SPDE one dimensional example

Elias T. Krainski

created: April 2, 2016, last update 2016-04-14

A SPDE model for one dimensional data

Introduction

In this example we show row to analyse a time series of daily temperature using a one dimension SPDE model. More details about it are on the paber at https://www.jstatsoft.org/article/view/v063i19

The data

We consider the daily weather data available at http://www.yr.no/. We have the following set the URL for the daily data for Trondheim considerint in the last 13 months

```
u0 <- paste0('http://www.yr.no/place/Norway/S%C3%B8r-Tr%C3%B8ndelag/',
             'Trondheim/Trondheim/detailed_statistics.html')
### browseURL(u0) ### to visit the web page
```

One can read and extract the desired data table (the second one at the URL) using the **XML** package with

```
require(XML)
d <- readHTMLTable(u0)[[2]]
```

However, it still need some pre-processing.

Without the **XML** package one can use

```
d0 <- readLines(u0) ### read it as text
i <- grep('<tr>', d0) ### index for each table line
i <- i[i>grep('<tbody>', d0)[2]] ### select those for the second table
```

The desired data we would like to analyse is the minimum and maximum temperature. Commands to extract and pre-process these data

```
dates <- as.Date(d0[i+1], format='      <th>%b %d, %Y</th>')
tmed <- as.numeric(gsub('<td>', '', gsub('°</td>', '', d0[i+4])))
(n <- length(dates)) ### it is daily over last 13 months
```

```
## [1] 410
```

Visualize it

```
pd <- pretty(c(dates, max(dates+30)), n=13)
par(mfrow=c(1,1), mar=c(3,3,0.5,2), mgp=c(2,.7,0), las=2, xaxs='i')
plot(dates, tmed, type='l', lwd=2,
     axes=FALSE, xlab='day', ylab='Temperature')
abline(h=0)
abline(h=3*(-8:9), v=pd, lty=3, col=gray(.5))
box()
axis(2, 3*(-8:9)); axis(4, 3*(-8:9))
axis(1, pd, months(pd, TRUE))
```

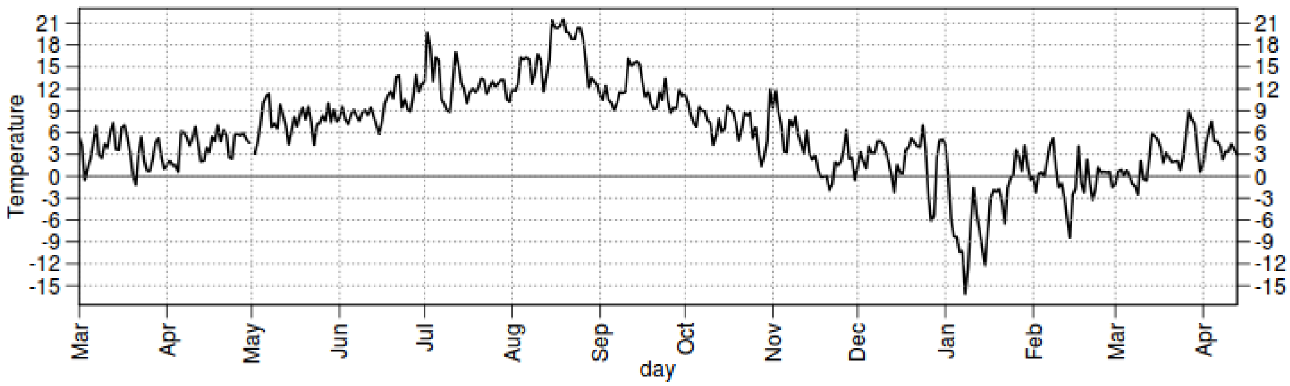

Figure 1: plot of chunk visualize

Model fitting

- **Mesh**: in 1d it is a matter of chosing a set of knots, the order of the basis functions and the boundary. Choosing first order basis function and Neumann boundary.

```
coo <- as.numeric(dates-min(dates)) ## have numeric temporal coordinates
mesh <- inla.mesh.1d(loc=seq(min(coo), max(coo), by=7), ## knots (7 days)
                 boundary='neumann', degree=2)  ### boundary and basis function degree
```

- **Define the** n x m **projector matrix** to project the process at the mesh nodes to locations

```
A <- inla.spde.make.A( ## projector creator
    mesh=mesh, ## provide the mesh
    loc=coo) ### locations where to project the field
dim(A) ## an 'n' by 'm' projector matrix

## [1] 410  58

summary(rowSums(A)) ### each line sums up to one

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1

summary(colSums(A)) ### 'how many' observations per knot

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.000   7.000   7.000   7.069   7.000  10.500
```

- **Build the SPDE model** on the mesh. Set $\alpha = 2$ to build the precision

```
spde <- inla.spde2.matern( ## precision components creator
    mesh=mesh, ## mesh supplied
    alpha=2) ## smoothness parameter
```

- **Create a data stack** to organize the data. This is a way to allow models with complex linear predictors. In our case, we have a SPDE model defined on m nodes. It must be combined with the covariate (and the intercept) effect at n locations. We do it using different projector matrices.

```
stk.e <- inla.stack( ## stack creator
  data=list(y=tmed),  ## response
  effects=list(## two elements:
    data.frame(b0=rep(1, n)), ## regressor part
    i=1:spde$n.spde),  ## RF index
  A=list(## projector list of each effect
    1, ## for the covariates
    A), ## for the RF
  tag='est') ## tag
```

- **Fit** the posterior marginal distributions for all model parameters

```
formula <- y ~ 0 + b0 + ## fixed part
  f(i, model=spde) ## RF term
res <- inla( ## main function in INLA package
  formula, ## model formula
  data=inla.stack.data(stk.e), ## dataset
  control.predictor=list( ## inform projector needed in SPDE models
    A = inla.stack.A(stk.e), compute=TRUE)) ## projector from the stack data
```

Posterior marginal distributions - PMDs

Summary of the regression coefficients PMDs

```
round(res$summary.fixed, 4)
```

```
##      mean     sd 0.025quant 0.5quant 0.975quant   mode kld
## b0 5.8624 1.9326     1.9638   5.8649     9.7465 5.8691   0
```

We have to transform the precision PMD to have the variance PMD. It can be done and visialized by

```
m.prec <- res$marginals.hyperpar$'Precision for the Gaussian observations' ## the marginal
post.s2e <- inla.tmarginal(## function to compute a tranformation
  function(x) 1/x, ## inverse transformation
  m.prec) ## marginal to be applied
plot(post.s2e, type='l', ylab='Density',
     xlab=expression(sigma[e]^2))
```

The SPDE approach uses a local variance, τ^2, such that $\sigma_s^2 = 1/(2\pi\kappa^2\tau^2)$. On **INLA** we work $\log(\tau^2)$ and $\log(\kappa)$. So, especially for σ_s^2, we have to do an additional computation. The PMDs for all RF parameters on user scale are computed by

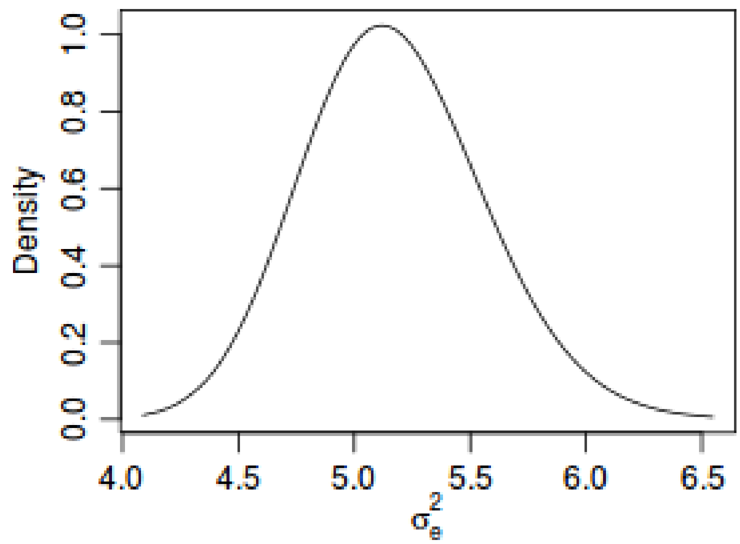

Figure 2: plot of chunk nugget

```
rf <- inla.spde.result( ## function to compute the 'interpretable' parameters
    inla=res, ## the inla() output
    name='i', ## name of RF index set
    spde=spde, ## SPDE model object
    do.transf=TRUE) ## to user scale
```

It can be visualized by

```
par(mfrow=c(1,3), mar=c(3,3,0.3,0.3), mgp=c(2,0.5,0))
plot(rf$marginals.var[[1]], ty='l',
     xlab=expression(sigma[s]^2), yla='Density')
plot(rf$marginals.kap[[1]], type='l',
     xlab=expression(kappa), ylab='Density')
plot(rf$marginals.range[[1]], type='l',
     xlab='range nominal', ylab='Density')
```

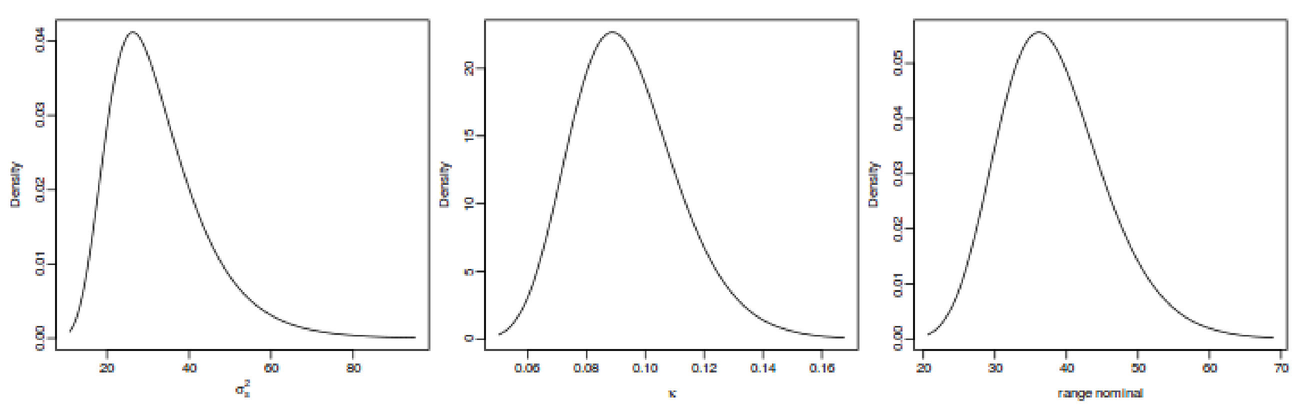

Figure 3: plot of chunk parameters

Predicted

Visualize it with the commands bellow

```
par(mfrow=c(1,1), mar=c(3,3,0.3,2), mgp=c(2,0.5,0), las=2, xaxs='i')
id <- inla.stack.index(stk.e, tag='est')$data
plot(dates, tmed, type='l', axes=FALSE, ylab='Temperature')
for (j in 3:5)
  lines(dates, res$summary.fitted.values[id, j], lty=3)
box(); axis(2, 3*(-8:9)); axis(4, 3*(-8:9))
axis(1, pd, months(pd, T))
abline(h=0)
abline(h=3*(-8:9), v=pd, lty=3, col=gray(.5))
```

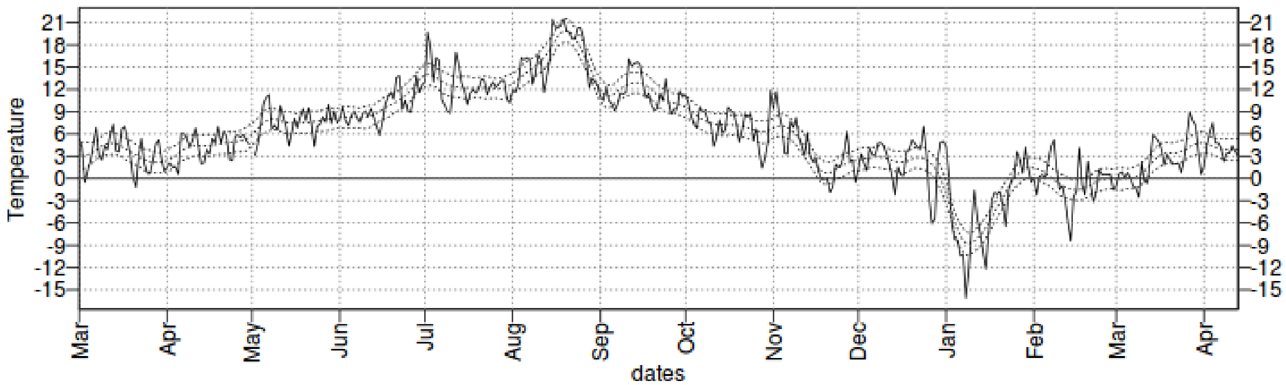

Figure 4: plot of chunk predicted

Just a look to the rest of the data

Pre-processing the maximum, minimum and normal temperature, the precipitation, and the average and maximum wind:

```
tmax <- as.numeric(gsub('<td>', '', gsub('°</td>', '', d0[i+2])))
tmin <- as.numeric(gsub('<td>', '', gsub('°</td>', '', d0[i+3])))
tnormal <- as.numeric(gsub('<td>', '', gsub('°</td>', '', d0[i+5])))
prec <- as.numeric(gsub('<td>', '', gsub('mm</td>', '', d0[i+6])))
wind <- as.numeric(gsub('<td>', '', gsub('m/s</td>', '', d0[i+10])))
wmax <- as.numeric(gsub('<td>', '', gsub('m/s</td>', '', d0[i+9])))
```

Visualize it

```
par(mfrow=c(3,1), mar=c(0.1,3,0.1,2), mgp=c(2,.7,0), las=2, xaxs='i')
plot(dates, tmed, type='l', ylim=range(tmin, tmax, na.rm=TRUE),
     axes=FALSE, xlab='', ylab='Temperature', col='green')
lines(dates, tmin, col='blue')
lines(dates, tmax, col='red')
lines(dates, tnormal)
legend(dates[which.min(tmin)], par()$usr[4], c('normal', 'max.', 'aver.', 'min.'),
      col=1:4, lty=1, ncol=2, xjust=0.5, bty='n')
abline(h=5*(-5:6), v=pd, lty=3, col=gray(.5))
```

```
box(); axis(2, 5*(-5:6)); axis(4, 5*(-5:6))

plot(dates, prec, type='l', axes=FALSE, xlab='')
box(); axis(2); axis(4)
abline(v=pd, h=10*(1:4), lty=3, col=gray(0.5))

par(mar=c(3, 3, 0.1, 2), new=FALSE)
plot(dates, wind, type='l', axes=FALSE, xlab='',
     ylim=range(wind, wmax, na.rm=TRUE))
lines(dates, wmax, col=2)
box(); axis(2); axis(4)
abline(v=pd, h=5*(1:3), lty=3, col=gray(0.5))
axis(1, pd, months(pd, TRUE))
```

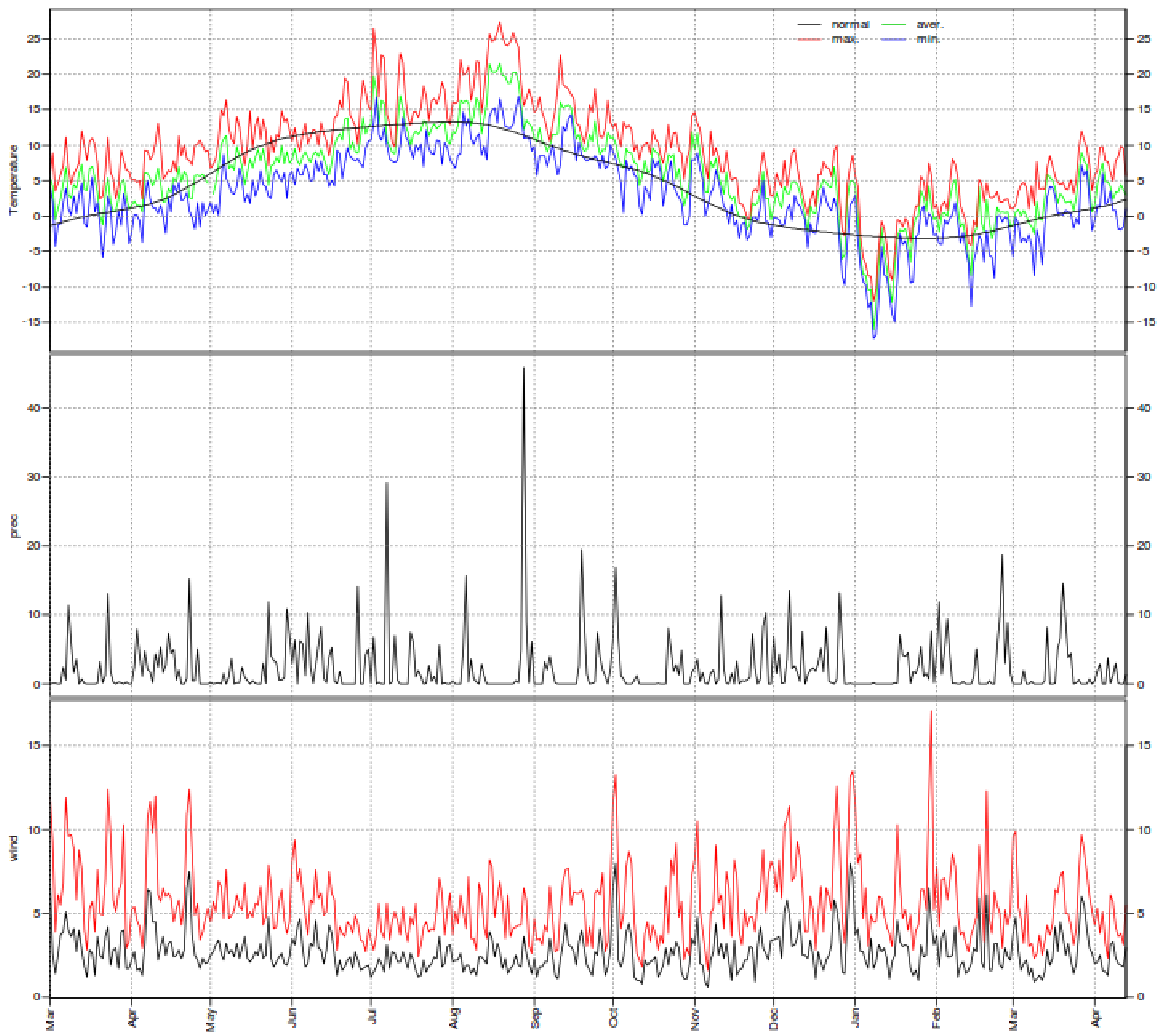

Figure 5: plot of chunk visualize3

We can have a look at the difference between the daily mean temperature and the normal temperature.

```
par(mar=c(3, 3, 0.1, 2), mgp=c(2,0.7,0), las=2, xaxs='i')
plot(dates, tmed-tnormal, type='l', axes=FALSE,
     xlab='', ylab='Deviation from the normal temperature')
box(); axis(2); axis(4)
abline(h=5*(-2:2), v=pd, lty=2, col=gray(0.5))
axis(1, pd, months(pd, TRUE))
```

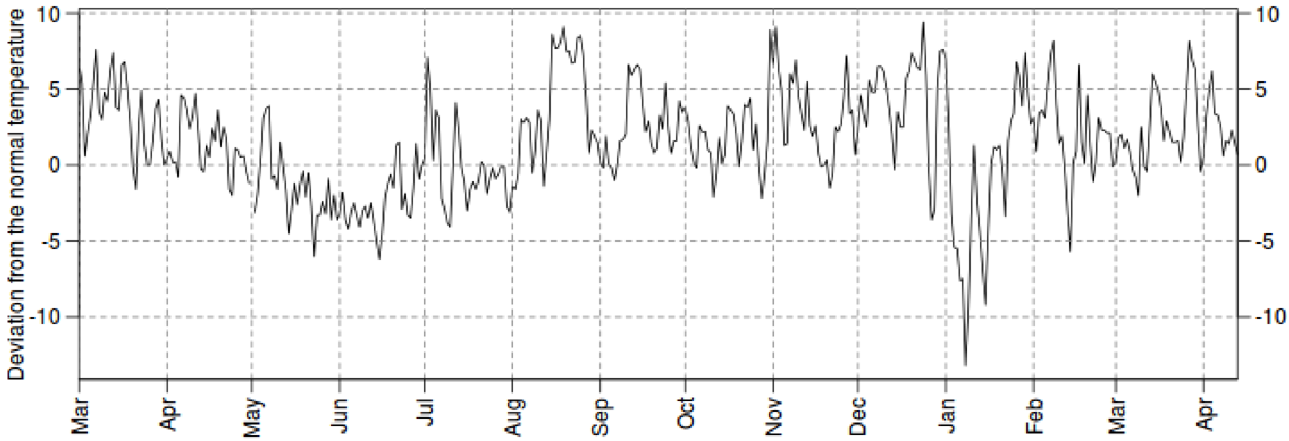

Figure 6: plot of chunk anomalia