The Ornstein-Uhlenbeck process

Parametrization

The Ornstein-Uhlenbeck process is defined with (mean zero), as the SDE

$$dx_t = -\phi x_t + \sigma dW_t$$

where $\phi > 0$ and W_t is the Wiener process. This is the continuous time analogue to the discrete time AR(1) model.

The process has a Markov property. Let $x = (x_1, x_2, \ldots, x_n)$ be value of the process at increasing time-points $t = (t_1, t_2, \ldots, t_n)$, then the conditional distribution

$$x_i \mid x_1, \ldots, x_{i-1}, \qquad i = 2, \ldots, n,$$

is Gaussian with mean

$$x_{i-1} \exp(-\phi\delta_i)$$

and precision

$$\tau \left(1 - \exp(-2\phi\delta_i)\right)^{-1}$$

where

$$\delta_i = t_i - t_{i-1}, \qquad i = 2, \ldots, n$$

and

$$\tau = 2\phi/\sigma^2.$$

The marginal distribution for x_1 is taken to be the stationary distribution, which is a zero mean Gaussian with precision τ.

Hyperparameters

The precision parameter τ is represented as

$$\theta_1 = \log(\tau)$$

where τ is the *marginal* precision for the Ornstein-Uhlenbeck process given above.

The parameter ϕ is represented as

$$\theta_2 = \log(\phi)$$

and the prior is defined on $\theta = (\theta_1, \theta_2)$.

Specification

The Ornstein-Uhlenbeck model is specified inside the `f()` function as

```
f(<whatever>, model="ou", values=<values>, hyper = <hyper>)
```

The argument `values` gives the time-points where the process is defined/observed on (default $1, 2, \ldots, n$).

Hyperparameter specification and default values

hyper

- **theta1**
  - **name** log precision
  - **short.name** prec
  - **prior** loggamma
  - **param** 1 5e-05
  - **initial** 4
  - **fixed** FALSE
  - **to.theta**
  - **from.theta**
- **theta2**
  - **name** log phi
  - **short.name** phi
  - **prior** normal
  - **param** -2 1
  - **initial** -1
  - **fixed** FALSE
  - **to.theta**
  - **from.theta**

constr FALSE

nrow.ncol FALSE

augmented FALSE

aug.factor 1

aug.constr

n.div.by

n.required FALSE

set.default.values FALSE

pdf ou

Example

```
## simulate an OU-process and estimate its parameters back.
phi = -log(0.95)
sigma = 1
marg.prec = 2*phi/sigma^2
n = 1000
locations = cumsum(sample(c(1, 2, 5, 20),n, replace=TRUE))

## do it sequentially and slow (for clarity)
x = numeric(n)
```

```
x[1] = rnorm(1, mean=0, sd = sqrt(1/marg.prec))
for(i in 2:n) {
    delta = locations[i] - locations[i-1]
    x[i] = x[i-1] * exp(-phi * delta) +
        rnorm(1, mean=0,  sd = sqrt(1/marg.prec * (1-exp(-2*phi*delta))))
}

## observe it with a little noise
y = 1 + x + rnorm(n, sd= 0.01)
plot(locations, x, type="l")

formula = y ~  1 + f(locations, model="ou", values=locations)
r = inla(formula, data = data.frame(y, locations))
summary(r)
```

Notes

The Ornstein-Uhlenbeck process is the continuous-time analogue to the discrete AR(1) model (for positive lag-one correlation only), but they are parameterised slightly different.