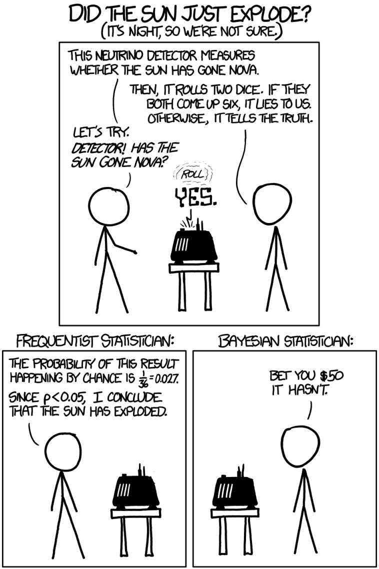
Bayesian Analysis in R: brms vs rstanarm

Alice Milivinti
a.lice.milivinti@gmail.com

12 april 2018

Bayesians vs Frequentist



Frequentists

Fundamentally related to the frequencies of repeated events.

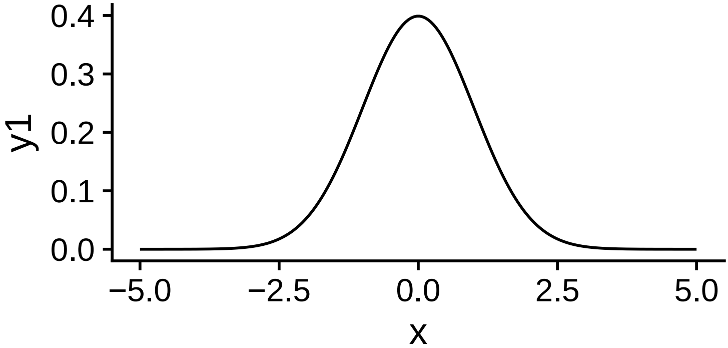
$$\mathbf{Pr(A \mid B) \propto Pr(B \mid A)}$$

$Pr(A \mid B)$: Posterior;
$Pr(B \mid A)$: Likelihood = Data Knowledge.
Given the observed data, what is the best estimate of the true value?

Maximum Likelihood

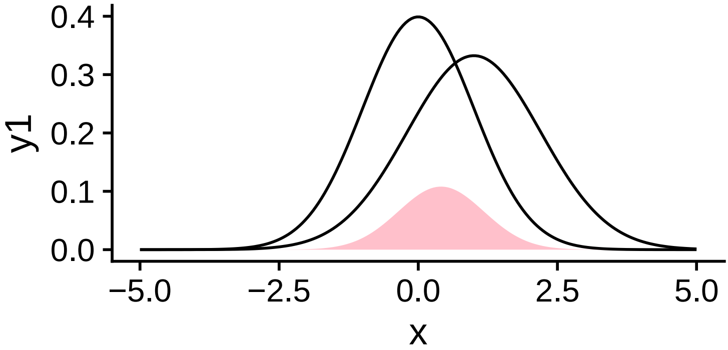
Model: each observation A_i drawn from a Gaussian of width e_i

$$P(B_i \mid A_{true}) = \frac{1}{\sqrt{2\pi e_i^2}} \exp\left[-\frac{(A_i - A_{true})^2}{2e_i}\right]$$



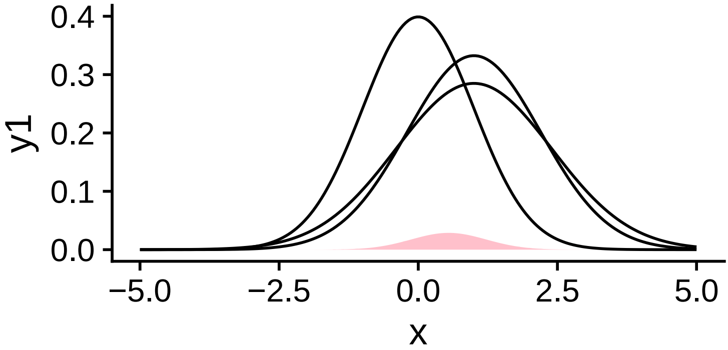
Buldind the Maximum Likelihood

$$\mathcal{L}(B \mid A_{true}) = \prod_{i=1}^{N} P(B_i \mid A_{true})$$



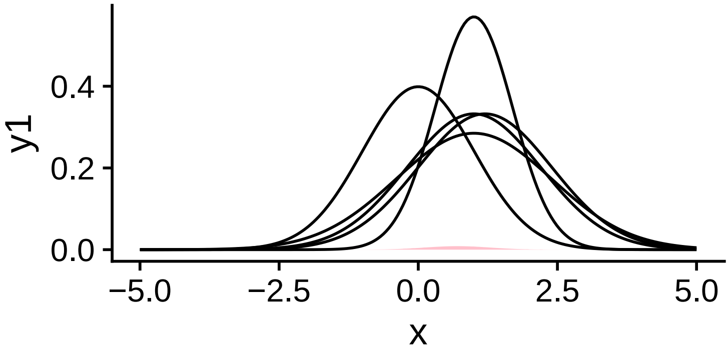
Buldind the Maximum Likelihood

$$\mathcal{L}(B \mid A_{true}) = \prod_{i=1}^{N} P(B_i \mid A_{true})$$



Buldind the Maximum Likelihood

$$\mathcal{L}(B \mid A_{true}) = \prod_{i=1}^{N} P(B_i \mid A_{true})$$



Probability is...

Bayesian

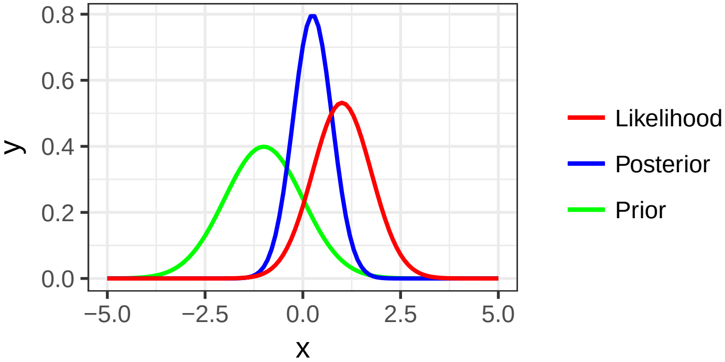
Fundamentally related to OUR OWN certainty or uncertainty of events.

$$\mathbf{Pr}(\mathbf{A} \mid \mathbf{B}) \propto \mathbf{Pr}(\mathbf{B} \mid \mathbf{A})\mathbf{Pr}(\mathbf{A})$$

$Pr(A \mid B)$: Posterior;
$Pr(B \mid A)$: Likelihood = Data Knowledge.
$Pr(A)$: Prior = Our Knowledge.

Prior, Likeloihood and Posterior

$$\mathbf{Pr(A \mid B) \propto Pr(B \mid A) Pr(A)}$$

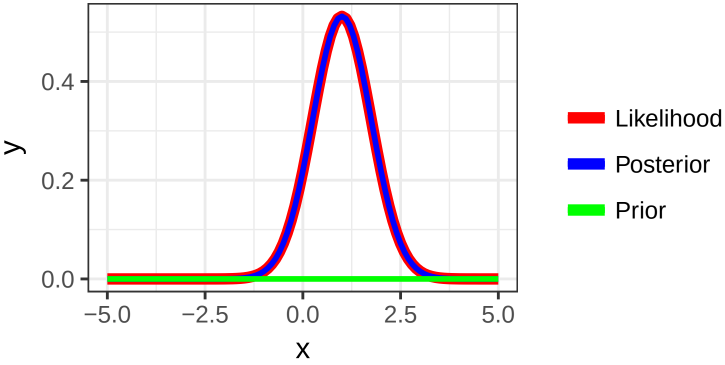


A word on the priors...

Priors can be: informative, weakly informative or uninformative.
A Bayesian analysis which uses an uninformative prior, such as

$$\mathcal{U}\left(-\infty, +\infty\right)$$

will give the same result as a frequentist analysis.



Sample the Posterior

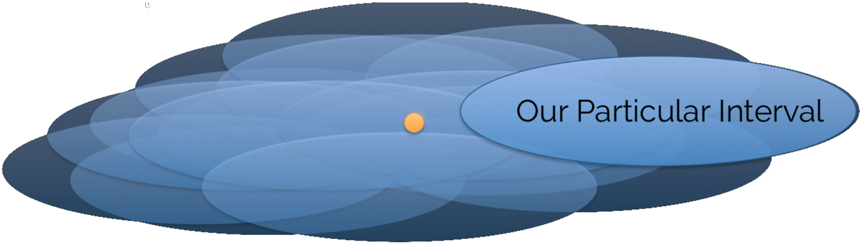
So far we have investigated how to use Bayesian techniques to determine posterior probability distribution for a set of parameters in light of some data.
However, our parameter set may be highly-dimensional, and we may only be interested in a sub-set of (marginalized) parameters.
Markov Chain Monte Carlo (MCMC) is an efficient approach to this problem.

Markov Chain Monte Carlo

Straight-forward Monte Carlo integration suffers from some problems...(especially if your posterior probability is peaked in a small volume of your parameter space).
Need for a method to throw down more points into the volume in regions of interest, and not waste points where the integrand is negligible.
We can use a Markov Chain to walk through the parameter space, vagabonding in regions of high significance, and avoiding everywhere else.
MCMC Algorithms: ex. Metropolis-Hasting, Gibbs, etc.

Frequestist Approach

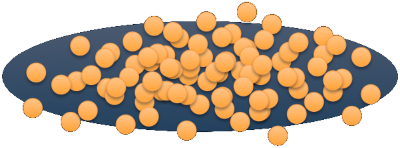
Frequentism is a probabilistic statement about a recipe for generating confidence intervals given a fixed model parameter



Credits: VanderPlas, ((2014))

Bayesian Approach

Bayesianism is aprobabilisitic statement about model parameters given a fixed credible region



Credits: VanderPlas, ((2014))

Please Remember This

A frequentist 95% confidence interval is NOT 95% likely to contain the true value!

This very common mistake is a Bayesian interpretation of a frequentist construct.

Take Home Message

Frequentists A 95% of such Confidence Intervals in repeated experiments will contain the true value!

Bayesians A 95% Credible Region is 95% likely to contain the true value!

STAN

Stan language (Stan Development Team, 2015) which makes use of the Hamiltonian Monte-Carlo Sampler (Neal et al., 2011) of Hybrid Monte-Carlo Sampler (HMC) (Duane et al., 1987) and its extension No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014).

- Much higher effective sample size per iteration for complex posteriors.
- Overall, much higher number of effective samples per second.
- Does not require any special behaviour for conjugate priors, which much impact the priors' choice (Hoffman and Gelman, 2014).

STAN & R

- `rstan`: R Interface to Stan C++ library for Bayesian estimation. Upload your stan code and run it through R.

- `rstanarm` & `brms`: Estimate previously compiled regression models using `rstan`. Users specify models via the R syntax with a formula and data.frame plus some additional arguments for priors.

rstanarm & brms

`rstanarm`

- manual's pages: 121
- topics documented: 47
- authors & contributors: 17, Jonah Gabry, Imad Ali, Sam Brilleman, Jacqueline Buros Novik, AstraZeneca, Trustees of Columbia University, Simon Wood, R Core Deveopment Team, Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker, Brian Ripley, William Venables, Ben Goodrich

`brms`

- manual's pages: 154
- topics documented: 144 (so detailed!)
- author: 1, Paul-Christian Bürkner

Syntax Comparison with mtcars

With default weakly informative priors:

`rstanarm`:

```
stan_glm(formula = mpg ~ wt + am + cyl, data = mtcars,
         prior = NULL, family = gaussian(), chains = 4,
         iter = 2000, warmup = 1000)
```

`brms`:

```
brm(formula = mpg ~ wt + am + cyl, data = mtcars,
    prior = NULL, family  = "gaussian", chains = 4,
    iter = 2000, warmup = 1000)
```

system.time() and Marvok Chains

`rstanarm`

system.time():

user system elapsed

0.924 0.000 0.918

1st Chain

0.107433 seconds (Warm-up)

0.09435 seconds (Sampling)

0.201783 seconds (Total)

`brms`

system.time():

Compiling the C++ model

user system elapsed

50.728 1.276 52.083

1st Chain

0.070122 seconds (Warm-up)

0.068211 seconds (Sampling)

0.138333 seconds (Total)

Random Effect Coefficients

Generalized linear models with group-specific terms:

`rstanarm`:

```
stan_glmer(formula = mpg ~ wt + am + (1|cyl), data = mtcars,
           prior = NULL, family = gaussian(), chains=4,
           iter=2000, warmup=1000)
```

`brms`:

```
brm(formula = mpg ~ wt + am + (1|cyl), data = mtcars,
    prior = NULL, family="gaussian", chains=4,
    iter=2000, warmup=1000)
```

Smooth Terms

The implementation is similar to that used in the gamm4 package:
rstanarm:

```
dat <- mgcv::gamSim(1, n = 200, scale = 2)

stan_gamm4(y ~ s(x0) + x1 + (1|x2) + s(x3), data = dat,
                  prior = NULL, family = gaussian(), chains=4,
                  iter=2000, warmup=1000)
```

brms:

```
brm(y ~ s(x0) + x1 + (1|x2) + s(x3), data = dat,
    prior = NULL, family="gaussian", chains=4,
    iter=2000, warmup=1000)
```

Priors' Specification: the Dirty Job

The packages offer all the priors' you would ever need: Student t family, Hierarchical shrinkage family, Laplace family, Product-normal family, Dirichlet family, etc.

`rstanarm`:

```
stan_glmer(mpg ~ wt + am + (1|cyl), data = mtcars,
           prior = student_t(df=4, location=0, scale=2.5),
           prior_intercept = cauchy(location=0, scale=10))
```

In `rstanarm` you cannot (to the best of my knowledge), choose different priors for different coefficeints since this would break vectorization. In `brms` cou can, but it may slow down the process.

Getting into the Priors'

In brms only you can start with:

```
get_prior(formula = mpg ~ wt + am + (1|cyl),
          data = mtcars, family="gaussian")

##                  prior     class      coef group resp dpar
## 1                              b
## 2                              b        am
## 3                              b        wt
## 4 student_t(3, 19, 10) Intercept
## 5  student_t(3, 0, 10)        sd
## 6                             sd             cyl
## 7                             sd Intercept   cyl
## 8  student_t(3, 0, 10)     sigma
```

Priors' Specification: the Dirty Job

```
prior <- c(set_prior("normal(0,10)", class = "b"),
           set_prior("normal(1,2)", class = "b", coef = "wt"),
           # Sd of group-level (random) effects
           set_prior("cauchy(0,2)", class = "sd",
            group = "cyl", coef = "Intercept")),
           set_prior("student_t(3, 0, 10)", class = "sigma"))


brm(mpg ~ wt + am + (1|cyl), data = mtcars, prior = prior)
```

For gamm you can specify the standard deviation of the smooth terms: `class = sds`.

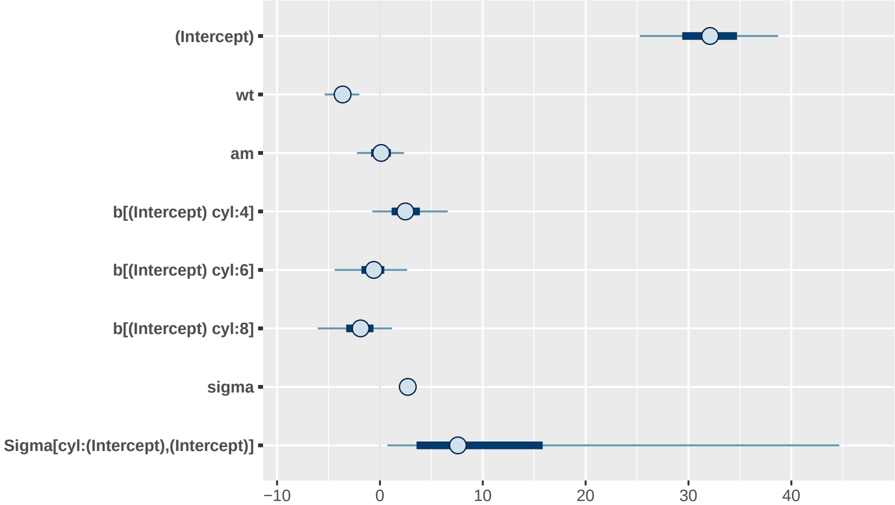
Parallelize the Chains

It is possible to parallelaze the Markov chains in both packages by using the argument `cores = ...` within the function.
Or by using more general parallel syntax:

```
options (mc.cores=parallel::detectCores ())
```

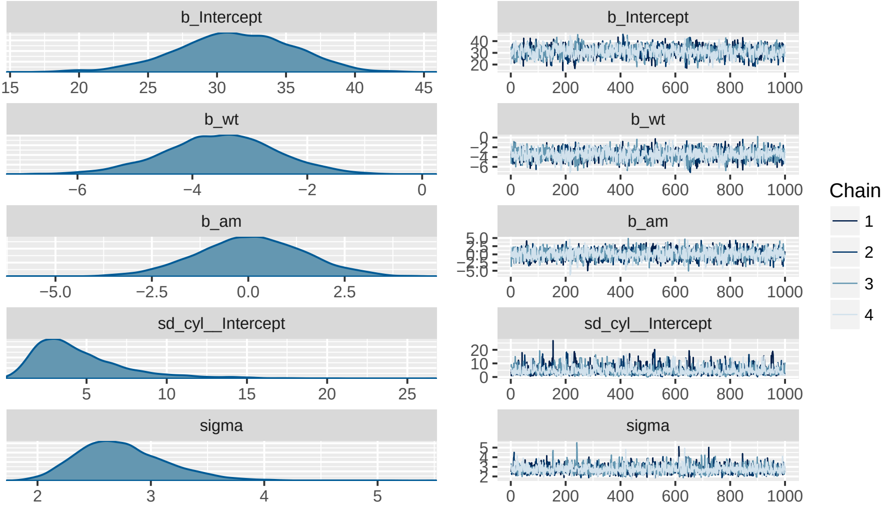
Plot the Results'

```
plot(rstanarm_glmer)
```



Plot the Results'

```
plot(brm_re)
```



Results' Diagnostics: shinystan

shinystan workd both for stanreg and brmsfit objects:

```
m1 <- stan_glmer(formula = mpg ~ wt + am + (1|cyl),
                 data = mtcars,  prior = NULL,
                 family = gaussian())

launch_shinystan(m1)
```

Get the STAN code

`rstanarm: shinystan`
`brms: make_stancode!`

```
make_stancode(mpg ~ wt + am + cyl, data = mtcars,
    prior = NULL, family  = "gaussian")
```

model
vector[N] mu = Xc * b;
for (n in 1:N)
mu[n] = mu[n] + (r_1_1[J_1[n]]) * Z_1_1[n];
// priors including all constants
target += student_t_lpdf(temp_Intercept — 3, 0, 10);
target += student_t_lpdf(sd_1 — 3, 0, 10)

Some Differences

With `brms` you can also implement multinomial logistic regressions

```
brm(Species ~ Petal.Length + Petal.Width + Sepal.Length +
      Sepal.Width, data=iris, family="categorical",
    prior=c(set_prior("normal (0, 8)")))
```

Correlations

With brms you can also deal easily with different correlation structures by specifieng:

- **cor_arma**: autoregressive-moving average (ARMA) structure.
- **cor_arr**: response autoregressive (ARR) structure
- **cor_car**: Spatial conditional autoregressive (CAR) structure
- **cor_sar**: Spatial simultaneous autoregressive (SAR) structure
- **cor_bsts**: Bayesian structural time series (BSTS) structure
- **cor_fixed**: fixed user-defined covariance structure

```
brm(mpg ~ wt + am + (1|cyl), data = mtcars, prior = NULL,
    family="gaussian", cor_arma(formula = ~1, q = 1))
```

Conclusions

- `rstanarm` can be the easiest package to start with since precompiled, but it might be limiting (intentionally) for more advanced needs.
- `brms` is more flexible and customizable.
- `rstan` fully flexible, but you need to learn STAN programming (maybe with the help of `make_stancode`).

stanarm

josie hughes @josie_shoes · 24 Mar 2017
I have good results! On Friday afternoon! The #rstanarm R pkg is lovely.
Thanks @mcmc_stan. #rstats

1 6

Heidi Lorimor @heidi_lorimor · 28 Oct 2016
running my first bayesian models today! #rstanarm and #shinystan are
amazing!

2

Roland Schäfer @codeslapper · 11 Apr 2016
But I want to make it clear: #rstanarm is REALLY a great tool, esp. for a
#frequentist who wants to play around...

> **Roland Schäfer** @codeslapper
> With #rstanarm I can now easily verify that all my #glmer models turn out
> exactly the same with #Bayesian estimation taking 50x longer ;)

brms

Frank Harrell
@f2harrell

Follow

Bayesian regression modeling: R brms package a breakthrough, & article by Bürkner is as well written as it is useful: jstatsoft.org/article/view/v...

Shravan Vasishth @vasishthlab · 22 Sep 2017
Replying to @f2harrell
Bürkner deserves a prize.

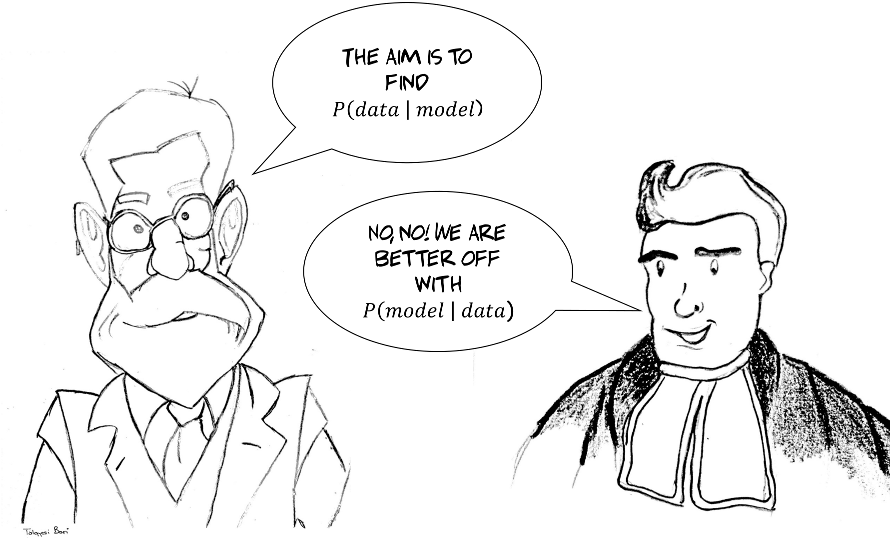
Stephen Martin @smartin2018 · 23 Sep 2017
And brms is just crazy potent. Want a location-scale-shape crossed random effects mixture model? You can. Goodness.

1

Should I be Bayesian?

Sampling can be slow
You need to be really careful about diagnostics
You need to have ideas about priors
BUT!
It can help when dealing with small data
In forecasts since:
"...in terms of forecasting ability, ...a good Bayesian will beat a non-Bayesian, who will do better than a bad Bayesian."
C.W.J. Granger (1986, p. 16)
At the end it is a real philosophical question about how you intend statistics.

The End


THE AIM IS TO FIND $P(data \mid model)$
NO, NO! WE ARE BETTER OFF WITH $P(model \mid data)$

References I

- Duane, Simon et al. (1987). “Hybrid monte carlo”. In: *Physics letters B* 195(2), pp. 216–222.
- Hoffman, Matthew D and Andrew Gelman (2014). “The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” In: *Journal of Machine Learning Research* 15(1), pp. 1593–1623.
- Neal, Radford M et al. (2011). “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov Chain Monte Carlo* 2, pp. 113–162.
- Stan Development Team (2015). *Stan Modeling Language User's Guide and Reference Manual, Version 2.10.0*. URL: http://mc-stan.org/.
- VanderPlas, Jake (2014). URL: https://speakerdeck.com/jakevdp/frequentism-and-bayesianism-whats-the-big-deal-scipy-2014.