



Mixed Non-Parametric and Parametric Estimation Techniques in R Package `etasFLP` for Earthquakes' Description

Marcello Chiodi
University of Palermo

Giada Adelfio
University of Palermo

Abstract

`etasFLP` is an R package which fits an epidemic type aftershock sequence (ETAS) model to an earthquake catalog; non-parametric background seismicity can be estimated through a forward predictive likelihood approach, while parametric components of triggered seismicity are estimated through maximum likelihood; estimation steps are alternated until convergence is obtained and for each event the probability of being a background event is estimated. The package includes options which allow its wide use. Methods for `plot`, `summary` and `profile` are defined for the main output class object. The paper provides examples of the package's use with description of the underlying R and Fortran routines.

Keywords: `etasFLP`, R, Fortran, point process, ETAS, earthquakes.

1. Introduction

A reliable estimation of the conditional intensity function is crucial in the description of seismic data through a space time point process. Often the conditional intensity function presents a superposition of a parametric component over a non-parametric component, e.g., in the epidemic type aftershock sequence (ETAS) model (Ogata 1988), widely used in statistical seismology. Indeed if we want to predict large earthquakes in presence of clusters of aftershocks, these may complicate the statistical analysis of the background seismic activity (Adelfio, Chiodi, De Luca, Luzio, and Vitale 2006) so it could be useful to study the features of independent events separately from the study of the strongly correlated ones in order to describe the seismicity of an area in space, time and magnitude domains.

Zhuang, Ogata, and Vere-Jones (2002) proposed a stochastic method (also implemented as

an option in the package **etasFLP**) which associates a probability that each event is either a background event or an offspring generated by other events, based on the ETAS model as in a latent class cluster model. The probabilities are then used to generate a thinned catalog.

However, the simultaneous estimation of non-parametric and parametric components is one of the main issues from a statistical point of view, as well as from a computational point of view. In the ETAS model, non-parametric components are associated mainly to background seismicity, while parametric ones are related to induced or triggered seismicity.

Adelfio, Chiodi, and Luzio (2010) presented a seismic sequences detection technique that identifies the conditional intensity function of a model describing the seismic activity as an ETAS model; Adelfio (2010a) used non-parametric methods to estimate the intensity function of a space-time point process and interpreted clustering results by a second-order diagnostic approach; see also Adelfio and Schoenberg (2009), as well as Adelfio and Chiodi (2009). Console, Jackson, and Kagan (2010) proposed a stochastic method associating to each event a probability to be either a background event or an offspring generated by other events; Marsan and Lengliné (2008) used the concept of cascade triggering without using models.

Adelfio and Chiodi (2013) proposed a method for the simultaneous solution of these problems, with an optimal selection of the bandwidth for the estimation of background seismicity. The **etasFLP** package is mainly based on those ideas.

In Adelfio and Chiodi (2013, 2015a), we classified events according to their probability of being a background or an offspring event, as proposed by Zhuang *et al.* (2002), and then estimated the space-time intensity of the generating point process of the different components by mixing non-parametric and parametric approaches, applying a forward predictive likelihood estimation approach to semi-parametric models (Chiodi and Adelfio 2011; Adelfio and Chiodi 2015a). The probabilities of being a background event are used as weights in non-parametric intensity estimation of the background seismicity.

The package **ETAS** (Jalilian 2016) in the R environment (R Core Team 2016), provides a porting of the Fortran code by Jiancang Zhuang, to estimate the components of an ETAS model. However **ETAS** cannot be used to implement the mixed estimation approach proposed in this paper; also for this reason we created the package **etasFLP** (Chiodi and Adelfio 2017), which provides open source tools to estimate a wide class of the ETAS model, developed in the R environment. The package has various options which allow to deal with different models of the ETAS family, and also different kinds of estimation strategy.

In Section 2 we recall some formal definitions of point processes and of the ETAS model. In Section 3 we report the definition of the kernel estimator for intensity functions, then the basic method for non-parametric estimation is introduced in Section 4; the algorithm for simultaneous estimation of non-parametric and parametric components of an ETAS model used in **etasFLP** is exposed in Section 4.1. The structure and details of the code are explained in Section 5 with some application of the ETAS model in Section 6, while final remarks and future developments are presented in Section 7.

2. Branching point processes: The ETAS model

A point process is a random collection of points, each one representing the time and space coordinates of a single event. Let $Z^d = T \times S^{d-1}$ be a general d -dimensional closed region, with T the time domain and S^{d-1} a two or three dimensional space. The *conditional intensity*

function (Daley and Vere-Jones 2003) is the expected frequency of events in a space-time unit region, conditional on the history \mathcal{H}_t of the point process up to time t , i.e.,

$$\lambda(\mathbf{z}) = \lambda(t, \mathbf{s} | \mathcal{H}_t) = \lim_{\mathcal{L}(\Delta t), \mathcal{L}(\Delta \mathbf{s}) \rightarrow 0} \frac{\mathbb{E} [N([t, t + \Delta t] \times [\mathbf{s}, \mathbf{s} + \Delta \mathbf{s}] | \mathcal{H}_t)]}{\mathcal{L}(\Delta t) \mathcal{L}(\Delta \mathbf{s})},$$

where $\mathcal{L}(x)$ is the Lebesgue measure of x , $\Delta t, \Delta \mathbf{s}$ are time and space increments. Generally, intensities $\lambda(\mathbf{z})$ depend on some unknown parameter.

The ETAS model, introduced by Ogata (1988), is a space time branching process for earthquake description, widely used in seismological context. Seismic events are usually collected in seismic catalogs, that represent the time-space-magnitude dimensions of n observed events, where the i th event is identified by its occurrence time T_i , its hypocentral coordinates (x_i, y_i, z_i) and its magnitude m_i .

The ETAS conditional intensity function can be written as follows:

$$\lambda_{\theta}(\mathbf{s}, y | \mathcal{H}_t) = \mu f(\mathbf{s}) + \sum_{t_j < t} \nu_{\phi}(t - t_j, \mathbf{s} - \mathbf{s}_j | m_j), \quad (1)$$

where m_j is the magnitude of the j th event and

$$\nu_{\phi}(t - t_j, \mathbf{s} - \mathbf{s}_j | m_j) = g(t - t_j | m_j) \ell(x - x_j, y - y_j | m_j).$$

Therefore, in the ETAS model, background seismicity has a time stationary intensity $\mu f(\mathbf{s})$, with $f(\cdot)$ a density function, while the occurrence rate of aftershocks at time t , following the earthquake of time t_j and magnitude m_j , is described by the following parametric model:

$$g(t - t_j | m_j) = \frac{\kappa_0 e^{(\alpha - \gamma)(m_j - m_0)}}{(t - t_j + c)^p}, \quad \text{with } t > t_j,$$

where κ_0 is a constant which measures the *aftershocks productivity*, c and p are parameters of the modified Omori's law (Utsu 1961); p is useful in order to characterize the pattern of seismicity of the given region, indicating the decay rate of aftershocks in time.

For the spatial rate, conditioned to the magnitude of the generating event, the following distribution is often used:

$$\ell(x - x_j, y - y_j | m_j) = \left\{ \frac{(x - x_j)^2 + (y - y_j)^2}{e^{\gamma(m_j - m_0)}} + d \right\}^{-q}. \quad (2)$$

The last equation relates the occurrence rate of aftershocks to mainshock magnitude m_j , through parameters α, γ that measure the influence on the relative weight of each sequence; m_0 is the completeness threshold of magnitude i.e., the lower bound for which earthquakes with higher magnitude are surely recorded in the catalog; d and q are two parameters related to the spatial influence of mainshocks.

Finally, the following eight-parameters function gives the whole intensity function in a point of coordinates (x, y, t) , conditioned to the past history \mathcal{H}_t :

$$\lambda_{\theta}(x, y, t | \mathcal{H}_t) = \mu f(x, y) + \sum_{t_j < t} \frac{\kappa_0 e^{(\alpha - \gamma)(m_j - m_0)}}{(t - t_j + c)^p} \left\{ \frac{(x - x_j)^2 + (y - y_j)^2}{e^{\gamma(m_j - m_0)}} + d \right\}^{-q}. \quad (3)$$

In the package **etasFLP** we used this formulation of the model with an eight-components vector of parameters: $\boldsymbol{\theta} = (\mu, \kappa_0, c, p, \alpha, \gamma, d, q)$, so that in the package some simpler model can be fitted; for example a 7-parameters model can be fitted fixing $\gamma = 0$, as explained in Section 5.

$\boldsymbol{\theta}$ is usually estimated by the maximum likelihood (ML) approach, with nontrivial numerical techniques (Ogata 1998). A procedure to maximize the expected complete data log-likelihood function, based on the expectation-maximization algorithm, was introduced in Veen and Schoenberg (2008). In Adelfio and Ogata (2010) a naive likelihood cross-validation is optimized to obtain the bandwidth of the smoothing kernel used to estimate the intensity for earthquake occurrence of northern Japan.

Instead, the first component $f(\cdot)$ of model (1) is generally estimated by non-parametric techniques. In particular, in kernel-type approaches either a fixed (Vere-Jones 1992) or adaptive kernel smoothing (Zhuang *et al.* 2002) can be used. Zhuang *et al.* (2002) also estimated the probability for each event of being a background event ($\rho_i, i = 1, \dots, n$) in order to obtain a thinned catalog, that includes events with a bigger probability of being mainshock, so that for the thinned process spatial intensity is described by an inhomogeneous Poisson process. Package **etasFLP** gives the option to follow this approach.

The simultaneous estimation of the background intensity and the triggered intensity components of an Epidemic type model is a crucial statistical issue.

In our algorithm, according to Console *et al.* (2010), we use ρ_i as weights for the kernel estimation of the background seismicity to get a simultaneous estimate of the intensity components of the ETAS model (1).

The problem of the estimation of the non-parametric component is addressed in the following sections.

3. Kernel estimator for the intensity function

Given n observed events $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ in an r -dimensional closed region, the kernel estimator of the unknown density $f(\cdot)$ (Silverman 1986; Wand and Jones 1994) in a generic point $\mathbf{u} \in \mathbb{R}^r$ is:

$$\hat{f}_{\boldsymbol{\Sigma}}(\mathbf{u}) = \frac{\sum_{i=1}^n w_i K(\mathbf{u} - \mathbf{u}_i, \boldsymbol{\Sigma})}{\sum_{i=1}^n w_i}, \quad (4)$$

where $K(\cdot, \cdot)$ is a multivariate kernel function centered at observed points and $\boldsymbol{\Sigma}$ is a matrix of smoothing constants and w_i are weights. The $w_i, i = 1, \dots, n$, can be given by the estimated probability that the i th point belongs to the background. A common choice for $K(\cdot, \cdot)$ is the normal multivariate density; to take into account highly variable patterns in a space region, variable smoothing matrices $\boldsymbol{\Sigma}_i$ can be more suitable (Terrell and Scott 1992), but in package **etasFLP** we simply used a bivariate normal kernel for the spatial distribution of background seismicity, with two smoothing constants, the bandwidths h_x, h_y (now denoted with $\mathbf{h} = (h_x, h_y)$), which can be estimated according to the FLP approach, sketched in Section 4.

Adelfio (2010b) found a convenient method of displaying some second-order properties in a neighborhood of a selected point of the process, based on very general high-dimensional kernel estimator developed to define a more general version of the conditional intensity function of a multi-dimensional point process.

The usual maximization of the likelihood with respect to the smoothing parameters, as known, would produce bandwidths of length zero and degenerate intensities on the observed points, so we used another strategy of estimation for non-parametric components, which however uses a concept linked to likelihood and to time-dependent nature of data.

4. Forward predictive likelihood (FLP)

Suppose that in a space-time point process the intensity function $\lambda(\cdot)$ depends on a set of parameters $\boldsymbol{\psi}$. Let $\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\cdot)$ denote a generic estimator of $\lambda(\cdot)$, based on observations until t_k , that could include smoothing constants in a semi-parametric context.

Assume that a realization of the process is observed in the space region $\Omega_{\mathbf{s}}$ and the time interval $(T_0; T_{\max})$. The log-likelihood for the point process, given the k observed values \mathbf{z}_i and computed using the estimator $\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\cdot)$ is:

$$\log L(\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}); H_{t_k}) = \sum_{i=1}^k \log \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}_i) - \int_{T_0}^{T_{\max}} \int_{\Omega_{\mathbf{s}}} \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}) \, d\mathbf{s} \, dt. \quad (5)$$

In this package, we use the method proposed in [Chiodi and Adelfio \(2011\)](#), that measures the ability of the observations and estimates until t_k to give information on the next observation.

Let $\log L(\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}); H_{t_{k+1}})$ be the log-likelihood computed on the first $k+1$ observations, but using the estimates based on the first k ones, defined as:

$$\log L(\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}); H_{t_{k+1}}) = \sum_{i=1}^{k+1} \log \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}_i) - \int_{T_0}^{t_{k+1}} \int_{\Omega_{\mathbf{s}}} \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}) \, d\mathbf{s} \, dt \quad (6)$$

For example, in Equation 6, $\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}_{k+1})$ could be the intensity of the $(k+1)$ th point estimated by a kernel method using centers given by the previous k points.

Then, in order to measure the *predictive information* of the first k observations on the $(k+1)$ th, we define $\delta_{k,k+1}$, the difference between (6) and (5), as:

$$\begin{aligned} \delta_{k,k+1}(\boldsymbol{\psi}) &= \log L(\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}); H_{t_{k+1}}) - \log L(\hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}); H_{t_k}) = \\ &= \left(\sum_{i=1}^{k+1} \log \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}_i) - \int_{T_0}^{t_{k+1}} \int_{\Omega_{\mathbf{s}}} \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}) \, d\mathbf{s} \, dt \right) - \\ &\quad \left(\sum_{i=1}^k \log \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}_i) - \int_{T_0}^{t_k} \int_{\Omega_{\mathbf{s}}} \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}) \, d\mathbf{s} \, dt \right) = \\ &= \log \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}_{k+1}) - \int_{t_k}^{t_{k+1}} \int_{\Omega_{\mathbf{s}}} \hat{\lambda}_{\boldsymbol{\psi}; H_{t_k}}(\mathbf{z}) \, d\mathbf{s} \, dt. \end{aligned}$$

Therefore, given n observations, the target function is:

$$FLP_{k_1}(\boldsymbol{\psi}) \equiv \sum_{k=k_1}^{n-1} \delta_{k,k+1}(\boldsymbol{\psi}), \quad (7)$$

where k_1 is a fixed constant, for example $k_1 = \lfloor \frac{n}{2} \rfloor$.

In this paper, $\boldsymbol{\psi} = \mathbf{h} = (h_x, h_y)$, although in the more general kernel estimator in Equation 4 we could have $\boldsymbol{\psi} = \boldsymbol{\Sigma}$.

Algorithm 1 Algorithm for mixed estimation used by `etasclass()`

```

1: procedure ALGORITHM FOR MIXED ESTIMATION
    ▷ INITIALIZATION STEPS:
2:    $v \leftarrow 1$ ,
3:   fix  $v_{\max}$  the maximum number of iterations and  $\epsilon_{\max}$  the maximum difference between
   estimates in successive iterations;
4:    $\epsilon \leftarrow \epsilon_{\max}$ ;
5:   compute  $\hat{f}_{\mathbf{h}^{(0)}}(x, y)$ , the starting estimation of the background intensity, obtained by
   kernel estimators; default values for the bandwidth  $\mathbf{h}^{(0)}$  are obtained by the Silverman's
   rule;
6:   the choice of a starting value for  $\boldsymbol{\theta}$  is explained in Section 5.3.
7:   Initially the weights  $\rho_i$  are fixed to a common value, that is, in the first step no
   weighting is performed.
    ▷ END OF INITIALIZATION STEPS.
    ▷ BODY OF THE ALGORITHM:
8:   while  $(v < v_{\max}) \cap (\epsilon \geq \epsilon_{\max})$  do
9:     Get the ML estimator  $\hat{\boldsymbol{\theta}}^{(v)}$  of the parameters of the model  $\boldsymbol{\theta} = (\boldsymbol{\phi}, \mu)^\top$ , by means
   of the maximization of the whole likelihood (5) and compute the values:
10:     $\lambda_{\hat{\boldsymbol{\theta}}^{(v)}}(t_i, x_i, y_i | \mathcal{H}_{t_i}) \leftarrow \hat{\mu}^{(v)} \hat{f}_{\mathbf{h}^{(v-1)}}(x_i, y_i) + \sum_{t_j < t_i} \nu_{\hat{\boldsymbol{\phi}}^{(v)}}(t_i - t_j, x_i - x_j, y_i - y_j), \quad i =$ 
     $1, \dots, n.$ 
    On the basis of the estimated parameters, for each point of the data set estimate the
    weight:
11:     $\rho_i^{(v)} \leftarrow \frac{\hat{\mu}^{(v)} \hat{f}_{\mathbf{h}^{(v-1)}}(x_i, y_i)}{\lambda_{\hat{\boldsymbol{\theta}}^{(v)}}(t_i, x_i, y_i | \mathcal{H}_{t_i})}, \quad i = 1, \dots, n.$ 
    The values  $\rho_i^{(v)}$  are an estimation of the probability to belong to the background seismicity
    and are used as weights for the non-parametric estimation of the background intensity in
    (4), with  $w_i = \rho_i$ .
12:    Estimate the optimal bandwidths  $\mathbf{h} = (h_x, h_y)$  of the kernel estimator, through the
    FLP approach, that is maximizing (7) with respect to  $(h_x, h_y)$  and holding fixed triggered
    intensities  $\sum_{t_j < t_i} \nu_{\hat{\boldsymbol{\phi}}^{(v)}}(t_i - t_j, x_i - x_j, y_i - y_j)$ ,  $i = 1, \dots, n.$ 
13:    Update the estimation of the background intensity  $\hat{\mu} \hat{f}_{\mathbf{h}^{(v)}}(x_i, y_i)$ , through a
    weighted normal kernel estimator with bandwidths  $\hat{h}_x, \hat{h}_y$  and weights  $\rho_i^{(v)}$ ,  $i = 1, \dots, n.$ 
14:     $v \leftarrow v + 1$ 
15:     $\epsilon_1 \leftarrow \max_i |\hat{f}_{\mathbf{h}^{(v)}}(x_i, y_i) - \hat{f}_{\mathbf{h}^{(v-1)}}(x_i, y_i)|$ ,  $\epsilon_2 \leftarrow \max |\hat{\boldsymbol{\theta}}^{(v)} - \hat{\boldsymbol{\theta}}^{(v-1)}|$ ,  $\epsilon \leftarrow \max(\epsilon_1, \epsilon_2)$ ;
     $\epsilon$  is used to check for convergence
16:   end while
    ▷ END OF THE BODY OF THE ALGORITHM.
17: end procedure

```

4.1. The algorithm for mixed estimation

In this section we explain the general structure of the estimation algorithm, used in the R function `etasclass()`, which executes the estimation procedure in the more complete situation, which is an eight-parameters model, with space-dependent background seismicity,

declustering and weighting steps and bandwidth estimation through the FLP criterion of Equation 7.

In order to estimate the different components of the space-time point process ETAS with intensity (eq:etasfull), we describe the simultaneous estimation of non-parametric and parametric components of the ETAS model. In other words, we alternate the standard parametric likelihood method, to estimate the parameter θ of the offsprings component, with the FLP approach, used just to compute the smoothing parameters h of background intensity in (4). Further, the proposed mixed procedure estimates the probability ρ_i of each event to belong to one of the two model components and uses the ρ_i as weights in (4).

The main function of the package **etasFLP** is the R function `etasclass()` and is based on the following procedure. Given an earthquake catalog of n events, which occurred in a given space-time region, and the intensity given in Equation (3), the full iterative process of simultaneous estimation of the non-parametric and parametric components of the model, proceeds as given in Algorithm 1.

5. The R and Fortran implementation

The package in R is installed and loaded in the usual way, e.g.:

```
R> install.packages("etasFLP")
```

Once the package is installed, then it must be loaded through the usual:

```
R> library("etasFLP")
```

Two sample data sets are present in the package: a small sample catalog of 2,158 Italian earthquakes of magnitude at least 3.0 from year 2005 to year 2013 (`italycatalog`, see at INGV site; <http://www.ingv.it/>) and a catalog of 18,545 Northern California earthquakes of magnitude at least 3.0 from year 1968 to year 2012 (`californiacatalog`, see at <http://www.ncedc.org/ncedc/>). They are included in the package **etasFLP** only to check and test the software: they have been downloaded without checking for completeness and without events relocation. The selected regions have unspecific seismological features.

```
R> data("italycatalog", package = "etasFLP")
R> data("californiacatalog", package = "etasFLP")
```

The structure of these data frames is the usual structure for a general catalog used as input for the function `etasclass()`. Any catalog *must* contain at least five variables with names `time`, `lat`, `long`, `z`, `magn1`:

```
R> str(italycatalog)

'data.frame':  2158 obs. of  5 variables:
 $ time : num  3.59e+09 3.59e+09 3.59e+09 3.59e+09 3.59e+09 ...
 $ lat  : num  38.8 37.7 44 46.2 44.8 ...
 $ long : num  15.72 13.86 7.53 12.49 8.98 ...
 $ z    : num  211.2 1 9.2 5.8 10.4 ...
 $ magn1: num  3.1 3.1 3 3.2 3.1 3.1 3.4 3.2 3.4 3.4 ...
```

```
R> str(californiacatalog)
```

```
'data.frame':  18545 obs. of  5 variables:
 $ time : num  2.15e+09 2.15e+09 2.15e+09 2.15e+09 2.15e+09 ...
 $ lat  : num  36.6 37.2 37.2 36.8 37.3 ...
 $ long : num  -121 -122 -122 -122 -122 ...
 $ z    : num   6.84 8.49 6.95 5.35 4.39 ...
 $ magn1: num   3 3 3.8 3 3 4.3 3.2 3 3.42 3.14 ...
```

5.1. Structure of the code

The code is written mainly in R, in order to use an easy accessible open source system and to use the input/output facilities. However some time consuming routines are written in good old Fortran, especially for computations hard to be optimized in R.

5.2. List of functions and subroutines

The main R functions are:

```
etasclass(), plot.etasclass(), print.etasclass(), plot.profile.etasclass(),  
print.profile.etasclass(), profile.etasclass().
```

Other auxiliary R functions are:

```
b.guten(), bwd.nrd(), eqcat(), kde2d.integral(), kde2dnew.fortran(),  
magn.plot(), MLA.freq(), plot.eqcat(), summary.eqcat(), time2date(),  
timechar.to.seq(), timecharunique2seq(), xy.grid().
```

Not all functions are documented in the package; other R functions are intended for internal use only. Fortran code is present in the source code in the file `fortran_etas_code.f90` in the directory `src`.

5.3. The function `etasclass()`

The main function of the package **etasFLP** is the R function `etasclass()` which performs the optimization steps of the algorithm described in Section 4.1. Some options are present, which allow to fit a variety of models with different kind of estimation procedures. Details on input arguments are given in the help files of the package.

```
etasclass(cat.orig,  
  magn.threshold = 2.5, magn.threshold.back = magn.threshold + 2,  
  mu = 1, k0 = 1, c = 0.5, p = 1.01, a = 1.2, gamma = .5, d = 1., q = 1.5,  
  params.ind = replicate(8, TRUE), declustering = TRUE, hdef = c(1, 1),  
  thinning = FALSE, flp = TRUE, ndeclust = 5, onlytime = FALSE,  
  w = replicate(nrow(cat.orig), 1), is.backconstant = FALSE,  
  description = "", cat.back = NULL, back.smooth = 1.0,  
  sectoday = TRUE, longlat.to.km=TRUE, usenlm = TRUE, method = "BFGS",  
  compsqm = TRUE, epsmax = 0.0001, iterlim = 100, ntheta = 100)
```


The input arguments of `etasclass` are:

- `cat.orig`: The only compulsory argument, an earthquake catalog, a `data.frame` with variables of names `time`, `lat`, `long`, `z`, `magn1`. The space time region is the rectangular area embedding the catalog.
- `magn.threshold`: Threshold magnitude (events of magnitude at least `magn.threshold` will be selected). Default value = 2.5. Some idea on the magnitude threshold of the catalog can be obtained using the function `magn.plot()`, which plots the logarithm of the back cumulated frequencies of the magnitudes; a change of slope in this graph is an indication of a magnitude threshold. However in the first trials on a new catalog, a good strategy is to begin with a high value of `magn.threshold`.
- `magn.threshold.back`: Threshold magnitude used to build the catalog `cat.back` for the first estimation of the background seismicity. Default value = `magn.threshold + 2`. This is only a simple technique to obtain a first estimate of the space background seismicity (namely, $\hat{f}_{h^{(0)}}(x, y)$ in the algorithm in Section 4.1).
- `mu`, `k0`, `c`, `p`, `a`, `gamma`, `d`, `q`: Values of the eighth-parameters of the ETAS model. They will be used as starting values or as fixed values according to the value of `params.ind`.
- `params.ind`: Vector of 8 indicators; `params.ind[i] = TRUE` means that the i th parameter must be estimated. `params.ind[i] = FALSE` means that the i th parameter is fixed to its input value (the order of parameters is: `mu`, `k0`, `c`, `p`, `a`, `gamma`, `d`, `q`). Default value = `replicate(8, TRUE)`, that is, `etasclass()` estimates all parameters.

Then three flags follow and an integer which regulates the kind of declustering, namely `declustering`, `thinning`, `flp` and `ndeclust`:

- `hdef`: Starting values for the x, y bandwidths used in the kernel estimator of background seismicity. Default value = 1,1.
- `declustering`: If `TRUE` the catalog is iteratively declustered for a maximum of `ndeclust` steps to optimally estimate the background intensity. Default value = `TRUE`.
- `thinning`: If `thinning = TRUE` a background catalog is obtained sampling from the original catalog with probabilities estimated during the iterations. Default value = `FALSE` and in this case declustering is made by weighting with weights ρ_i .
- `flp`: If `flp = TRUE` then background seismicity is estimated through forward predictive likelihood (described in Section 4). Otherwise Silverman's rule is used. Default value = `TRUE`.
- `ndeclust`: Maximum number of iterations for the general procedure, that is v_{\max} in the algorithm of Section (4.1). Default value = 5. A value of 3 can be however good enough to have reliable estimates.
- `w`: initial weights

Other control parameters:

- `onlytime`: If `TRUE` then a time process is fitted to data, regardless to space location.
- `is.backconstant`: If `TRUE` then the background seismicity is assumed to be homogeneous in space (and `declustering`, `flp` are set to `FALSE`).
- `description`: A description string used for the output. Default value = "".
- `cat.back`: Catalog used for the estimation of the background seismicity. Default value = `NULL`.
- `back.smooth`: Controls the level of smoothing for the background seismicity . Default value = 1. Meaningful only if `flp = FALSE`; in this case if h_S is the Silverman's rule bandwidth, the actual bandwidth will be $h = \text{back.smooth} \times h_S$.
- `sectoday`: If `TRUE`, then `time` variable of the input catalog is converted from seconds to days. Default value = `TRUE`.
- `longlat.to.km`: If `TRUE`, then `long` and `lat` variables of `cat.orig` are treated as geographical coordinates and converted to kilometers. Default value = `TRUE`.
- `usenlm`: If `TRUE`, then the R function `nlm()` (Gauss-Newton method) is used in the maximum likelihood steps; if `FALSE`, then the R function `optim()` is used (with `method = "BFGS"`). Default value = `TRUE`.
- `method`: Method used by `optim`. Default value = `"BFGS"`; meaningful only if `usenlm = FALSE`.
- `compsqm`: If `TRUE`, then standard errors are computed. Default value = `TRUE`.
- `epsmax`: Maximum allowed difference between estimates in subsequent iterations. Default value = `10e-04`.
- `iterlim`: Maximum number of iterations in the maximum likelihood steps (used in `nlm()` or `optim()`). Default value = 100.
- `ntheta`: Number of divisions of the round angle, used in the approximation of the integral involved in the likelihood computation of the ETAS model, as explained in the following subsection. Default value = 100.

The ML step

The main call of the ML step can be tuned with the argument `usenlm`: If this argument is set to the default value of `TRUE`, then the maximization of the likelihood is performed through a call to the standard Newton-type optimization function `nlm()`, present in the `stats` package of R. If problems are experienced with this call, then with `usenlm = FALSE` the function `optim()`, of R package `stats`, is used with the option `method = "BFGS"`, which performs a quasi-Newton optimization. In general `nlm()` is preferred for efficiency and `optim()` should be used only if there are problems with convergence. Here we report the piece of code with this optimization call:

```

if (usenlm) {
  result <- nlm(etas.mod2, params, hessian = TRUE, ...)
  ...
  params.optim <- result$estimate
  l.optim <- result$minimum
} else {
  result <- optim(params, etas.mod2, method = "BFGS", hessian = TRUE, ...)
  ...
  params.optim <- result$par
  l.optim <- result$value
}

```

The function minimized is $-\log L$, according to Equation 5, where k is set to n in order to have the whole likelihood and is computed by the function `etas.mod2()`, which internally calls the Fortran subroutine `etasfull18()`, used for the computation of the triggered intensity.

In the space domain, the integral involved in the likelihood is solved through a transformation to polar coordinates centered on each observed point (Ogata 1988): The round angle is then divided in `ntheta` parts; the default value (`ntheta = 100`) is enough to obtain a very good approximation. Each of the $n \times ntheta$ slices is then integrated on the time domain. This integration is efficiently computed in `etas.mod2()` with few lines of R code, using the advantage and the speed of R for computation involving whole matrices.

Briefly, to solve the integral of the component in (2), putting the origin on the point x_j, y_j , transforming to polar coordinates (ρ, θ) we obtain (when $q \neq 1$):

$$\iint_{\Omega(x,y)} \left\{ \frac{x^2 + y^2}{e^{\gamma(m_j - m_0)}} + d \right\}^{-q} dx dy = \iint_{\Omega(\rho,\theta)} \rho \left\{ \frac{\rho^2}{e^{\gamma(m_j - m_0)}} + d \right\}^{-q} d\rho d\theta.$$

Now integration with respect to θ is substituted by a finite sum, dividing the round angle in n_θ slices. A good approximation of the integral is therefore given by:

$$\sum_{l=1}^{n_\theta} \frac{\Delta(\theta) e^{\gamma(m_j - m_0)}}{1 - q} \left[\left(d + \frac{a_{jl}^2}{q} \right)^{1-q} - d^{1-q} \right],$$

where $\Delta(\theta)$ is the angle of the region, and a_{jl} is the distance from the j th point to the border of the space region in the direction θ_l ($l = 1, 2, \dots, n_\theta$). This step can be memory and time consuming, since here a matrix ($n \times n_\theta$) with elements a_{jl} must be stored and managed. Elements a_{jl} are computed once in the initialization steps of `etasclass()`.

The integration on the time domain of the components $g(t - t_j | m_j)$ is straightforward, with a particular case when $p = 1$.

In the likelihood computation of the ML step, the value of the background intensity is kept fixed to the last value obtained, whichever method is used for the background. Also the weights ρ_i are kept fixed and passed as argument to `etas.mod2()`, together with the vector of the background densities.

The optimization is performed only with respect to the parameters with `params.ind[i] = TRUE`. The parameters are in the sequence: `params = (mu, k0, c, p, a, gamma, d, q)`. For example a model with a fixed value `p = 1` (and `params.ind[4] = FALSE`) can be estimated

and compared with the model where \mathbf{p} is estimated (`params.ind[4] = TRUE`). A similar result can be obtained, for a single parameter, using the `profile()` method.

The optimization is made with respect to the logarithm of parameters, in order to respect the positivity constraints for all parameters.

Choice of starting values

The optimization algorithm depends on the choice of initial values. Some default guess choice is performed inside the function for parameters without input starting values. For example if there is no input value for μ , the function `etasclass()` will start with $\mu = \frac{n}{2\Delta t}$. This approximation comes from the solution of:

$$\frac{\partial \log L}{\partial \mu} = 0, \quad \frac{\partial \log L}{\partial \kappa_0} = 0, \quad (8)$$

with respect to μ, κ_0 , supposing initially $\rho_i = \frac{1}{2}, i = 1, 2, \dots, n$.

Other reasonable values are given for other parameters, by the function `etas.starting` of the package. The `etas.starting` function provides, as output, starting values for the parameters, that can be used as input in `etasclass`. These are only guess values, based on an exploratory analysis of the input catalog and on some approximation, for example the one used in Schoenberg (2013) to approximate the integral in (5), when $\lambda(\cdot)$ is the intensity function of a space-time ETAS model. Initial value for \mathbf{c} is based on the empirical distribution of lag times between successive events. Starting value for \mathbf{d} is based on the empirical distribution of space distances between successive events. μ and κ_0 starting values are obtained solving the likelihood equations (8) using the approximation for the integral given in Schoenberg (2013); for the other parameters the initial rough guesses are: $\mathbf{p} = 1$, $\mathbf{q} = 2$, $\mathbf{a} = 1.5$, $\mathbf{gamma} = 0.5$. The weights ρ are initialized to 0.5

Although it is only an heuristic solution for a non trivial problem, in our experience, it seems to individuate satisfactory starting points.

In general, any previous ETAS model fitting to similar data can be useful as a starting point.

If convergence problems are experienced for a given catalog, a useful strategy can be to start with a high magnitude threshold value m_0 (that is, with a smaller catalog which contains only big earthquakes), and then using this first output as starting guess for a new execution of `etasclass()` with a lower magnitude threshold value m_0 .

In this trial executions declustering should be avoided (`declustering = FALSE`) or `ndeclust` however must be fixed to a small value, no more then 3.

The FLP step

The FLP step is performed only if `flp = TRUE`. In this case we must have `declustering = TRUE`. To maximize expression (7) with respect to the bandwidths (h_x, h_y) , we used the R function `flp1.etas.nlm()` which minimizes the function `flpkospace()` through the optimization function `nlm()`. `flpkospace()` substantially computes $-FLP$, defined in (7) and is an interface to the Fortran subroutine `deltafl1kospace()`.

The kernel density is computed by the R function `kde2dnew.fortran()`, which is a standard bivariate normal kernel estimator, substantially an interface to the Fortran subroutine `density2()`. The arguments of the function are:

- `xkern`, `ykern`: x -values, y -values of kernel points of length `n` (`n = length(xkern)`).
- `gx`, `gy`: x -values, y -values of the points where densities must be estimated.
- `h`: Bandwidths; a length 2 numerical vector.
- `factor.xy`: Expansion factor for bandwidths (density will be smoother if `factor.xy > 1`).
- `w`: Vector of weights to give to observed points (length `n`).
- `eps`: Enlargement factor for the region.

The used kernel weighted estimator is as in (4), with $w_i = \rho_i$ and Σ a diagonal matrix with elements h .

5.4. Object and methods defined

The object of class ‘etasclass’

An object of S3 class ‘etasclass’ is returned by the function `etasclass()` and contains information on the estimation procedure. The main items of the output are:

- `this.call`: The exact call that originated the output.
- `params.ind`: Indicates which parameters have been estimated.
- `params`: ML estimates of the ETAS parameters.
- `sqm`: Estimates of standard errors of the ML estimates of the ETAS parameters (`sqm[i] = 0` if `params.ind[i] = FALSE`).
- `AIC.iter`: AIC values at each iteration.
- `hdef`: Final bandwidths used for the kernel estimation of background spatial intensity (however estimated, with `flp = TRUE` or `flp = FALSE`).
- `rho.weights`: Estimated probability for each event to be a background event (ρ).
- `time.res`: Rescaled time residuals (for time processes only).
- `params.iter`: A matrix with the estimates values at each iteration.
- `sqm.iter`: A matrix with the estimates of the standard errors at each iteration.
- `rho.weights.iter`: A matrix with the values of `rho.weights` at each iteration.

Also the values of optional input arguments are returned in the output.

5.5. The output: Text and graphical output

plot and summary methods for object 'etasclass'

As usual in R, the output is managed through the standard S3 methods `print()` and `summary()` for textual output and the `plot()` method for default graphic output defined for objects of class 'etasclass', the standard output object of the function `etasclass()`.

The text output of `summary()` contains essential information on the catalog, on the ML estimates and their standard errors and on the general iterative estimation process, giving the sequence of AIC values.

The `plot` method has some specific optional argument (the first argument is a compulsory 'etasclass' object):

- `pdf`: If TRUE, then 2D plots are sent to a pdf file;
- `file`: Name of the pdf file;
- `ngrid`: Number of points for each direction (x, y) of a `ngrid` × `ngrid` space points grid, where estimated intensities must be evaluated. Default value = 200, which gives a good graphical resolution;
- `flag.3D`: Option of the `plot` method. If TRUE a 3D plot is also produced with estimated intensities for each observed point;
- `flag.log`: Option of the `plot` method. If TRUE then a log scale is used to plot intensities. The default is FALSE; of course the plot method can be called twice with two different scale options.

The computation of the grid of intensities has been left to the `plot` method to avoid too many computations in the main routine `etasclass()`. Three intensity plots are computed: the whole intensity, the background and the triggered ones.

Values of the triggered intensity on a space grid of values are efficiently computed through the Fortran subroutine `etasfull8tintegrated()`, which essentially computes the space intensities through an integration on the time domain of the components $g(t - t_j|m_j)$:

$$\lambda_{\text{triggered}}(x, y) = \sum_{j=1}^n \int_{t_j}^{T_{\max}} g(t - t_j|m_j) \ell(x - x_j, y - y_j|m_j) dt,$$

where (x, y) is one of the `ngrid` × `ngrid` points of the spatial grid. As in other parts of the package **etasFLP**, time consuming routines are written in Fortran.

In our package we use function `image.plot()`, from package **fields** (Nychka, Furrer, Paige, and Sain 2016), to plot the intensities on the maps (geographic maps are computed with the function `map()` of the package **maps**; Becker, Wilks, Brownrigg, and Minka 2016).

The values of the intensities on the grid are however given as output, so that users could use this `plot` method as a grid computing engine and can use anything else for graphical plotting (e.g., using some Google maps resources). Some more graphical diagnostic outputs and residual analysis tools are provided by the `plot` method, as described in Adelfio and Chiodi (2015b), and used also in Nicolis, Chiodi, and Adelfio (2015).

`profile()` and `plot()` methods for objects ‘`etasclass`’ and ‘`profile.etasclass`’

Approximate inference on a single parameter can be made through the `profile` method for ‘`etasclass`’ objects. Given the likelihood function $L(\boldsymbol{\theta})$ for the whole model, where the unconstrained ML estimate of the unknown parameter vector $\boldsymbol{\theta}$ is $\hat{\boldsymbol{\theta}}$ (of k components, $k \leq 8$), denote by θ_j the single parameter of interest for which the profile must be built. Let $\boldsymbol{\theta}_{(-j)}$ be the $k - 1$ components vector of the remaining parameters, and let $\hat{\boldsymbol{\theta}}_{(-j)}(\theta_j)$ be the ML estimate of $\boldsymbol{\theta}_{(-j)}$ corresponding to a fixed value of (θ_j) . The function $L[\theta_j, \hat{\boldsymbol{\theta}}_{(-j)}(\theta_j)]$ is the profile likelihood function for θ_j .

Indeed we used a rescaled version $(-2 \log(LR))$ of the profile likelihood, so that we define:

$$l_p(\theta_j) = -2[\log(L[\theta_j, \hat{\boldsymbol{\theta}}_{(-j)}(\theta_j)]) - \log(L(\hat{\boldsymbol{\theta}}))].$$

The study of this function versus θ_j is useful to make approximate inference on a single parameter; for example we can build the $1 - \alpha$ confidence interval for θ_j , selecting the interval for which $l_p(\theta_j) < \chi_{\alpha}^2$.

To avoid heavy computations, $l_p(\theta_j)$ is computed only for a few values of θ_j , and then a spline interpolation is used to have a reasonable sketch of the plot of the profile likelihood; the approximation will be in general quite good since the profile function is convex near the global solution and furthermore we know the global minimum value. This method is numerically convenient since each computation of a value $l_p(\theta_j)$ is computationally very expensive: A likelihood maximization problem must be solved, with $k - 1$ unknown components instead of the k components of the unconstrained problem. We will need only one ML step, since the non-parametric components will be the same of the full model.

The `profile` method needs an object of class ‘`etasclass`’ as input, in order to have a full model already estimated without constraints. The computation of the profile likelihood will leave unchanged all options used to obtain the ‘`etasclass`’ object, e.g., fixed parameters, weighting and declustering options, background seismicity.

The call of the `profile` method (with arguments and their default values) is:

```
profile(fitted, iprofile = 4, nprofile = 7, kprofile = 3,
       profile.approx = FALSE)
```

- `fitted`: An object of class ‘`etasclass`’.
- `iprofile`: An integer in the range 1–8. The profile likelihood will be computed with respect to the parameter of index `iprofile`. The order of parameters is: μ , κ_0 , c , p , α , γ , d , q .
- `nprofile`: The number of values of `params[iprofile]` for which the profile likelihood must be computed. Default value = 7. An odd number of grid points `nprofile` is advised, so that the central point is the unconstrained ML estimate for the profiled parameter, and the spline interpolation will have a better quality.
- `kprofile`: Maximum absolute standardized value for `params[iprofile]`.

The global ML estimates are in `m1 = fitted$params[profile]`, with asymptotic standard errors `sd = fitted$params.sqm`. The profile likelihood is computed in the range `(m1 - sd * kprofile, m1 + sd * kprofile)`. Default value = 3.

- `profile.approx`: If TRUE, then a conditional-likelihood approach is used to obtain a first value $\hat{\boldsymbol{\theta}}_{(-j)}^*(\theta_j)$ for each maximization step in the profile likelihood computation. Default value = FALSE. The formula used is:

$$\hat{\boldsymbol{\theta}}_{(-j)}^*(\theta_j) = \hat{\boldsymbol{\theta}}_{(-j)} - (\theta_j - \hat{\theta}_j) \left\{ \frac{\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_{(-j)} \partial \boldsymbol{\theta}_{(-j)}^T} \right\}^{-1} \frac{\partial^2 l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_{(-j)} \partial \theta_j}.$$

The `plot` method for an object of the class `'profile.etasclass'` is so defined:

```
plot(x, prob = c(0.9, 0.95, 0.99), ...)
```

- `x`: Is an object of the class `'profile.etasclass'` (the output of the `profile` method for `'etasclass'` objects).
- `prob`: Is a vector of coverage probability for the asymptotic confidence interval computed using $-2 \log(LR)$. Default value `prob = c(0.9, 0.95, 0.99)`.

In the `plot` method for `'profile.etasclass'` objects the profile likelihood is computed and plotted with a spline interpolation on `nprofile` points. Inverse spline interpolation is used to obtain approximate confidence intervals with coverage probabilities given as input in the vector `prob`.

6. Examples

The following examples have been executed on a PC with a Intel Core CPU i7-3770 (3.40GHz) with a RAM of 8Gb; we used the R version 3.2.2 (64-bit) running under Ubuntu 14.04 LTS.

6.1. Example with the Italian data

Let consider a sample earthquake catalog:

```
R> data("italycatalog", package = "etasFLP")
```

Assigning to the catalog the class `'eqcat'`, `summary()` and `plot()` methods are defined to have elementary descriptive functions on an earthquake catalog:

```
R> class(italycatalog) <- "eqcat"
R> summary(italycatalog)
R> plot(italycatalog)
```

On this sample earthquake catalog we can run a sample execution (many options are however fixed to their default values, and in the following example are assigned explicitly only to explain their meaning):

```
R> etas.flp <- etasclass(italycatalog, magn.threshold = 3.0,
+   magn.threshold.back = 3.5, k0 = 0.005, c = 0.005, p = 1.01, a = 1.05,
+   gamma = 0.6, q = 1.52, d = 1.1,
```



```
+   params.ind = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE),
+   declustering = TRUE, thinning = FALSE, flp = TRUE, ndeclust = 15,
+   onlytime = FALSE, is.backconstant = FALSE, description = "etas flp",
+   sectoday = TRUE, usenlm = TRUE, epsmax = 0.001)
```

The previous call of `etasclass()` fits a space time ETAS model to the catalog `italycatalog`, only for events with a minimum magnitude of `magn.threshold = 3.0`. The events with magnitude at least 3.5 (`magn.threshold.back = 3.5`) are used to obtain a catalog with bigger events: A default kernel estimator of the x, y density on this catalog will be the basis for the first approximation $\hat{f}_{h^{(0)}}(x, y)$.

The options `declustering = TRUE` and `ndeclust = 15` mean that the whole mixed procedure described in Section 4.1 will be used, with a maximum of 15 steps (the input value of `ndeclust`).

The option `thinning = FALSE` means that declustering is made with weighting and not with thinning.

`flp = TRUE` means that the forward predictive likelihood (Section 4) is used to estimate the bandwidths.

Indeed the full algorithm described in Section 4.1 is obtained with the options:

```
onlytime = FALSE, declustering = TRUE, thinning = FALSE, flp = TRUE
```

- `k0 = 0.005, c = 0.005, p = 1.01, a = 1.05, gamma = 0.6, q = 1.52, d = 1.1`: Initial values are specified for 7 parameters; initial value for μ will be computed inside the function.
- `params.ind = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE)`: All parameters will be estimated.
- `onlytime = FALSE`: Means that a space-time model will be fitted.
- `is.backconstant = FALSE`: The background intensity will not be constant in the space region.
- `sectoday = TRUE`: The variable time in catalog `italycatalog` is in seconds, and this option transforms it into days.
- `usenlm = TRUE`: The default `nlm()` function of R will be used in the ML step.
- `epsmax = 0.001`: Controls the convergence criterion of the declustering steps.

After the execution, we obtain the object `etas.flp` of class ‘`etasclass`’; a `summary` method can be applied to obtain the main results:

```
R> summary(etas.flp)
```

Call:

```
etasclass(cat.orig = italycatalog, magn.threshold = 3, magn.threshold.back =
```

```

3.5, k0 = 0.005, c = 0.005, p = 1.01, a = 1.05, gamma = 0.6, d = 1.1,
q = 1.52, params.ind = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
  TRUE, TRUE), declustering = TRUE, thinning = FALSE, flp = TRUE,
ndeclust = 15, onlytime = FALSE, is.backconstant = FALSE,
description = "etas flp", sectoday = TRUE, usenlm = TRUE,
epsmax = 0.001)

```

```

etas flp
Execution started:                2015-09-30 16:04:59
Elapsed time of execution (hours) 0.2404301
Number of observations            2158
Magnitude threshold              3
declustering                     TRUE
Number of declustering iterations 4
Kind of declustering             weighting
flp                              TRUE
sequence of AIC values for each iteration
49620.08 48458.86 48418.2 48415.17

```

ETAS Parameters:

	Estimates	std.err.
mu	0.355850	0.011294
k0	0.008373	0.002053
c	0.009404	0.001795
p	1.121630	0.016271
a	1.509371	0.064077
gamma	0.857945	0.084688
d	1.915139	0.306384
q	1.836391	0.067067

To empirically evaluate the advantage of the FLP approach for the bandwidth estimation, we report here the summary results for the same execution of `etasclass()` but with the option `flp = FALSE`.

```

R> etas.noflp <- etasclass(italycatalog, magn.threshold = 3.0,
+   magn.threshold.back = 3.5, k0 = 0.005, c = 0.005, p = 1.01, a = 1.05,
+   gamma = 0.6, q = 1.52, d = 1.1,
+   params.ind = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE),
+   declustering = TRUE, thinning = FALSE, flp = FALSE, ndeclust = 15,
+   onlytime = FALSE, is.backconstant = FALSE, description = "etas flp",
+   sectoday = TRUE, usenlm = TRUE, epsmax = 0.001)

```

Now `flp = FALSE` means that the Forward predictive likelihood (Section 4) is not used to estimate the bandwidth.

```
R> summary(etas.noflp)
```

```
Call:
```

```
etasclass(cat.orig = italycatalog, magn.threshold = 3, magn.threshold.back =
  3.5, k0 = 0.005, c = 0.005, p = 1.01, a = 1.05, gamma = 0.6, d = 1.1,
  q = 1.52, params.ind = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
    TRUE, TRUE), declustering = TRUE, thinning = FALSE, flp = FALSE,
  ndeclust = 15, onlytime = FALSE, is.backconstant = FALSE,
  description = "etas flp", sectoday = TRUE, usenlm = TRUE,
  epsmax = 0.001)
```

```
etas flp
```

```
Execution started:                2015-09-30 16:29:15
Elapsed time of execution (hours)  0.1860959
Number of observations             2158
Magnitude threshold                3
declustering                       TRUE
Number of declustering iterations  3
Kind of declustering               weighting
flp                                 FALSE
sequence of AIC values for each iteration
49620.08 49303.98 49295.08
```

```
-----
ETAS Parameters:
```

	Estimates	std.err.
mu	0.334571	0.011611
k0	0.006994	0.001481
c	0.006723	0.001321
p	1.070908	0.014472
a	1.408809	0.065284
gamma	0.763376	0.091095
d	1.872523	0.300733
q	1.688741	0.054613

We see that the execution for the simpler case (`flp = FALSE` with output `etas.noflp`), is much faster than the FLP case (`flp = TRUE` with output `etas.flp`): about 10 minutes against about 22 minutes, due essentially to the fact that this FLP execution needed 7 iterations for declustering instead of 3. However the recursive estimation of the background seismicity, with the FLP method, leads to a final AIC value much better than the case without FLP (48415.17 against 49285.08).

Furthermore we can see (from the whole structure of the object `etas.flp`) some more detail on the execution:

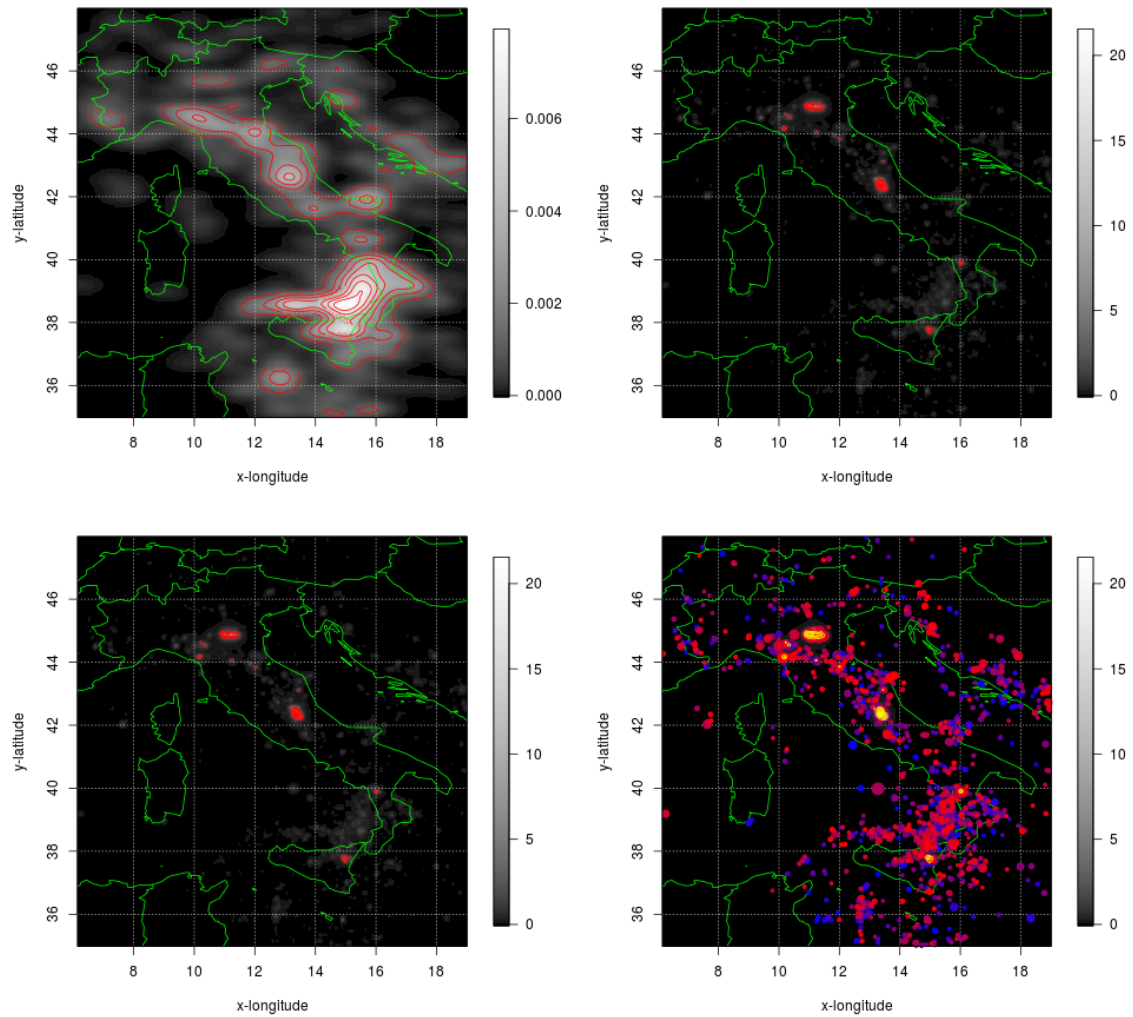


Figure 1: Output of the `plot` method for ‘`etasclass`’ objects: estimated background (top-left) intensity, triggered intensity (top-right), total intensity (bottom-left), and total intensity with observed points – circles areas proportional to magnitudes, red: recent, blu: older – (bottom-right) of the Italian seismicity (2005–2013, magnitude threshold 3.0), obtained with `etasclass()` choosing `flp = TRUE`.

```
R> round(etas.flp$params.iter, 4)
```

	mu	k0	c	p	a	gamma	d	q
1	0.3223	0.0063	0.0060	1.0551	1.3694	0.7220	1.8148	1.6290
2	0.3532	0.0083	0.0091	1.1163	1.4987	0.8489	1.9144	1.8223
3	0.3556	0.0084	0.0094	1.1211	1.5083	0.8578	1.9130	1.8348
4	0.3559	0.0084	0.0094	1.1216	1.5094	0.8579	1.9151	1.8364

This matrix contains the ML estimates of the 8 parameters of the ETAS model for each declustering iteration reported in each row. We can see that the values of the parameters estimates vary a certain amount for the first iterations, but then values remain almost unchanged.

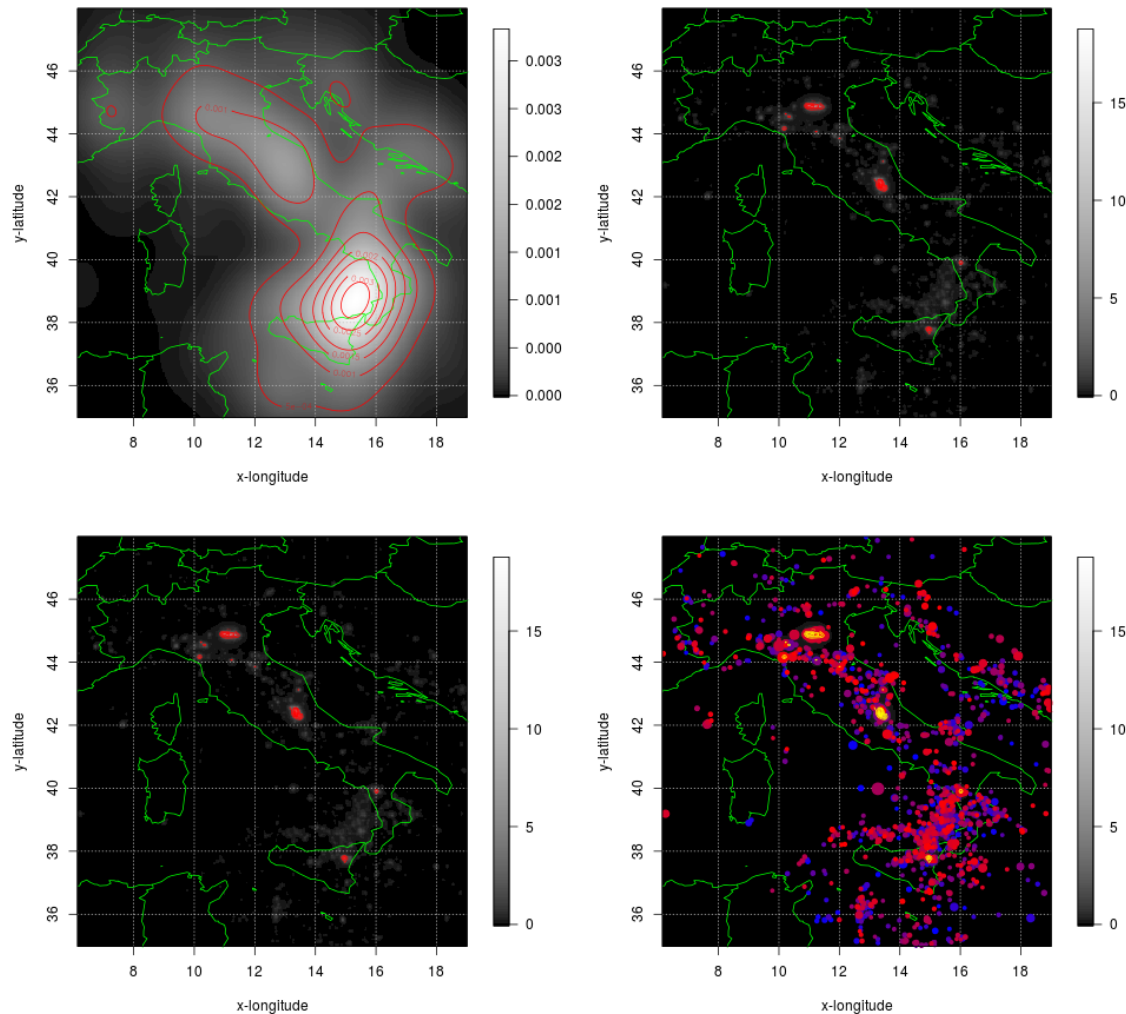


Figure 2: Output of the `plot` method for ‘`etasclass`’ objects: estimated background (top-left) intensity, triggered intensity (top-right), total intensity (bottom-left), and total intensity with observed points - circles areas proportional to magnitudes, red: recent, blu: older - (bottom-right) of the Italian seismicity (2005–2013, magnitude threshold 3.0), obtained with `etasclass()` choosing `flp = FALSE`.

To evaluate the information of the declustering and weighting process, we can examine the values of the weights ρ_i , $i = 1, 2, \dots, n$ reported in the vector `rho.weights`.

```
R> table(cut(etas.flp$rho.weights, seq(0, 1, by = 0.2)))
```

(0,0.2]	(0.2,0.4]	(0.4,0.6]	(0.6,0.8]	(0.8,1]
976	41	38	80	1023

A good situation is when the weights are all close to 0 or 1 (that is, each event is clearly a background event or a triggered one).

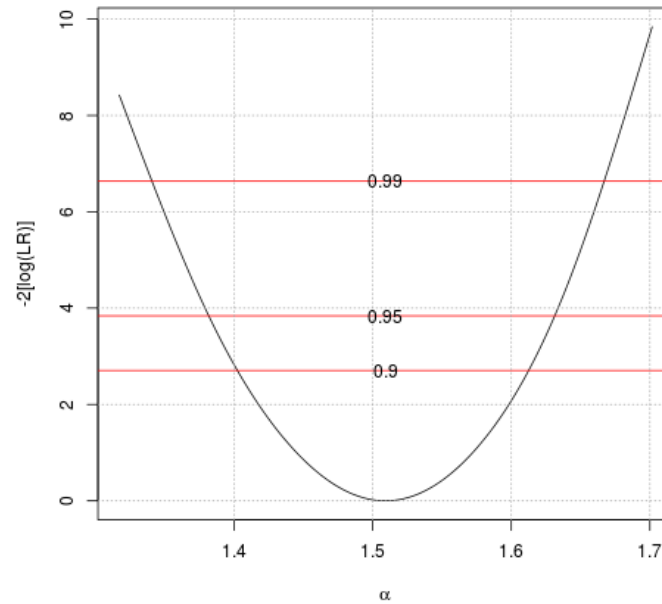


Figure 3: Profile likelihood for parameter α obtained with the `plot` method for ‘`profile.etasclass`’ objects, for the the Italian catalog.

In Figure 1 we plot the images of the background and triggered space intensities estimated by the FLP approach, obtained by:

```
R> plot(etas.flp)
```

In Figure 2 we plot the images of the background and triggered space intensities estimated without the FLP approach, obtained by:

```
R> plot(etas.noflp)
```

As an example, profile is computed for α using the object `etas.flp`:

```
R> prof.flp <- profile(etas.flp, nprofile = 7, iprofile = 5)
R> plot(prof.flp)
```

Asymptotic confidence intervals:

	Coverage	Lower	Upper
1	0.90	1.393	1.624
2	0.95	1.381	1.635
3	0.99	1.346	1.659

In Figure 3 we report the graphical output of the command `plot(prof.flp)`, which displays the profile likelihood (computed with `profile(etas.flp)`) of the parameter α , for the model estimated with the option `flp = TRUE`. The method used to plot the profile is explained in Section 5.5. Let’s have a look at the profile results: We obtained a 95% asymptotic confidence interval (based on the LR test) around α of (1.381–1.635); we can compare this value with the

classical asymptotic normal based interval ($\theta \mp z_{1-\alpha} s_{\theta}$) obtaining for α (the 5th parameter): $1.509371 \mp 1.96 \times 0.064077 = 1.384 - 1.635$, very close to the LR based interval.

6.2. An example with California data

Another execution with a different catalog, is shown with an application to the North California data:

```
R> data("californiacatalog", package = "etasFLP")
R> etas.flp.cal <- etasclass(californiacatalog, magn.threshold = 3.5,
+   magn.threshold.back = 5, declustering = TRUE, epsmax = 0.0001,
+   k0 = 0.005, c = 0.005, p = 1.01, a = 1.05, gamma = 1.7,
+   q = 1.52, d = 1.77,
+   params.ind = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE),
+   description = "etas flp:      California =>3.5 from 1968",
+   ndeclust = 25)
```

We selected events with a magnitude of at least 3.5 (`magn.threshold = 3.5`). Here we used many default choices, so `flp = TRUE` and `thinning = FALSE`. All the eight parameters of the ETAS model will be estimated. The summary of the results is reported below.

```
R> summary(etas.flp.cal)
```

Call:

```
etasclass(cat.orig = californiacatalog, magn.threshold = 3.5,
  magn.threshold.back = 5, k0 = 0.005, c = 0.005, p = 1.01,
  a = 1.05, gamma = 1.7, d = 1.77, q = 1.52, params.ind = c(TRUE,
    TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE), declustering = TRUE,
  ndeclust = 25, description = "etas flp:      California =>3.5 from 1968",
  epsmax = 1e-04)
```

```
etas flp:      California =>3.5 from 1968
Execution started:      2015-09-30 17:15:35
Elapsed time of execution (hours)  4.063476
Number of observations      6141
Magnitude threshold      3.5
declustering      TRUE
Number of declustering iterations  9
Kind of declustering      weighting
flp      TRUE
sequence of AIC values for each iteration
128763.2 127278.7 127093.9 126996.1 126957 126941.5 126935.3 126933.5
126932.9
```

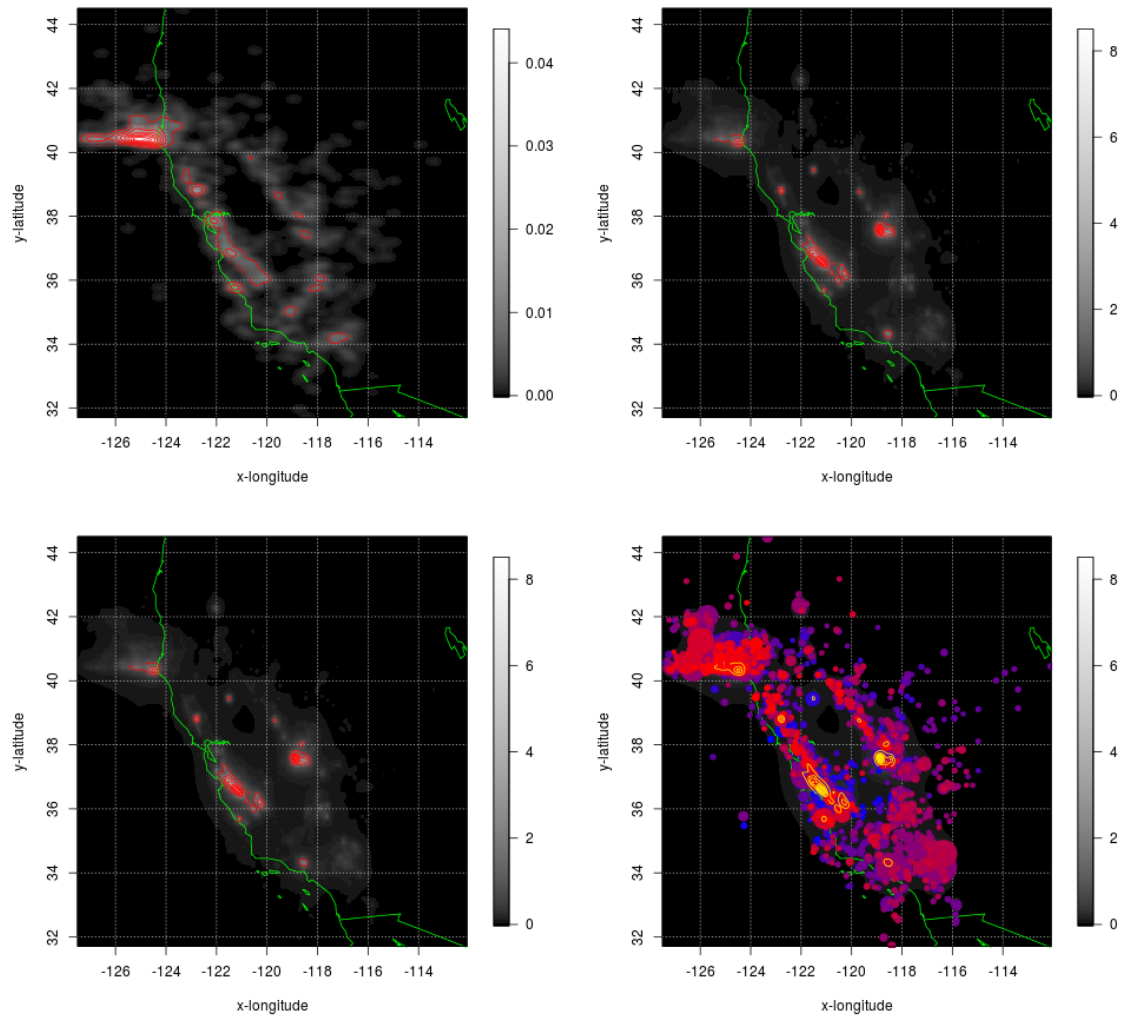


Figure 4: Output of the `plot` method for ‘`etasclass`’ objects: estimated background (top-left) intensity, triggered intensity (top-right), total intensity (bottom-left), and total intensity with observed points – circles areas proportional to magnitudes, red: recent, blu: older – (bottom-right) of the North California seismicity (1968–2012, magnitude threshold 3.5), obtained with `etasclass()` choosing `flp = TRUE`.

ETAS Parameters:

	Estimates	std.err.
<code>mu</code>	0.086153	0.002798
<code>k0</code>	0.001873	0.000093
<code>c</code>	0.004057	0.000468
<code>p</code>	0.997162	0.004836
<code>a</code>	0.973744	0.026071
<code>gamma</code>	1.640199	0.056597
<code>d</code>	0.159074	0.013577
<code>q</code>	1.396250	0.013165

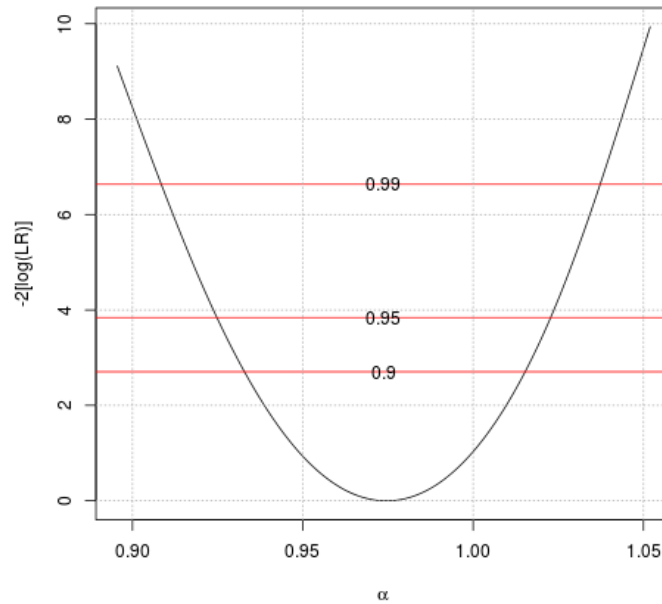


Figure 5: Profile likelihood for a parameter obtained with the `plot` method for ‘`profile.etasclass`’ objects, for the North California catalog.

Notice that in this example the larger size of the catalog with respect to the Italian catalog example leads to a considerably larger execution time.

Plotting results and profile likelihood for α :

```
R> plot(etas.flp.cal)
R> prof.flp.cal <- profile(etas.flp.cal, nprofile = 7, iprofile = 5)
R> plot(prof.flp.cal)
```

Asymptotic confidence intervals:

	Coverage	Lower	Upper
1	0.90	0.928	1.021
2	0.95	0.923	1.025
3	0.99	0.911	1.034

In Figure 4 the estimated background and triggered space intensities evaluated on the region are reported (obtained with the command `plot(etas.flp.cal)`). In Figure 5 we report the graphical output of the command `plot(prof.flp.cal)`, which shows the profile likelihood for the parameter α . The object `prof.flp.cal` has been obtained with the method `profile`.

7. Final remarks and future enhancement

The package **etasFLP** has been published on the Comprehensive R Archive Network (CRAN) and is available from <https://CRAN.R-project.org/package=etasFLP>. Some computing improvement can be made (together with some generalization). The iterative architecture of declustering will remain unchanged, but some improvements will be made: Particularly we

are studying the possibility of using very approximate estimates of parameters in the first steps of declustering, and then improving the precision in later steps. This should increase speed.

After some trial, we decided to use, in the optimization steps, the functions `nlm()` and `optim()` with the numerical computation of derivatives. Actually we experimented in `nlm()` the use of analytical derivatives instead of numerical derivatives, but we did not see significant advantages, both for the computational times and for convergence.

The main functionality improvement will be the use of the FLP approach with a variable kernel estimator, to allow for pattern of points with different geographical orientation.

Feedback from users will be however the best way to improve the package, in an open source philosophy.

Acknowledgments

This paper has been partially supported both by the grant of the University of Palermo (Italy): “2012-ATE-0332 - FFR 2012–2013 - Metodi statistici per dati spazio-temporali applicati all’analisi, monitoraggio e previsione ambientale. PI: Gianfranco Lovison” and by the national grant of the Italian Ministry of Education University and Research (MIUR) for the PRIN-2015 program (Progetti di ricerca di Rilevante Interesse Nazionale), “Prot. 20157PRZC4 - Research Project Title Complex space-time modeling and functional analysis for probabilistic forecast of seismic events. PI: Giada Adelfio”

We want also to thank Dr. Luigi Augugliaro for some useful advice.

References

- Adelfio G (2010a). “An Analysis of Earthquakes Clustering Based on a Second-Order Diagnostic Approach.” In F Palumbo, CN Lauro, MJ Greenacre (eds.), *Data Analysis and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 309–317. Springer-Verlag. doi:10.1007/978-3-642-03739-9_35.
- Adelfio G (2010b). “Kernel Estimation and Display of a Five-Dimensional Conditional Intensity Function.” *Nonlinear Processes in Geophysics*, **17**(2), 237–244. doi:10.5194/npg-17-237-2010.
- Adelfio G, Chiodi M (2009). “Second-Order Diagnostics for Space-Time Point Processes with Application to Seismic Events.” *Environmetrics*, **20**(8), 895–911. doi:10.1002/env.961.
- Adelfio G, Chiodi M (2013). “Mixed Estimation Technique in Semi-Parametric Space-Time Point Processes for Earthquake Description.” In V Muggeo, V Capursi, B Giovanni, G Lovison (eds.), *Proceedings of the 28th International Workshop on Statistical Modelling 8–13 July, 2013, Palermo*, pp. 65–70.
- Adelfio G, Chiodi M (2015a). “Alternated Estimation in Semi-Parametric Space-Time Branching-Type Point Processes with Application to Seismic Catalogs.” *Stochastic Environmental Research and Risk Assessment*, **29**(2), 443–450. doi:10.1007/s00477-014-0873-8.

- Adelfio G, Chiodi M (2015b). “FLP Estimation of Semi-Parametric Models for Space-Time Point Processes and Diagnostic Tools.” *Spatial Statistics*, **14**(B), 119–132. doi:10.1016/j.spasta.2015.06.004.
- Adelfio G, Chiodi M, De Luca L, Luzio D, Vitale M (2006). “Southern-Tyrrhenian Seismicity in Space-Time-Magnitude Domain.” *The Annals of Geophysics*, **49**(6), 1245–1257.
- Adelfio G, Chiodi M, Luzio D (2010). “An Algorithm for Earthquakes Clustering Based on Maximum Likelihood.” In F Palumbo, CN Lauro, MJ Greenacre (eds.), *Data Analysis and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 25–32. Springer-Verlag. doi:10.1007/978-3-642-03739-9_3.
- Adelfio G, Ogata Y (2010). “Hybrid Kernel Estimates of Space-Time Earthquake Occurrence Rates Using the Etas Model.” *The Annals of the Institute of Statistical Mathematics*, **62**(1), 127–143. doi:10.1007/s10463-009-0268-7.
- Adelfio G, Schoenberg FP (2009). “Point Process Diagnostics Based on Weighted Second-Order Statistics and Their Asymptotic Properties.” *The Annals of the Institute of Statistical Mathematics*, **61**(4), 929–948. doi:10.1007/s10463-008-0177-1.
- Becker RA, Wilks AR, Brownrigg R, Minka TP (2016). **maps**: Draw Geographical Maps. R package version 3.1.1, URL <https://CRAN.R-project.org/package=maps>.
- Chiodi M, Adelfio G (2011). “Forward Likelihood-Based Predictive Approach for Space-Time Point Processes.” *Environmetrics*, **22**(6), 749–757. doi:10.1002/env.1121.
- Chiodi M, Adelfio G (2017). **etasFLP**: Mixed FLP and ML Estimation of ETAS Space-Time Point Processes. R package version 1.4.0, URL <https://CRAN.R-project.org/package=etasFLP>.
- Console R, Jackson DD, Kagan YY (2010). “Using the ETAS Model for Catalog Declustering and Seismic Background Assessment.” *Pure and Applied Geophysics*, **167**(6–7), 819–830. doi:10.1007/s00024-010-0065-5.
- Daley DJ, Vere-Jones D (2003). *An Introduction to the Theory of Point Processes*. 2nd edition. Springer-Verlag. doi:10.1007/978-1-4757-2001-3.
- Jalilian A (2016). **ETAS**: Modeling Earthquake Data Using Epidemic Type Aftershock Sequence Model. R package version 0.3, based on Fortran code by Jiancang Zhuang, URL <https://CRAN.R-project.org/package=ETAS>.
- Marsan D, Lengliné O (2008). “Extending Earthquakes’ Reach Through Cascading.” *Science*, **319**(5866), 1076–1079. doi:10.1126/science.1148783.
- Nicolis O, Chiodi M, Adelfio G (2015). “Windowed ETAS Models with Application to the Chilean Seismic Catalogs.” *Spatial Statistics*, **14**, Part B, 151–165. doi:10.1016/j.spasta.2015.05.006.
- Nychka D, Furrer R, Paige J, Sain S (2016). **fields**: Tools for Spatial Data. R package version 8.10, URL <https://CRAN.R-project.org/package=fields>.

- Ogata Y (1988). “Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes.” *Journal of the American Statistical Association*, **83**(401), 9–27. doi:[10.1080/01621459.1988.10478560](https://doi.org/10.1080/01621459.1988.10478560).
- Ogata Y (1998). “Space-Time Point-Process Models for Earthquake Occurrences.” *The Annals of the Institute of Statistical Mathematics*, **50**(2), 379–402. doi:[10.1023/a:1003403601725](https://doi.org/10.1023/a:1003403601725).
- R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Schoenberg FP (2013). “Facilitated Estimation of ETAS.” *Bulletin of the Seismological Society of America*, **103**(1), 601–605. doi:[10.1785/0120120146](https://doi.org/10.1785/0120120146).
- Silverman BW (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.
- Terrell GR, Scott DW (1992). “Variable Kernel Density Estimation.” *The Annals of Statistics*, **20**(3), 1236–1265. doi:[10.1214/aos/1176348768](https://doi.org/10.1214/aos/1176348768).
- Utsu T (1961). “A Statistical Study on the Occurrence of Aftershocks.” *Geophysical Magazine*, **30**, 521–605.
- Veen A, Schoenberg FP (2008). “Estimation of Space-Time Branching Process Models in Seismology Using an EM-Type Algorithm.” *Journal of the American Statistical Association*, **103**(482), 614–624. doi:[10.1198/016214508000000148](https://doi.org/10.1198/016214508000000148).
- Vere-Jones D (1992). “Methods for the Description and Display of Earthquake Catalogues.” In A Walden, A Guttorp (eds.), *Data Analysis and Classification*, pp. 220–236. Edward Arnold, London.
- Wand MP, Jones C (1994). “Multivariate Plug-In Bandwidth Selection.” *Computational Statistics*, **9**(2), 97–116.
- Zhuang J, Ogata Y, Vere-Jones D (2002). “Stochastic Declustering of Space-Time Earthquake Occurrences.” *Journal of the American Statistical Association*, **97**(458), 369–379. doi:[10.1198/016214502760046925](https://doi.org/10.1198/016214502760046925).

Affiliation:

Marcello Chiodi, Giada Adelfio
Dipartimento di Scienze Economiche, Aziendali e Statistiche
Università di Palermo

viale delle Scienze Ed. 13

90144 Palermo, Italy

E-mail: marcello.chiodi@unipa.it, giada.adelfio@unipa.it

URL: <http://dssm.unipa.it/chiodi/>

Tel.: +39/91/23895236

Journal of Statistical Software

published by the Foundation for Open Access Statistics

January 2017, Volume 76, Issue 3

doi:[10.18637/jss.v076.i03](https://doi.org/10.18637/jss.v076.i03)

<http://www.jstatsoft.org/>

<http://www.foastat.org/>

Submitted: 2014-02-27

Accepted: 2015-10-02
