Sensitivity test
   a. how sensitive is the SBC test result w.r.t hyperparameter(alpha)
   b. Sensitivity of SBC result: $\int\int\int p(\tilde{\theta}|\alpha)p(\tilde{y}|\tilde{\theta})p(\theta|\tilde{y},\alpha)\ell(\theta,\tilde{\theta})d\theta d\tilde{y}d\tilde{\theta}$
   c. covariance form of joint distribution of $\theta, \tilde{\theta}, \tilde{y},$

$$\left.\frac{d\mathbb{E}_{p_\alpha}\left[g\left(\theta\right)\right]}{d\alpha^{\intercal}}\right|_{\alpha_0} = \mathrm{Cov}_{p_0}\left(g\left(\theta\right), \left.\frac{\partial\rho\left(\theta,\alpha\right)}{\partial\alpha}\right|_{\alpha_0}\right)$$

2. Different power of test
   a. SBC may not be sensitive to misspecification when posterior dispersion << prior dispersion
   b. Truncating prior to, 1.25 * the box containing posterior samples before running SBC