

Conditional sampling from a fitted model

Haavard Rue (hrue@r-inla.org)

Aug 19th 2017

Details of this vignette requires the classic-mode, but it is possible to make it work for the new compact-mode. Please contact help@r-inla.org if that is required.

```
inla.setOption(inla.mode="classic")
```

This short note describe how to do conditional sampling from a fitted model, where we want to condition on new events. An easy example which motivated this note, is the following.

```
n = 50
grp.len = 5
x = rnorm(n)
g = rnorm(grp.len)
grp = rep(1:grp.len, each = n %% grp.len)
y = 1 + x + g[grp] + rnorm(n, sd = 0.01)
```

and the task is to provide the posterior a new observation, condition on that the random effect ‘g’ is zero. Although it is possible to get this directly (an exercise for the reader), in general, we will show that it is easier to compute this using Monte Carlo sampling and the function `inla.posterior.sample()`. With a little hack.

To fit the model, we do

```
r = inla(y ~ x + f(grp, model = "iid"),
        data = data.frame(y, x, grp),
        control.compute = list(config=TRUE))
```

```
## Warning in normalizePath(inla.dir):
## path[1]="/tmp/RtmpggzAMI/filed8b64fe580e7": No such file or directory
```

adding `config=TRUE` to prepare for the use of `inla.posterior.sample()`.

The trick, is to add the conditioning as constraints, like

$$Ax = e$$

where x is the latent field. In our case, it is simply `t grp.len` constraints, $\{g_i = 0\}$ for i from 1 to 5. To do this, we need to create the matrix A , and vector e , and add it to the list of constraints (if any). To do this, we need to know the index of the g -term, which we find here

```
r$misc$configs$contents
```

```
## $tag
## [1] "Predictor"    "grp"          "(Intercept)" "x"
##
## $start
## [1] 1 51 56 57
##
```

```
## $length
## [1] 50  5  1  1
```

so we have both the start index and the length of this vector. So we create a constraint $Ax = 0$ to represent constraints, $\{g_i = 0\}$ for i from 1 to 5.

```
m = sum(r$misc$configs$contents$length)
grp.idx = which(r$misc$configs$contents$tag == "grp")
grp.len = r$misc$configs$contents$length[grp.idx]
A = matrix(0, grp.len, m)
e = matrix(0, grp.len, 1)
for(i in 1:grp.len) {
  A[i, r$misc$configs$contents$start[grp.idx] + i - 1] = 1
  e[i] = 0
}
```

Then we need to append it to the existing one if any, or to add a new one.

```
constr = r$misc$configs$constr
if (is.null(constr)) {
  ## nothing there from before
  r$misc$configs$constr = list(
    nc = dim(A)[2],
    A = A,
    e = e)
} else {
  ## create a new one
  r$misc$configs$constr = list(
    nc = constr$nc + dim(A)[2],
    A = rbind(constr$A, A),
    e = rbind(constr$e, e))
}
```

When we now do

```
xx = inla.posterior.sample(1000, r)
```

we additional condition on $Ax = e$, but g would still appear in the list of samples as zero or something very close to it. The rest should be straight forward.

PS: Note that you can add any linear combination of x as a constraint, you just do not want to many, as the cost is quadratic the number of constraints. If the constraints simply set variables to zero, you can also add a large number to $Q_{i,i}$, which you will find in `r$misc$configs`, one for each configuration.