Intercept-slope model

Parametrization

The intercept-slope model is a convenient re-implementation of a commonly used construct, where

$$(a, b)$$

is bi-variate Gaussian with a Wishart prior for the precision matrix[1], and various forms of

$$\gamma(a + bz), \tag{1}$$

where z is a covariate and γ is a (random) scaling, goes into the linear predictor. Replicates of (a, b) is indexed by *subject*, $i = 1, \ldots, n$, and the various scaling of Eq. 1 by *strata* $j = 1, \ldots, m$, leading to a model for (a subset of)

$$\{\gamma_j(a_i + b_i z_{ij}), \quad i = 1, \ldots, n, \quad j = 1, \ldots, m\},$$

as not all combinations need to be present.

Hyperparameters

The hyperparameters are $(\theta_1, \theta_2, \theta_3)$ as in the model "iid2d" (related to the precisions of a and b, and their correlation), and $\theta_4 = \gamma_1, \ldots, \theta_{13} = \gamma_{10}$. Since m is defined in the input, only $\gamma_1, \ldots, \gamma_m$ are used. m is limited to $m \leq 10$. **Please note** that all γ_i's are by default **fixed** to 1.

Specification

The is specified as

```
f(idx, model="intslope", hyper = ...,
  precision = exp(14),
  args.intslope = list(subject=i, strata=j, covariate = z))
```

The definition of the model is through the `args.intslope` argument, where `i` and `j` are factors/integers and `z` is numerical, all with same length N, say. The argument `idx`, index which row that is used for the linear predictor, hence values of `idx` must take integer values in the interval 1 to N. The precision argument, defines the tiny small noise added to each $\gamma(a+bz)$ to avoid a singular joint model. The `subject` and `strata` argument, is converted internally into integers $1, 2, \ldots$, using

```
subject = as.numerical(as.factor(subject))
strata  = as.numerical(as.factor(strata))
```

and the results is shown after this conversion.

Hyperparameter specification and default values

doc Intecept-slope model with Wishart-prior

hyper

theta1

hyperid 16101

name log precision1

[1]The documentation for the model "iid2d" gives the details of the definition of the parameterization of the precision matrix and the Wishart-prior.

short.name prec1
initial 4
fixed FALSE
prior wishart2d
param 4 1 1 0
to.theta `function(x) log(x)`
from.theta `function(x) exp(x)`

theta2

hyperid 16102
name log precision2
short.name prec2
initial 4
fixed FALSE
prior none
param
to.theta `function(x) log(x)`
from.theta `function(x) exp(x)`

theta3

hyperid 16103
name logit correlation
short.name cor
initial 4
fixed FALSE
prior none
param
to.theta `function(x) log((1 + x) / (1 - x))`
from.theta `function(x) 2 * exp(x) / (1 + exp(x)) - 1`

theta4

hyperid 16104
name gamma1
short.name g1
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta5

hyperid 16105
name gamma2
short.name g2
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta6

hyperid 16106

name gamma3
short.name g3
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta7

hyperid 16107
name gamma4
short.name g4
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta8

hyperid 16108
name gamma5
short.name g5
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta9

hyperid 16109
name gamma6
short.name g6
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta10

hyperid 16110
name gamma7
short.name g7
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta11

hyperid 16111
name gamma8
short.name g8
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta12

hyperid 16112
name gamma9
short.name g9
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

theta13

hyperid 16113
name gamma10
short.name g10
initial 1
fixed TRUE
prior normal
param 1 36
to.theta `function(x) x`
from.theta `function(x) x`

constr FALSE

nrow.ncol FALSE

augmented FALSE

aug.factor 1

aug.constr

n.div.by

n.required FALSE

set.default.values TRUE

status experimental

pdf intslope

Example

```
library(mvtnorm)
n = 300
idx = 1:n
nstrata = 3
strata = sample(1:nstrata, n, replace=TRUE)
nsubject = n %/% nstrata
subject = sample(1:nsubject, n, replace=TRUE)
z = rnorm(n)
gam = c(1, 1 + rnorm(nstrata-1, sd = 0.2))

rho = sqrt(3)/2
Sigma = matrix(c(1/1, NA, NA, 1/2), 2, 2)
Sigma[1,2] = Sigma[2,1] = rho*sqrt(Sigma[1,1]*Sigma[2,2])

ab = rmvnorm(nsubject, sigma=Sigma)
a = ab[,1]
b = ab[,2]
s = 0.01
y = gam[strata] * (a[subject] + z * b[subject]) + rnorm(n, s = 0.01)
r = inla(y ~ -1 + f(idx, model = "intslope",
                    args.intslope = list(subject = subject,
                                         strata = strata,
                                         covariates = z),
                    ## this is for nstrata = 3
                    hyper = list(gamma1 = list(fixed = TRUE),
                                 gamma2 = list(fixed = FALSE),
                                 gamma3 = list(fixed = FALSE))),
         data = list(y = y,
                     idx = idx,
                     subject = subject,
                     strata = strata,
                     z = z),
         control.family = list(hyper = list(
                                   prec = list(initial = log(1/s^2),
                                               fixed=TRUE))))
summary(r)
```

Notes

- With $n_s = \max(\text{subject})$, the internal storage of this model is

$$(\gamma_{j_1}(a_{i_1} + z_1 b_{i_1}), \ldots, \gamma_{j_N}(a_{i_N} + z_N b_{i_N}), a_1, \ldots, a_{n_s}, b_1, \ldots, b_{n_s}),$$

  i.e. a vector of length $N + 2n_s$.