

Tratamiento de modelos paramétricos de terremotos

Maribel Borrajo García



Departamento de Análise Matemática, Estatística e Optimización

Universidad de Santiago de Compostela



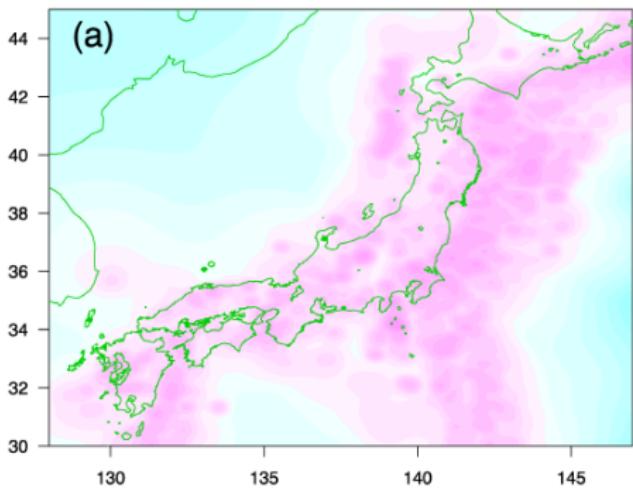
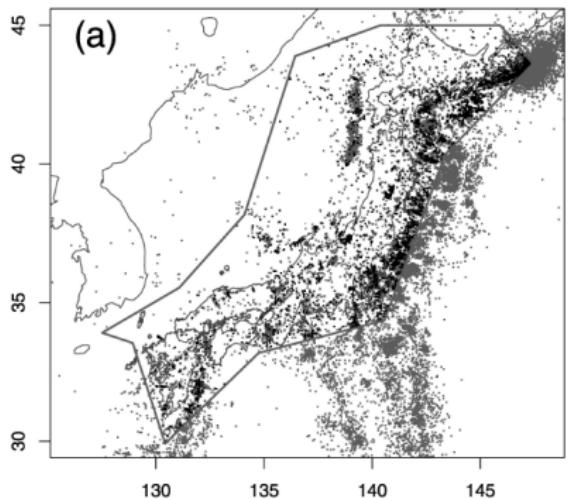
Como se estudan os terremotos?

Analizan a **estrutura xeométrica** de patróns formados por obxetos (**eventos**) distribuidos en unha, dúas ou tres dimensíons

Analizan a **estrutura xeométrica** de patróns formados por obxetos (**eventos**) distribuidos en unha, dúas ou tres dimensíóns

- A localización dos eventos é aleatoria
- O número de eventos tamén é aleatorio
- Dominio acotado
- Información adicional: marcas ou covariables

A función de intensidade computa o **número medio de eventos por unidad de área (volume)**





- Como son os datos de terremotos? $\longrightarrow (x_i, y_i, m_i, t_i)$



- Como son os datos de terremotos? $\longrightarrow (x_i, y_i, m_i, t_i)$

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{i:t_i < t} k(m_i)g(t - t')f(x - x_i, y - y_i, m_i)$$



- Como son os datos de terremotos? $\rightarrow (x_i, y_i, m_i, t_i)$

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{i:t_i < t} k(m_i)g(t - t')f(x - x_i, y - y_i, m_i)$$

- **Magnitudes:** $k(m) = Ae^{\alpha(m-m_c)}$, $m \geq m_c$,



- Como son os datos de terremotos? $\longrightarrow (x_i, y_i, m_i, t_i)$

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{i:t_i < t} k(m_i)g(t - t')f(x - x_i, y - y_i, m_i)$$

- **Magnitudes:** $k(m) = Ae^{\alpha(m-m_c)}$, $m \geq m_c$,
- **Background (xeración 0):** Poisson homoxéneo (μ)



- Como son os datos de terremotos? $\longrightarrow (x_i, y_i, m_i, t_i)$

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{i:t_i < t} k(m_i)g(t - t')f(x - x_i, y - y_i, m_i)$$

- **Magnitudes:** $k(m) = Ae^{\alpha(m-m_c)}$, $m \geq m_c$,
- **Background (xeración 0):** Poisson homoxéneo (μ)
- **Compoñente temporal:** $g(t) = \frac{p-1}{c} \left(1 + \frac{t}{c}\right)^{-p}$, $t > 0$,



- Como son os datos de terremotos? $\rightarrow (x_i, y_i, m_i, t_i)$

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{i: t_i < t} k(m_i)g(t - t')f(x - x_i, y - y_i, m_i)$$

- **Magnitudes:** $k(m) = Ae^{\alpha(m-m_c)}$, $m \geq m_c$,
- **Background (xeración 0):** Poisson homoxéneo (μ)
- **Compoñente temporal:** $g(t) = \frac{p-1}{c} \left(1 + \frac{t}{c}\right)^{-p}$, $t > 0$,
- **Compoñente espacial:**

$$f(x, y, m) = \frac{q-1}{\pi De^{\gamma(m-m_c)}} \left[1 + \frac{(x-x_i, y-y_i)S(x-x_i, y-y_i)^t}{De^{\gamma(m-m_c)}} \right]^{-q}$$



- Como son os datos de terremotos? $\rightarrow (x_i, y_i, m_i, t_i)$

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{i:t_i < t} k(m_i)g(t - t')f(x - x_i, y - y_i, m_i)$$

- **Magnitudes:** $k(m) = Ae^{\alpha(m-m_c)}$, $m \geq m_c$,
- **Background (xeración 0):** Poisson homoxéneo (μ)
- **Compoñente temporal:** $g(t) = \frac{p-1}{c} \left(1 + \frac{t}{c}\right)^{-p}$, $t > 0$,
- **Compoñente espacial:**

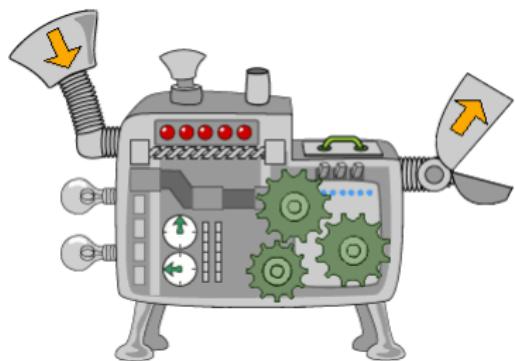
$$f(x, y, m) = \frac{q-1}{\pi De^{\gamma(m-m_c)}} \left[1 + \frac{(x-x_i, y-y_i)S(x-x_i, y-y_i)^t}{De^{\gamma(m-m_c)}} \right]^{-q}$$

¡Moitos parámetros: $\mu, A, \alpha, c, p, D, q, \gamma, \beta, \theta, \lambda_1!$

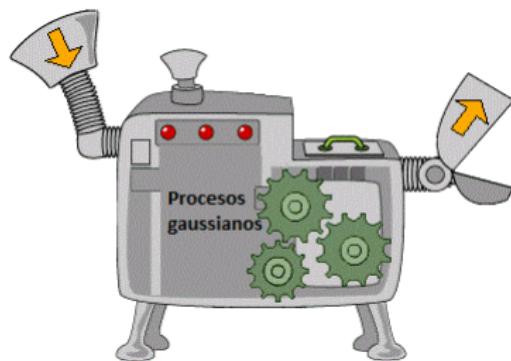


Son **modelos estatísticos** que, usando procesos gaussianos, proporcionan unha **versión simplificada e abordable** dun problema máis complexo

Son **modelos estatísticos** que, usando procesos gaussianos, proporcionan unha **versión simplificada e abordable** dun problema máis complexo



Simulador



Emulador

- **LHS:** Latin Hypercube Sampling
- **MICE:** Mutual Information for Computer Experiments



BECK, J. E GUILLAS, S. (2015) Sequential design with Mutual Information for Computer Experiments (MICE): emulation of a tsunami model. SIAM/ASA Journal on Uncertainty Quantification, SIAM, 4, pp. 739-766

- **LHS:** Latin Hypercube Sampling

- Usamos o paquete `lhs`
- `maximinLHS(n, k=11)` no cubo unidade
- Reescalamos segundo os rangos dos parámetros

- **MICE:** Mutual Information for Computer Experiments



BECK, J. E GUILLAS, S. (2015) Sequential design with Mutual Information for Computer Experiments (MICE): emulation of a tsunami model. *SIAM/ASA Journal on Uncertainty Quantification*, SIAM, 4, pp. 739-766

- **LHS:** Latin Hypercube Sampling

- Usamos o paquete `lhs`
- `maximinLHS(n, k=11)` no cubo unidade
- Reescalamos segundo os rangos dos parámetros

- **MICE:** Mutual Information for Computer Experiments

- Código dispoñible en Python
- Non hai paquete en R
- Linkamos os dous softwares ou adaptamos o código



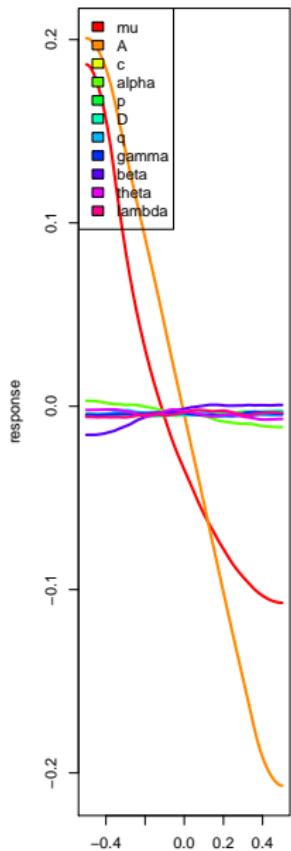
BECK, J. E GUILLAS, S. (2015) Sequential design with Mutual Information for Computer Experiments (MICE): emulation of a tsunami model. SIAM/ASA Journal on Uncertainty Quantification, SIAM, 4, pp. 739-766



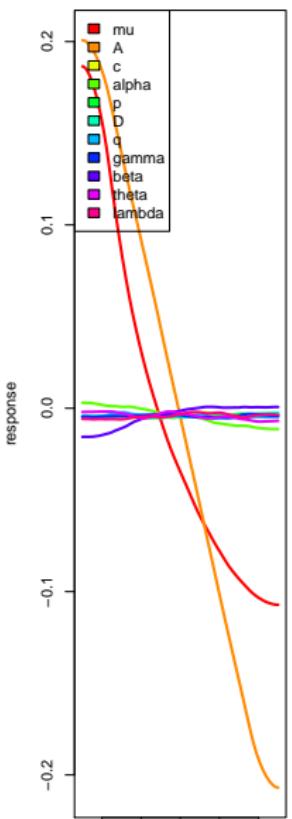
```
input<-res_simu
X<-input[,3:13]
Z<-input[,15]

model.t<-btgp(X=X, Z=Z, bprior = "b0", corr = "expsep",
BTE = c(500, 2000, 1), R = 1, m0r1 = TRUE, itemps = NULL,
pred.n = TRUE, krige = TRUE, zcov = FALSE, Ds2x = FALSE,
improv = FALSE, sens.p = NULL, nu = 1.5, trace = FALSE,
verb = 4,tree = c(0.5, 2))

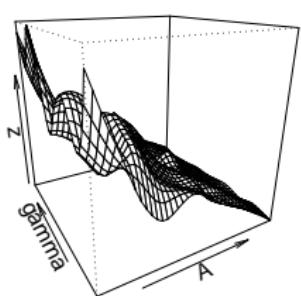
sf.t=sens(X=X,Z=Z, nn.lhs=500, model=btgp, verb=0)
```

Main Effects

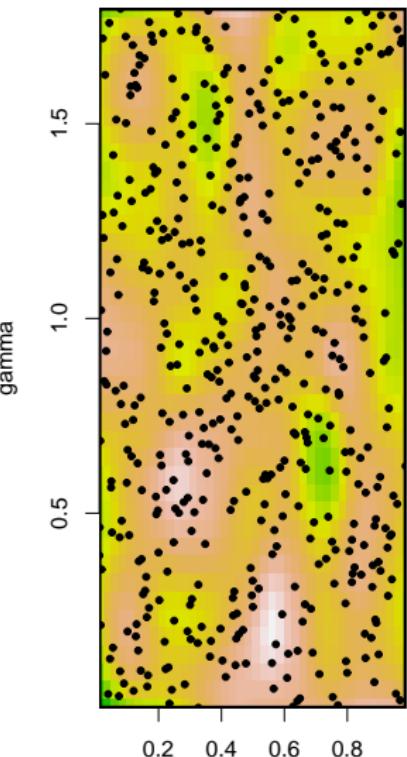
Main Effects



z mean



z quantile diff (error)





-  AKIMA, H. e GEBHARDT, A. (2015). akima: Interpolation of Irregularly and Regularly Spaced Data. R package version 0.5-12. URL: <http://CRAN.R-project.org/package=akima>
-  BADDELEY, A. e ROLF TURNER (2005). spatstat: An R Package for Analyzing Spatial Point Patterns. *Journal of Statistical Software*, 12(6), 1-42. URL: <http://www.jstatsoft.org/v12/i06/>
-  CARNELL, R. (2016). lhs: Latin Hypercube Samples. R package version 0.13. URL: <http://CRAN.R-project.org/package=lhs>
-  GRAMACY, R.R. (2007). tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models. *Journal of Statistical Software*, 19(9), 1-46. URL <http://www.jstatsoft.org/v19/i09/>

GRAZIAS





- **Background:**

```
background.simu<-function(mu,beta,xlim,ylim,tlim){  
  ### Poisson Process on Latitude and Longitude and Time ###  
  ## Time and space windows ##  
  xr=max(xlim)-min(xlim)  
  yr=max(ylim)-min(ylim)  
  tr=max(tlim)-min(tlim)  
  
  pp<-rpoispp3(mu, domain=box3(xrange=xlim, yrange=ylim,  
  zrange=tlim))  
  
  mags<-rexp(dim(pp$data)[1],beta)  
  simu.bgcata<-cbind(x=pp$data$x,y=pp$data$y,mag=mags,  
  t=pp$data$z)  
  simu.bgcata<-simu.bgcata[order(simu.bgcata [,4]),]  
  return(simu.bgcata<-simu.bgcata)}
```

Voltar

- **Compoñenete temporal:** método da inversión

```
F_1=function(t){para$cc*exp((log(1-t,base=exp(1))/(1-para$p))-1)}  
#F^{-1} function for the density g of the time interval between  
#a child and its parent  
res.T=0;count.T=0  
while(count.T<sum(Noff)){  
a=runif(1)  
if(F_1(a)<=tlim[2]){res.T=c(res.T,a);count.T=count.T+1}  
else{count.T=count.T}  
res.T=res.T[-1]  
offspring.T <- rep(start.cata[,4],Noff)  
offspring.T <- offspring.T + res.T}
```

[Voltar](#)

- **Compoñente espacial:** aceptación-rexeitamento

```
f=function(x,y){  
#for a straight line fault  
aux=(x^2*para$lambda+y^2*(para$lambda/2))*cos(para$theta) +  
x*y*sin(para$theta)*(para$lambda-(para$lambda/2))  
return(((para$q-1)/(pi*para$d*exp(para$gamma*m)))*  
*(1+aux/(para$d*exp(para$gamma*m)))^(-para$q))}  
  
res.X=0;res.Y=0  
for (k in 1:dim(start.cata)[1]) {  
m=start.cata[k,3]  
aux2=reject.sample.2d(Noff[k],f,5,xlim=c(-2m,2m),ylim=c(-2m,2m))  
aux.X=aux2$x; aux.Y=aux2$y  
res.X=c(res.X,aux.X)  
res.Y=c(res.Y,aux.Y)}  
res.X=res.X[-1];res.Y=res.Y[-1]  
offsp.X<-rep(start.cata[,1],Noff); offsp.X<-offsp.X+res.X  
offsp.Y<-rep(start.cata[,2],Noff); offsp.Y<-offsp.Y+res.Y
```

[Voltar](#)