Adaptive estimation of spatio-temporal intensities using **kernstadapt**

```r
# Main package
library(kernstadapt)

# Complementary packages
library(spatstat)
library(sparr)
```

Introduction

kernstadapt is an R package for the spatio-temporal estimation of the intensities of spatio-temporal point processes. The package uses an adaptive approach where the kernel bandwidth varies with each data point to estimate the expected number of points in an observation window and a time interval. **kernstadapt** provides tools for separability testing, adaptive bandwidth computation, and fast calculation of the intensity using a bandwidth partitioning algorithm.

Intensity function

The intensity function of a spatio-temporal point process gives the expected number of points per unit area in a given spatial region W when the observations depend on time. That is, the intensity functions denotes the expected number of the process events at $\mathbf{u} \in W$ and $v \in T$:

$$\lambda(\mathbf{u}, v) = \lim_{|d\mathbf{u} \times dv| \to 0} \frac{\mathbb{E}\left[N(d\mathbf{u} \times dv)\right]}{d\mathbf{u}dv},$$

where $d\mathbf{u}$ and dv define small spatial ante temporal regions around the locations $\mathbf{u}$ and v; where $|\cdot|$ denotes area, and $N(\cdot)$ denotes the number of events of a given set.

Separable estimation of the intensity

A spatio-temporal point process is defined as *first-order separable*, when the intensity function can be factorised as a product of two spatial and temporal parts:

$$\lambda(\mathbf{u}, v) = \lambda_1(\mathbf{u})\lambda_2(v),$$

where $\lambda_1(\cdot)$ and $\lambda_2(\cdot)$ are non-negative functions.

Spatio-temporal adaptative estimator

The spatio-temporal adaptive kernel estimator is defined as follows.

$$\hat{\lambda}_{\epsilon,\delta}\left(\mathbf{u}, v\right) = \frac{1}{e_{\epsilon,\delta}\left(\mathbf{u}, v\right)} \sum_{i=1}^{n} K^{\mathrm{s}}_{\epsilon(\mathbf{u}_i)}\left(\mathbf{u} - \mathbf{u}_i\right) K^{\mathrm{t}}_{\delta(v_i)}\left(v - v_i\right), \qquad (\mathbf{u}, v) \in W \times T,$$

where the edge correction is given by

$$e_{\epsilon,\delta}(\mathbf{u},v) = \int_W \int_T K^{\mathrm{s}}_{\epsilon(\mathbf{u}')}(\mathbf{u}-\mathbf{u}')\, K^{\mathrm{t}}_{\delta(v')}(v-v')\, d\mathbf{u}' dv'.$$

In above equations, $K^{\mathrm{s}}_{\epsilon(\mathbf{u}_i)}(\cdot)$ and $K^{\mathrm{t}}_{\delta(v_i)}(\cdot)$ are spatial and temporal Gaussian kernels for space and time with bandwidths given by $\epsilon(\mathbf{u}_i)$ and $\delta(v_i) > 0$. In this case, since the bandwidths are functions of the data points, these functions must be adequately defined and estimated.

Fast estimation through partitioning algorithm

The partitioning algorithm performs the computation of the adaptive estimator by summing fixed-bandwidth estimates operating on appropriate subsets of the data points. The subsets of data points for the partitioning algorithm are defined by binning the variable bandwidths. The bins generate a partition of the point pattern X into $C_1 \times C_2$ subpatterns Y_{ij} and $X = \bigcup_{ij} Y_{ij}$. Then, we can compute the intensity as

$$\hat{\lambda}_{\epsilon,\delta}(\mathbf{u},v) \approx \sum_{i=1}^{C_1}\sum_{j=1}^{C_2} \hat{\lambda}^{*}_{\bar{\epsilon}_i,\bar{\delta}_j}(\mathbf{u},v|Y_{ij}),$$

where $\bar{\epsilon}_i$ and $\bar{\delta}_j$ are the midpoints of the ith spatial and jth temporal bins and $\hat{\lambda}^{*}_{\bar{\epsilon}_i,\bar{\delta}_j}(\mathbf{u},v|Y_{ij})$ is a fixed-bandwidth intensity estimate of Y_{ij}.

Data

kernstadapt has three datasets included to serve as working examples of the package capabilities.

Aegiss

This dataset is a spatio-temporal point pattern where the points stand for non-specific gastrointestinal infections in Hampshire, UK. The time covers from 2001 to 2003.

Santander's earthquakes

This dataset is a spatio-temporal point pattern where the points are earthquakes in Santander, Colombia, from 2000 to 2020.

Amazon fires

This dataset is a spatio-temporal point pattern where the points are locations of active deforestation fires (starting within past 24 hours) from 01/01/2021 to 10/10/2021 (284 days).

```
data(aegiss, santander, amazon)
par(mfrow = c(1,3))

plot(aegiss, main = "Aegiss", bg = rainbow(250))
plot(santander, main = "Santander", bg = rainbow(250))
plot(amazon[sample.int(amazon$n, 5000)], main = "Amazon fires", bg = rainbow(250))
```

Variable bandwidth

In the adaptive kernel estimation, each data point is equipped with a bandwidth value following the spatial and temporal functions given by

$$\epsilon(\mathbf{u}) = \frac{\epsilon^\star}{\gamma^{\mathrm{s}}}\sqrt{\frac{n}{\lambda^{\mathrm{s}}(\mathbf{u})}}, \quad \text{and} \quad \delta(v) = \frac{\delta^\star}{\gamma^{\mathrm{t}}}\sqrt{\frac{n}{\lambda^{\mathrm{t}}(v)}},$$

where $\epsilon^\star$ and $\delta^\star$ are *global bandwidths*, $\lambda^{\mathrm{s}}(\mathbf{u})$ and $\lambda^{\mathrm{t}}(v)$ are marginal intensity functions in space and time, γ^{s} and γ^{t} are the geometric means of the marginal intensities.

Here, we assign some global bandwidths using several methods, but the user may let **kernstadapt** choose its defaults.

```
# Cronie and van Lieshout's spatial bandwidth
bw.xy.aegiss <- bw.abram(aegiss, h0 = bw.CvL(santander))

# Modified Silverman's rule of thumb temporal bandwidth
bw.t.aegiss <- bw.abram.temp(aegiss$marks, h0 = bw.nrd(aegiss$marks))

# Scott’s isotropic rule of thumb for spatial bandwidth
bw.xy.santander <- bw.abram(santander, h0 = bw.scott.iso(santander))

# Unbiased cross-validation for temporal bandwidth
bw.t.santander <- bw.abram.temp(santander$marks,
                                h0 = bw.ucv(as.numeric(santander$marks)))
```

Separability test

In order to test separability in a spatio-temporal point process, **kernstadapt** uses a simple statistical test based on spatio-temporal quadrat counts.

```
sapply(list(aegiss, santander, amazon), separability.test)
```

Therefore, we conclude that only Santander dataset can be assumed as separable.

Separable estimation of the intensity

Estimator

In the separable case, the estimator is given by

$$\hat{\lambda}_{\epsilon,\delta}\left(\mathbf{u},v\right) = \frac{1}{n}\left(\frac{1}{e_\epsilon\left(\mathbf{u}\right)}\sum_{i=1}^{n} K^{\mathrm{s}}_{\epsilon(\mathbf{u}_i)}\left(\mathbf{u}-\mathbf{u}_i\right)\right)\left(\frac{1}{e_\delta\left(v\right)}\sum_{i=1}^{n} K^{\mathrm{t}}_{\delta(v_i)}\left(v-v_i\right)\right), \qquad (\mathbf{u},v)\in W\times T,$$

where $K^{\mathrm{s}}_\epsilon(\cdot)$ and $K^{\mathrm{t}}_\delta(\cdot)$ are bivariate and univariate kernels for space and time. $e_\epsilon\left(\mathbf{u}\right)$ and $e_\delta\left(v\right)$ are edge-correction factors.

kernstadapt can estimate the intensity directly by applying the above formula or approximating it using a fast partition method. We apply both methods to Santander’s data, the separable one.

Direct estimator

```
# Direct estimation, separable case
lambda <- dens.direct.sep(X = santander,
                          dimyx = 128, dimt = 64,
```

```
                                bw.xy = bw.xy.santander,
                                bw.t = bw.t.santander)
```

Now, we plot some snapshots of the spatio-temporal estimated intensity.

```
# We select some fixed times for visualisation
I <- c(12, 18, 23, 64)

# We subset the lists
SDS <- lapply(lambda[I], function(x) (abs(x)) ^ (1/6))

# Transform to spatial-objects-lists
SDS <- as.solist(SDS)

# We generate the plots
plot(SDS, ncols = 4, equal.ribbon = T, box = F,
     main = 'Direct estimation, separable case')
```

Partition algorithm estimator

```
# Partition algorithm estimation, separable case
lambda <- dens.par.sep(X = santander,
                       dimyx = 128, dimt = 64,
                       bw.xy = bw.xy.santander,
                       bw.t = bw.t.santander,
                       ngroups.xy = 20, ngroups.t = 10)
```

Now, we plot some snapshots of the spatio-temporal estimated intensity.

```
# We select some fixed times for visualisation
I <- c(12, 18, 23, 64)

# We subset the lists
SPS <- lapply(lambda[I], function(x) (abs(x)) ^ (1/6))

# Transform to spatial-objects-lists
SPS <- as.solist(SPS)

# We generate the plots
plot(SPS, ncols = 4, equal.ribbon = T, box = F,
     main = 'Partition algorithm estimation, separable case')
```

Non-separable estimation of the intensity

Estimator

In the non-separable case, an adaptive estimator for the intensity is

$$\hat{\lambda}_{\epsilon,\delta}\left(\mathbf{u},v\right)=\frac{1}{e_{\epsilon,\delta}\left(\mathbf{u},v\right)}\sum_{i=1}^{n}K^{\mathrm{s}}_{\epsilon(\mathbf{u}_i)}\left(\mathbf{u}-\mathbf{u}_i\right)K^{\mathrm{t}}_{\delta(v_i)}\left(v-v_i\right),\qquad \left(\mathbf{u},v\right)\in W\times T,$$

where the edge correction term is,

$$e_{\epsilon,\delta}(\mathbf{u}, v) = \int_W \int_T K^{\mathrm{s}}_{\epsilon(\mathbf{u}')}(\mathbf{u} - \mathbf{u}')\, K^{\mathrm{t}}_{\delta(v')}(v - v')\, d\mathbf{u}' dv'.$$

The **kernstadapt** package has the functionality of applying the direct estimator given above. It also has a fast partition method for estimating the intensity in a non-separable way.

Direct estimator

```
#Direct estimation, non-separable case
lambda <- dens.direct(aegiss,
                      dimyx = 32, dimt = 16,
                      bw.xy = bw.xy.aegiss,
                      bw.t = bw.t.aegiss,
                      at = "bins")
```

We have used very coarse spatial and temporal grids; as the direct estimator needs lots of computational resources.

Now, we plot some snapshots of the spatio-temporal estimated intensity.

```
# We select some fixed times for visualisation
I <- c(2, 5, 8, 16)

# We subset the lists
NSDA <- lapply(lambda[I], function(x) (abs(x)) ^ (1/6))

# Transform to spatial-objects-lists
NSDA <- as.solist(NSDA)

# We generate the plots
plot(NSDA, ncols = 4, equal.ribbon = T, box = F,
     main = 'Direct estimation, non-separable case')
```

Partition algorithm estimator

We apply the partition algorithm method and let the package decide about global bandwidths.

```
# Partition algorithm estimation, non-separable case
lambda <- dens.par.sep(X = amazon,
                       dimyx = 128, dimt = 64,
                       ngroups.xy = 20, ngroups.t = 10)
```

Now, the visualisation of the spatio-temporal estimated intensity.

```
# We select some fixed times for visualisation
I <- c(12, 18, 23, 64)

# We subset the lists
NSPA <- lapply(lambda[I], function(x) (abs(x)) ^ (1/6))

# Transform to spatial-objects-lists
```

```r
NSPA <- as.solist(NSPA)

# We generate the plots
plot(NSPA, ncols = 4, equal.ribbon = T, box = F,
     main = 'Partition algorithm estimation, non-separable case')
```