

Aggregated Gaussian

Parametrisation

This is shorthand to allow for aggregated Gaussian observations, where we have repeated observations with the same mean and (known scaled) precision. These can be aggregated into an equivalent likelihood reducing the computational effort.

Let $y = (y_1, \dots, y_n)$ be iid observations of a Gaussian with mean μ and precisions $s_i\tau$, where the s_i 's are fixed and known scaling parameters (default $s_i = 1$), then

$$f(y|\mu, \tau) = \prod_{i=1}^n (2\pi)^{-1/2} (s_i\tau)^{1/2} \exp\left(-\frac{1}{2}(s_i\tau)(y_i - \mu)^2\right) \quad (1)$$

$$= (2\pi)^{-n/2} \tau^{n/2} \left(\prod_{i=1}^n s_i^{1/2}\right) \exp\left(-\frac{1}{2}\tau \sum_{i=1}^n s_i (y_i - \mu)^2\right) \quad (2)$$

$$= (2\pi)^{-n/2} \tau^{n/2} \left(\prod_{i=1}^n s_i^{1/2}\right) \exp\left(-\frac{1}{2}m\tau [(\bar{y} - \mu)^2 + v]\right) \quad (3)$$

where

$$\begin{aligned} m &= \sum_{i=1}^n s_i \\ \bar{y} &= \frac{1}{m} \sum_{i=1}^n s_i y_i \\ v &= \frac{1}{m} \sum_{i=1}^n s_i y_i^2 - \bar{y}^2 \end{aligned}$$

Link-function

The mean μ is linked to the linear predictor η by

$$\mu = \eta$$

Hyperparameters

The hyperparameter is θ , where

$$\theta = \log \tau$$

and the prior is defined on θ .

Specification

This family require the response to be an `inla.mdata`-object, where each row¹ is

$$(v, \frac{1}{2} \sum_{i=1}^n \log(s_i), m, n, \bar{y})$$

This object is most easily constructed using the `inla.agaussian()` function, which gives the object to use directly.

- `family="agaussian"`
- Required arguments: An `inla.mdata`-object created with `inla.agaussian()`.

¹It is a list of vectors, so not strictly a "row".

Hyperparameter specification and default values

doc The aggregated Gaussian likelihood

hyper

theta

```
hyperid 66001
name log precision
short.name prec
output.name Precision for the AggGaussian observations
output.name.intern Log precision for the AggGaussian observations
initial 4
fixed FALSE
prior loggamma
param 1 5e-05
to.theta function(x) log(x)
from.theta function(x) exp(x)
```

survival FALSE

discrete FALSE

link default identity logit loga cauchit log logoffset

pdf agaussian

Example

```
## plain example, either with all n observations or to aggregate them
```

```
n <- 10
```

```
y <- rnorm(n)
```

```
Y <- inla.agaussian(y)
```

```
r <- inla(Y ~ 1,
```

```
  data = list(Y = Y),
```

```
  family = "agaussian")
```

```
rr <- inla(y ~ 1,
```

```
  data = data.frame(y),
```

```
  family = "gaussian")
```

```
print(r$mlik - rr$mlik)
```

```
inla.dev.new()
```

```
par(mfrow = c(1, 2))
```

```
plot(r$internal.marginals.hyperpar[[1]], pch = 19, main = "prec")
```

```
lines(rr$internal.marginals.hyperpar[[1]], lwd = 3)
```

```
plot(r$marginals.fixed$(Intercept)', pch = 19, main = "intercept")
```

```
lines(rr$marginals.fixed$(Intercept)', lwd = 3)
```

```
#####
#####
#####
```

```

## same example, but with different scalings for the precision for 'yy'
n <- 5
s <- 1:n ## scale the precision
y <- rnorm(n)
yy <- rnorm(n, sd = sqrt(1/s))
Y <- inla.agaussian(rbind(y, yy),
                    rbind(rep(1, n), s))

r <- inla(Y ~ 1,
          data = list(Y = Y),
          family = "agaussian",
          control.compute = list(cpo = TRUE, dic = TRUE))
rr <- inla(yyy ~ 1,
          data = data.frame(yyy = c(y, yy)),
          scale = c(rep(1, n), s),
          control.compute = list(cpo = TRUE, dic = TRUE),
          family = "gaussian")
print(r$mlik - rr$mlik)

inla.dev.new()
par(mfrow = c(1, 2))
plot(r$internal.marginals.hyperpar[[1]], pch = 19, main = "prec")
lines(rr$internal.marginals.hyperpar[[1]], lwd = 3)
plot(r$marginals.fixed$(Intercept)', pch = 19, main = "intercept")
lines(rr$marginals.fixed$(Intercept)', lwd = 3)

#####
#####
#####

## if one want to build the aggregated data for each replication
## at the time, one can do

y.agg <- unlist(inla.agaussian(y))
yy.agg <- unlist(inla.agaussian(yy, s))
agg.matrix <- rbind(y.agg, yy.agg)
Y.agg <- inla.mdata(agg.matrix)

## and then
r.agg <- inla(Y.agg ~ 1,
             data = list(Y.agg = Y.agg),
             family = "agaussian")

## For further details, see also INLA::inla.agaussian.test()

```

Notes

- Thanks to JW for suggesting this formulation and for providing the details.