Temporal Random Sets: Simulation and Estimation. A Matlab $${\rm Toolbox}^1$$

María Elena Díaz, Guillermo Ayala

September 8, 2008

¹©Authors. All rights reserved.

Version 1.3

Date 15 May 2008.

Authors María Elena Díaz <elena.diaz@uv.es> and Guillermo Ayala <guillermo.ayala@uv.es>. Substantial amounts of code were contributed by Rafael Sebastian and Ester Díaz.

Maintainer María Elena Díaz

Depends Matlab 2007b, Image processing Toolbox and Garch Toolbox. R (R-2.6.1 or newer) and fda, splines and spatstat (1.12-6 or newer, http://cran.r-project.org/) packages.

Description A toolbox for simulation and estimation of temporal random sets.

License GNU General Public License version 3 or newer http://www.gnu.org/licenses/gpl-3. 0.txt

URL http://www.uv.es/tracs

This manual is published under the license GFDL. A copy of this license is available at http://www.gnu.org/licenses/fdl.txt

Acknowledgement

We thank the Human Frontier Science Organization Program, RGY40/2003, and the Spanish Ministry of Science and Education for their financial support and TIN2007-67587 (FEDER FUNDS).

Contents

1	Ten	nporal Random Sets 1							
	1.1	Temporal Random Closed Set 1							
	1.2	Boolean model							
	1.3	Temporal Boolean Model							
	1.4	Simulation of Temporal Boolean Models							
		1.4.1 Function tbmsimulation.m $\ldots \ldots 4$							
		1.4.2 Function tbmgerms.m							
		1.4.3 Function tbmgrains.m							
		1.4.4 Function tbm durations.m $\ldots \ldots 7$							
	1.5	Estimation of Temporal Boolean Models							
		1.5.1 Function tbmcontminT.m $\ldots \ldots 7$							
		1.5.2 Function tbmcontminH.m							
		1.5.3 Function tbmmoments.m							
		1.5.4 Function tbmblockcorr.m							
		1.5.5 Function tracs2dtangentpoints.m							
		1.5.6 Function tracs3dtangentpoints.m							
		1.5.7 Function tbmspatialestimation.m							
		1.5.8 Function tbmalphaestimation.m							
		1.5.9 Function tbmtemporalestimation.m							
		1.5.10 Function tracsvolumefraction.m $\ldots \ldots 14$							
		1.5.11 Function tracsisolated clumps.m							
	1.6	Visualization of a TRACS							
		1.6.1 Function tracs3dplot.m $\dots \dots \dots$							
		1.6.2 Function tracs2dplot.m							
	1.7	A demo							
2	Ten	Temporal Boolean Functions							
	2.1	- Simulation of Temporal Boolean Functions							
		2.1.1 Function tbfsimulation.m							

		2.1.2	Function tbfgrains.m	21
		2.1.3	Function tbfbivariatesimulation.m	21
3	Biv	Bivariate Temporal Random Sets		
	3.1	Bivari	ate Temporal Random Sets	23
		3.1.1	Definition	23
		3.1.2	Summary statistics for bivariate temporal random sets	25
		3.1.3	Estimators	25
	3.2	Testin	g spatial and spatial-temporal dependencies	26
		3.2.1	Testing null hypotheses by using toroidal shift randomizations	27
		3.2.2	Estimating the spatio-temporal interval of dependencies	28
3.3 Functions for the analysis of dependencies between types		Funct	ions for the analysis of dependencies between types in bivariate TRACS	28
		3.3.1	Function tbmbivariatesimulation	28
		3.3.2	Function tracsrandomization.m	30
		3.3.3	Function tracscrossC.m	30
		3.3.4	Function tracscrossCh0.m	31
		3.3.5	Function tracscrossCht.m	31
		3.3.6	Function tracscrossK1.m	32
		3.3.7	Function tracscrossK.m	32
		3.3.8	Function tracspvalueMC.m	33
		3.3.9	Function tracspvaluemapMC.m	33
		3.3.10	Demos	34
	3.4	Visual	lization of a bivariate-TRACS	35
		3.4.1	Function tracsbi3dplot.m	35
3.5 Functions for the analysis of dependencies between types in		Funct	ions for the analysis of dependencies between types in bivariate fuzzy TRACS	35
		3.5.1	Function tracsfuzzygetalpha.m	35

4 Installation

 $\mathbf{37}$

List of Figures

1.1	A three dimensional reconstruction of a TBM with cylindrical grains			
1.2 Tangent points in a two-dimensional Boolean model in red. (a) top-do-				
	(b) bottom-up; (c) left-right and (d) right-left	11		
1.3	A spatial-temporal reconstruction of a TBM using function tracs 3 dplot .	16		
1.4	Several consecutive frames of a simulated TBM with cylindrical grains \ldots	18		
1.5	(a) Estimation of $\alpha(t)$, $\beta(t)$ and γ . (b) Estimated pdf of the durations	18		
2.1	A realization of a simulated bivariate temporal Boolean function . (a)-(c)			
	Three consecutive temporal cross-sections. The red color corresponds to events of			
	type 1, the green color to events of type 2. A linked-pairs model for the germs was			
	used in order to generate spatial-temporal dependencies between germs types. \ldots	20		
3.1	A realization of a simulated bivariate temporal random set. (a)-(c) Three			
	consecutive temporal cross-sections. The white color corresponds to events of type			
	1, the light grey to events of type 2, and the dark grey to overlapping regions. \dots	24		
3.2	A spatial-temporal reconstruction: the red color corresponds to type-1 events			
	and the green color to type-2 events	24		
3.3		35		

Chapter 1

Temporal Random Sets

In this chapter, we introduce the Temporal Random Closed Sets (TRACS).

In particular, we define the Temporal Boolean Model (TBM) in $\mathbb{R}^2 \times \mathbb{R}_+$ (see [1,7]) for further details on the definition and statistical analysis).

1.1 Temporal Random Closed Set

Intuitively, a random closed set Φ is a random mechanism producing random (closed) subsets in space (see [2] for further details).

We model each binary image of the sequence as a realization of a random set in \mathbb{R}^2 and the whole image sequence as a random set in $\mathbb{R}^2 \times \mathbb{R}_+$. Let us denote by $\Phi(t)$ the random set corresponding to the random binary image at the observation time t, with $t \ge 0$. If we stack the observed temporal cross-sections for every time t, we define the *temporal random set* in $\mathbb{R}^2 \times \mathbb{R}_+$ as

$$\Phi = \bigcup_{t \ge 0} \Phi(t). \tag{1.1}$$

This model can be considered as a particular case of a non-isotropic three dimensional random set.

1.2 Boolean model

Let us give some notation: ν_d will denote the Lebesgue measure in \mathbb{R}^d (the area in \mathbb{R}^2 and the volume in \mathbb{R}^3); $A \oplus B = \{a + b : a \in A, b \in B\}$ will be the Minkowski addition of the sets A and B and $\check{A} = \{-x : x \in A\}$ is the symmetric of A with respect to the origin.

Let $\Psi = \{y_1, y_2, \ldots\}$ be a stationary Poisson point process in \mathbb{R}^d with intensity λ [3]. Let Ξ_1, Ξ_2, \ldots be a sequence of independent and identically distributed (as Ξ_0) random compact sets in \mathbb{R}^d such that they are independent of the Poisson point process and $E\nu_d(\Xi_0 \oplus \check{K}) < +\infty$ for every compact set K. A Boolean model is defined as

$$\Phi = \bigcup_{n=1}^{\infty} (\Xi_n + y_n). \tag{1.2}$$

The points y_n are called the germs and the sets Ξ_n are the grains. The random set Ξ_0 is called the primary grain. The distribution of a random closed set is completely characterized by its capacity functional defined as $T(K) = P(\Phi \cap K \neq \emptyset)$ (with K an arbitrary compact subset). For a Boolean model, this functional is given by $T(K) = 1 - \exp\{-\lambda E\nu_d(\Xi_0 \oplus \check{K})\}$. In particular, if $K = \{0\}$, the value $T(\{0\}) = P(0 \in \Phi)$ is known as the area fraction, the mean area in \mathbb{R}^2 (mean volume in \mathbb{R}^3) covered by the random set per unit area (respectively, per unit volume). The area fraction in a d-dimensional Boolean model is given by

$$p = 1 - \exp\{-\gamma E\nu_d(\Xi_0)\}.$$
 (1.3)

Some problems of the statistical analysis of Boolean models are the estimation of: the intensity of the germ process, the mean area of the primary grain, a_0 , and the mean perimeter, u_0 .

1.3 Temporal Boolean Model

Let $\Psi = \{(x_i, t_i)\}_{i \ge 1}$ be a stationary Poisson point process in $\mathbb{R}^2 \times \mathbb{R}_+$ with intensity λ (the locations and birth times). Let $\{A_i\}_{i \ge 1}$ be a sequence of independent and identically distributed (as A_0) random compact sets in \mathbb{R}^2 . Let $\{d_i\}_{i \ge 1}$ be a sequence of independent and identically distributed (as D) positive random variables (the durations). We assume that Ψ , $\{A_i\}_{i \ge 1}$ and $\{d_i\}_{i \ge 1}$ are independent and that $E\nu_3(A_0 \times [0, D] \oplus \check{K}) < +\infty$ for any compact subset K of \mathbb{R}^3 . The *temporal Boolean model* is the random set defined as

$$\Phi = \bigcup_{i \ge 1} (A_i + x_i) \times [t_i, t_i + d_i].$$

$$(1.4)$$

The set $(A_i + x_i) \times [t_i, t_i + d_i]$ is a rectangle in $\mathbb{R}^2 \times \mathbb{R}_+$, the *i*-th event. Fig. 1.1 (a) displays a realization of a temporal Boolean model where A_0 is a random disc. The temporal cross-sections are (static) 2D Boolean models.

Our data consist of a discrete set of temporal cross-sections corresponding to the observation times $s_1 \leq \ldots \leq s_k$. Let us denote the temporal cross-section at time s as $\Phi_s = \Phi \cap (\mathbb{R}^2 \times \{s\})$. Each Φ_{s_i} is contained in the product space $W \times [0, T]$, where W is the observation window and [0, T]is the total time interval observed. The projection of Φ_s over \mathbb{R}^2 (that with an abuse of notation will be denoted Φ_s) is given by $\Phi_s = \bigcup_{\{i:t_i \leq s \leq t_i + d_i\}} (A_i + x_i)$. Φ_s is the binary image observed at time s.

Proposition 1 If Φ is a temporal Boolean model with intensity λ and primary grain $A_0 \times [0, D]$, then the temporal cross-section Φ_s is a 2D Boolean model with primary grain A_0 and intensity

$$\gamma = \lambda ED. \tag{1.5}$$



Figure 1.1: A three dimensional reconstruction of a TBM with cylindrical grains

Let us consider the estimation of the distribution of the random duration for the events, D. For $s_1 \leq s_2$, we will consider the following three random sets:

$$\Phi_{s_1, s_2} = \bigcup_{i: t_i \le s_1 \le s_2 \le t_i + d_i} A_i + x_i, \tag{1.6}$$

$$\Phi_{s_1, s_2^-} = \bigcup_{i: t_i \le s_1 \le t_i + d_i < s_2} A_i + x_i, \tag{1.7}$$

$$\Phi_{s_1^-, s_2} = \bigcup_{i: s_1 < t_i \le s_2 \le t_i + d_i} A_i + x_i.$$
(1.8)

 Φ_{s_1,s_2} corresponds to the union of the grains which are in Φ_{s_1} and Φ_{s_2} , Φ_{s_1,s_2^-} to the grains in Φ_{s_1} but not in Φ_{s_2} , and $\Phi_{s_1^-,s_2}$ to the grains in Φ_{s_2} but not in Φ_{s_1} .

Theorem 1 Let Φ be a temporal Boolean model. The random sets Φ_{s_1,s_2} , Φ_{s_1,s_2^-} and $\Phi_{s_1^-,s_2}$ are then independent 2D Boolean models. Let us denote $\beta(s_2 - s_1)$ the intensity of Φ_{s_1,s_2} (the mean number of germs that are alive in the time interval $[s_1, s_2]$), which is given by

$$\beta(s_2 - s_1) = \lambda p(s_2 - s_1), \tag{1.9}$$

where $p(s) = \int_{s}^{+\infty} P(D \ge v) dv$. Moreover, under stationarity, Φ_{s_1,s_2^-} and $\Phi_{s_1^-,s_2}$ have the same intensity, $\alpha(s_2 - s_1)$ (the mean number of germs that died in a time interval $[s_1, s_2]$) given by

$$\alpha(s_2 - s_1) = \lambda ED - \lambda p(s_2 - s_1) = \gamma - \beta(s_2 - s_1).$$
(1.10)

Proposition 2

$$P(0 \in \Phi_{s_1} | 0 \notin \Phi_{s_2}) = P(0 \in \Phi_{s_2} | 0 \notin \Phi_{s_1}) = 1 - \exp\{-\alpha(s_2 - s_1)a_0\}.$$
(1.11)

Let ϕ_{s_i} be the binary image observed at time s_i , then $P(0 \in \Phi_{s_1} | 0 \notin \Phi_{s_2})$ will be estimated as

$$\hat{P}(0 \in \Phi_{s_1} | 0 \notin \Phi_{s_2}) = \frac{\hat{P}(0 \in \Phi_{s_1} \setminus \Phi_{s_2})}{\hat{P}(0 \notin \Phi_{s_2})} = \frac{\nu_2(\phi_{s_1} \cap \phi_{s_2}^c \cap W)}{\nu_2(\phi_{s_2}^c \cap W)},$$
(1.12)

and by using (1.11) and (1.12), we will estimate $\delta(s_2 - s_1) = \alpha(s_2 - s_1)a_0$ by solving the equation

$$1 - \exp\{-\hat{\delta}(s_2 - s_1)\} = \frac{1}{2} \left(\frac{\nu_2(\phi_{s_1} \cap \phi_{s_2}^c \cap W)}{\nu_2(\phi_{s_2}^c \cap W)} + \frac{\nu_2(\phi_{s_2} \cap \phi_{s_1}^c \cap W)}{\nu_2(\phi_{s_1}^c \cap W)} \right).$$
(1.13)

The mean area (a_0) and the mean perimeter (u_0) of A_0 and γ were estimated from each image in the sequence by using the minimum contrast method [5]. The different estimates are correlated values of the same parameters since the model is stationary, therefore the estimates over the image sequence were then combined by using the batch-mean method [4]. Let \hat{a}_0 and $\hat{\gamma}$ be the estimates, the functions $\alpha(s)$ and $\beta(s)$ can be estimated as

$$\hat{\alpha}(s) = \frac{\hat{\delta}(s)}{\hat{a}_0},\tag{1.14}$$

and

$$\ddot{\beta}(s) = \hat{\gamma} - \hat{\alpha}(s). \tag{1.15}$$

From (1.10), it follows that $\alpha'(s) = \lambda(1 - F_D(s))$ and $\alpha''(s) = -\lambda f_D(s)$ where α' and α'' are the first and second derivatives of α , whereas F_D and f_D are the cumulative distribution function and the density function of the random variable D, respectively. The function $\alpha(s)$ is estimated at the observation points $\{s_1, \ldots, s_n\}$. The spatial-temporal intensity λ is obtained by taking into account that $\alpha'(0) = -\lambda p'(0) = \lambda P(D \ge 0) = \lambda$, yielding

$$\hat{\lambda} = \hat{\alpha}'(0). \tag{1.16}$$

The probability density of D, f_D , is calculated as

$$\hat{f}_D(s) = -\frac{1}{\hat{\lambda}} \hat{\alpha}''(s). \tag{1.17}$$

1.4 Simulation of Temporal Boolean Models

To obtain further details of each function of the Toolbox, you can type help functionname.m in the Matlab command window. See chapter 4 for installation details.

1.4.1 Function tbmsimulation.m

MOV = TBMSIMULATION(S,WO,T,LAMBDA,SHAPE,TYPESIZE,PARAMSIZE,TYPEDURATION,PARAMDURATION)

Description

This function generates a realization of temporal Boolean model.

Arguments

S vector of sampling times (seconds).

W0 binary image corresponding to the observation window.

T length of the time interval (seconds).

LAMBDA mean number of points per unit area and unit time.

SHAPE shape of the primary grain. Possible values are: 'disk', 'square'.

TYPESIZE distribution of the grain sizes. Possible values are: 'uniform' (**PARAMSIZE** is a vector with the minimum and maximum values), 'gaussian' (**PARAMSIZE** is a vector with the mean and standard deviation).

TYPEDURATION distribution of the grains durations (seconds). Possible values are: 'exponential' (**PARAMDURATION** is the mean), 'uniform' (**PARAMDURATION** is a vector with inferior and superior limits of the interval), 'gaussian' (**PARAMDURATION** is a vector with the mean and standard deviation), 'gamma' (**PARAMDURATION** is a vector with the shape and scale parameter).

Value

MOV. This function returns an AVI file with a realization of the temporal Boolean model at the specified observation window W and time interval T.

Details

The observation window is assumed to have no holes. The avi movie is a binary image with two gray-levels: 0 corresponds to the background and 1 is the model. In a TBM the locations and birth time are independent. We assume spatial-temporal stationarity. If the grains sizes follow a Gaussian distribution, we truncate the Gaussian, in this way, we only consider positive radii.

The strel('disk', R) function of the Image Processing Toolbox generates a disk with diameter 2 * R + 1, it contributes to the error in the estimation of the area of the primary grain. Moreover, this function does not generate perfect circular grains, specially in case of small radii. It may influence on the error in the estimation of the area and perimeter of the primary grain. See [1] for further information about the errors in the estimation.

1.4.2 Function tbmgerms.m

[XN, YN, TN] = TBMGERMS(W, T, LAMBDA, TYPEGRAIN, PARAMGRAIN, TYPEDUR, PARAMDUR)

Description

This function generates the germs, a spatial-temporal Poisson point process.

Arguments

W binary image that defines the observation window.

T length of the time interval (seconds).

LAMBDA mean number of points per unit area and unit time.

TYPEGRAIN distribution of grain sizes. Possible values are: 'uniform' (**PARAMGRAIN** is a vector with the inferior and superior limits of the interval), 'gaussian' (**PARAMGRAIN** is a vector with the mean and standard deviation). Values are given in pixels.

TYPEDUR is the distribution of the grain durations. Possible values are: 'exponential' (**PARAM-DUR** is the mean), 'uniform' (**PARAMDUR** is a vector with the inferior and superior limit of the interval), 'gaussian' (**PARAMDUR** is a vector with the mean and standard deviation), 'gamma' (**PARAMDUR** is a vector with the shape and scale parameter). Values are given in seconds.

Value

[XN, YN, TN]. This function returns three vectors of size $1 \times N$ with the N points i.i.d. uniformly distributed in $|W| \times T$. (XN, YN) are the locations of the germs and TN are the birth times of the germs. Locations and birth times are independent.

Details

The observation window is assumed to have no holes. If **TYPEGRAIN** is 'gaussian', a truncated Gaussian is generated. Only positive values are considered.

1.4.3 Function tbmgrains.m

[SIZESGR_N, ANGLESGR_N] = TBMGRAINS(N, TYPEPDF, PARAM)

Description

This function generates the sizes and orientations of the grains.

Arguments

N number of grains.

TYPEPDF distribution of the grains sizes. Possible values: 'uniform' (**PARAM** is a vector with the inferior and superior limit of the interval, 'gaussian' (**PARAM** is a vector with the mean and standard deviation).

Value

 $[SIZESGR_N, ANGLESGR_N]$. This function returns two vectors $SIZESGR_N, ANGLESGR_N$ with size $1 \times N$ which correspond to the grains sizes and the grains orientations, respectively. $SIZESGR_N$ are distributed as **TYPEPDF** with parameters **PARAM**. Discrete values are generated. $ANGLESGR_N$ are distributed in 0, 45, 90, 135, 180, 225, 270, 315.

Details

If **TYPEPDF** is 'gaussian', a truncated Gaussian is generated. Only positive values are considered.

1.4.4 Function tbmdurations.m

DN = TBMDURATIONS(N, TYPEPDF, PARAM)

Description

This function generates the grains durations.

Arguments

N number of grains.

TYPEPDF distribution of the grains durations. Possible values: 'exponential (**PARAM** is the mean), 'uniform' (**PARAM** is a vector with the inferior limit and superior limit of the interval), 'gaussian' (**PARAM** is a vector with the mean and standard deviation) and 'gamma' (**PARAM** is a vector with the shape parameter and scale parameter).

Value

 D_N , this function returns a vector of size $1 \times N$. The durations are distributed as **TYPEPDF** with parameters **PARAM**.

Details

If TYPEPDF is 'gaussian', a truncated Gaussian is generated. Only positive values are considered.

1.5 Estimation of Temporal Boolean Models

1.5.1 Function tbmcontminT.m

[GAMMA, PERIM, AREA, FV] = TBMCONTMINT(BW, W, KTYPE)

Description

Estimation of the parameters of a two-dimensional Boolean model by applying the Minimun Con-

trast Method using the capacity functional T.

Arguments

BW a realization of a two-dimensional Boolean model (logical).W the observation window (logical).Ktype the type of convex set: 'disk' or 'square'.

Value

[gamma, perimeter, area, volfrac]. This function returns the parameters of the two-dimensional boolean model. It provides a four-dimensional vector, being gamma: number of germs per unit area; perimeter: mean perimeter of the primary grain; area: mean area of the primary grain and volfrac: the volume fraction.

Details

W is assumed to have no holes. The capacity functional $T_K(t)$ takes values in the interval [0, 1] since it represents the cumulative distribution function and, as a consequence, the empirical values of $log(1 - T_K(t))$ tend to infinite. Hence, only values of $1 - T_K(t)$ in the interval [0, 0.8] are used in the fitting of the second order polynomial derived from the Steiner formula. The square structuring element is recommended to be used because its digital approximation is more accurate than the corresponding approximation of the disk.

1.5.2 Function tbmcontminH.m

[GAMMA, PERIMETER, FV] = TBMCONTMINH(BW, W, KTYPE)

Description

Estimation of the parameters of a two-dimensional Boolean model by applying the Minimun Contrast Method using the contact distribution function H.

Arguments

BW a realization of a two-dimensional Boolean model (logical).W the observation window (logical).Ktype the type of convex set: 'disk' or 'square'.

Value

[gamma, perimeter, volfrac]. This function returns the parameters of the two-dimensional Boolean model: a three-dimensional vector, being gamma: number of germs per unit area; perimeter: mean perimeter of the primary grain and volfrac the volume fraction.

Details

W is assumed to have no holes. Only values of $1 - H_K(t)$ in the interval [0,0.7] are used in the fitting of the second order polynomial derived from the Steiner formula. Function tbmcontminT.m led to better results than tbmcontminH.m in our simulation study.

1.5.3 Function tbmmoments.m

[GAMMA, PERIM, AREA, FV] = TBMMOMENTS(BW, W, TYPE, DIR)

Description

Estimation of the parameters of a two dimensional Boolean model using the method of the moments.

Arguments

BW a realization of a two-dimensional Boolean model.

 ${\bf W}$ the observation window.

TYPE a string with the set of moments to be used: 'pNL' | 'pNX'.

DIR direction to be use to calculate the specific convexity number: 'top-down' | 'bottom-up' | 'left-right' | 'right-left'.

Value

[gamma, perimeter, area, volfrac]. This function returns the parameters of the two-dimensional boolean model. It provides a four-dimensional vector, being gamma: number of germs per unit area; perimeter: mean perimeter of the primary grain; area: mean area of the primary grain and volfrac: the volume fraction.

Details

The method of moments is based on the area fraction p, the specific boundary length L, the Euler-Poincare characteristic χ and the specific convexity number N^+ . The convexity number is calculated using the configuration The convexity number is defined as the number of first entries in the model

Table 1.1: Calculation of the convexity number N^+

in a given direction

1.5.4 Function tbmblockcorr.m

[DATAM, INTERVAL] = TBMBLOCKCORR(TAMINI, DATA, ALPHA)

Description

Estimation of the mean and confidence interval using the batch-mean method.

Arguments

TamIni initial size of the blocks. A value of 2 is used by default.

Data vector with the data.

alpha p-value for the inverse of Student's T cumulative distribution function. A value of alpha = 1 - 0.05/2 = 0.975 is used.

Value

A vector with two elements [DataM, Interval]. DataM is the batch-mean and Interval is the confidence interval.

Dependencies

This function depends on the autocorr function of the Garch Matlab Toolbox.

1.5.5 Function tracs2dtangentpoints.m

NPOINTS = TRACS2DTANGENTPOINTS(BM,W,DIR)

[NPOINTS,X,Y] = TRACS2DTANGENTPOINTS(BM,W,DIR)

Description

Tangent points in a two dimensional random set in a given direction.

Arguments

BM a realization of a two-dimensional random set.W the observation window.DIR direction: 'top-down' |'bottom-up' |' left-right' | 'right-left'.

Details

The tangent points are defined as the first entries in the model in a given direction and are calculated

using rotations of the configuration given in Table 1.2.



Figure 1.2: **Tangent points in a two-dimensional Boolean model in red**. (a) top-down; (b) bottom-up; (c) left-right and (d) right-left

1.5.6 Function tracs3dtangentpoints.m

NPOINTS = TRACS3DTANGENTPOINTS(MOV,W,DIR)

[NPOINTS,X,Y,FRAME,MOVRES] = TRACS3DTANGENTPOINTS(MOV,W,DIR)

Description

Tangent points of a Temporal Random Set

Arguments

MOV a realization of a Temporal Random Set. **W** the observation window. DIR direction: 'top-down' |'bottom-up' |' left-right' | 'right-left'.

X,Y,FRAME are row vectors. MOVRES is a movie with the tangent points overlapped on the model in red color.

Details

The tangent points are defined as the first entries in the model in a given direction and are calculated using the transpose and/or flips in the direction left-right of the three-dimensional configuration

Table 1.2: Calculation of the tangent points in 3d

conf(:,:,1) =	$[0 0 0 \dots 0 0 0;$
	000000;]
conf(:,:,2) =	[000000;
	$0\ 1\ 1\\ 1\ 1\ 0;$]

Note this configuration cannot be applied to the first frame.

1.5.7 Function tbmspatialestimation.m

RES = TBMSPATIALESTIMATION(MOV, W, METHOD, TYPE)

RES = TBMSPATIALESTIMATION(MOV, W, METHOD, TYPE, DIR)

Description

Estimation of the spatial parameters of a Temporal Boolean model.

Arguments

MOV a realization of a Temporal Boolean Model.

W the observation window (logical).

METHOD minimum contrast method using the capacity functional T or the contact distribution function H. Values are: 'ContMinH', 'ContMinT'.

TYPE if METHOD is 'ContMinH'|'ContMinT', TYPE is the type of the convex set used in the minimum contrast method 'disk'|'square'. If METHOD is 'Moments', TYPE is the set of moments to be used 'pNL'|'pNX'.

DIR direction used to calculate the specific convexity number: 'top-down'|'bottom-up'|'left-right'|'right-left'.

Value

A structure with the estimation of the mean value and the confidence interval of the number of germs per unit area, of the mean perimeter, of the mean area of the primary grain and of the volume fraction. The fields of the structure are denoted as Gamma, Perimeter, Area, Vf. Each field is a two-dimensional vector with the mean and the confidence interval.

Details

W is assumed to have no holes.

1.5.8 Function tbmalphaestimation.m

ALPHAEST = TBMALPHAESTIMATION(MOV, W, AO, TFIN, FPS)

Description

Estimation of the α -function of a Temporal Boolean Model (the mean number of germs that died/are given born in a time interval).

Arguments

mov a movie with a realization of a Temporal Boolean Model. W the observation window. **a0** area of the primary grain. tfin estimation range for the α -function is [0:1/fps:tfin]. fps frames per seconds.

Value

It returns a matrix with the number of rows defined by length of [0:1/fps:tfin] and two columns. The first column is the value of the α -function and the second column is the confidence interval provided by the batch-mean method.

Details

tfin must be high enough to stabilize the value of the $\alpha\text{-function.}$

1.5.9 Function tbmtemporalestimation.m

RES = TBMTEMPORALESTIMATION(MOV, W, FPS, TFIN, AO, GAMMA)

Description

Estimation of the temporal parameters of a Temporal Boolean model.

Arguments

mov a movie with a realization of a Temporal Boolean Model. W the observation window. Fps frames per seconds. Tfin estimation range for the α -function is [0:1/fps:Tfin]. a0 area of the primary grain. gamma mean number of germs per unit area in any frame.

Value

It returns a structure with the estimation of the of λ , $\alpha(s)$, p(s), $\beta(s)$, the probability density function of the durations and the time interval. The fields of the structure are denoted as Lambda, Alpha, Beta, Ps, Durationspdf and Time.

Details

The value of λ (mean number of germs per unit area and time) is estimated by fitting a first order polynomial at $\alpha(0)$ and calculating its slope using the first four points, see [1].

1.5.10 Function tracsvolumefraction.m

F = TRACSVOLUMEFRACTION(MOV)

Description

Estimation of the spatial-temporal volume fraction of a realization of a Temporal Random Closed Sets (TRACS).

Arguments

 \mathbf{MOV} the movie.

Value

This function returns a scalar, the spatial-temporal volume fraction.

1.5.11 Function tracsisolatedclumps.m

DATA = TRACSISOLATEDCLUMPS(MOV, THECC)

Description

This function segments the isolated circular clumps, i.e. clumps which boundary is not covered by any other clump/grain.

Arguments

 \mathbf{MOV} the movie.

THECC threshold for the eccentricity that is applied to filter the circular shapes.

Value

This function returns a structure with two fields. DATA.POINTS is a structure with the information of each isolated clump: X,Y is the location of the center, RAD is the radius of the disk, LIFE is a vector with the frames in which the grain is alive and N is the grain duration (in frames). DATA.MOV is a movie in which the isolated grains segmented are represented in light grey.

1.6 Visualization of a TRACS

1.6.1 Function tracs3dplot.m

TRACS3DPLOT(MOV,TIME,COLOR)

Description

This functions plots a three-dimensional reconstruction of a Temporal Random Set.

Arguments

mov a movie with a realization of a Temporal Random Set.Time observation time interval.Color a vector with the red, green and blue components.

Value

None.

1.6.2 Function tracs2dplot.m

TRACS2DPLOT (MOV, FRAME)

Description

This functions plots a frame (section) of a Temporal Random Set.



Figure 1.3: A spatial-temporal reconstruction of a TBM using function tracs3dplot

Arguments

mov a movie with a realization of a Temporal Random Set. **frame** frame to be visualized.

Value

None.

1.7 A demo

An example of simulation and estimation of the parameters of a TBM.

```
W = ones (256,256);

T = 30;

Fps = 2;

S = [0:1/Fps:T];

lambda = 0.0002;

shape = 'disk';

typesize = 'uniform';

paramsize = [8 \ 12];
```

```
typeduration = 'uniform';
paramduration = \begin{bmatrix} 6 & 12 \end{bmatrix};
mov = \ldots
tbmsimulation (S,W,T, lambda, shape, typesize, paramsize, typeduration, paramduration)
movie2avi(mov, 'temporalBM.avi', 'FPS', 1, 'COMPRESSION', 'none');
method = 'ContMinT';
Ktype ='square';
resSpa = tbmspatialestimation (mov, W, method, Ktype);
Tfin = 20; % in seconds, to calculate alpha-function
resTemp = tbmtemporalestimation(mov, W, Fps, Tfin, resSpa.Area(1), resSpa.Gamma(1));
%Showing results
resSpa.Gamma(1)
resSpa.Perimeter(1)
resSpa.Area(1)
resSpa.Vf(1)
%smoothing the pdf of the durations
resTemp.Durationspdf = smooth(resTemp.Durationspdf,'rlowess');
figure (1);
plot(resTemp.Time, resTemp.Durationspdf, 'LineWidth', 2);
xlabel('Time (seconds)', 'FontSize', 14);
ylabel('pdf durations', 'FontSize',14);
figure (2);
plot (resTemp.Time, resTemp.Alpha(:,1), 'b', 'LineWidth',2);
hold on;
plot (resTemp.Time, resTemp.Beta(:,1), 'r', 'LineWidth',2);
plot (resTemp.Time, resTemp.Beta(:,1) + resTemp.Alpha(:,1), 'g', 'LineWidth',2);
legend(' \land alpha(t)', ' \land beta(t)', ' \land gamma')
xlabel('Time (seconds)', 'FontSize',14);
% estimating the mean duration
ED = trapz (resTemp.Time, resTemp.Time.*resTemp.Durationspdf)
```

%value of the spatio-temporal intensity of the germ process:

%
the mean number of germs per unit area and time res
Temp.Lambda

%to visualize the first frame of the sequence image(mov(1).cdata), colormap(gray(2)); axis off



Figure 1.4: Several consecutive frames of a simulated TBM with cylindrical grains



Figure 1.5: (a) Estimation of $\alpha(t)$, $\beta(t)$ and γ . (b) Estimated pdf of the durations

Chapter 2

Temporal Boolean Functions

Let $\{(x_i, t_i, d_i)\}_{i \ge 1}$ be such that $\{(x_i, t_i)\}_{i \ge 1}$ is a spatial-temporal point process (the locations and the birth times) and $\{d_i\}_{i \ge 1}$ a sequence of positive random variables (the durations). Let $\{f_i\}_{i \ge 1}$ be a sequence of random functions (from \mathbb{R}^2 to \mathbb{R}) independent and identically distributed (as f_0) almost surely upper semicontinuous and such that $\{x : f_i(x) = z\}$ is almost surely compact. Then we can consider

$$f(x,t) = \sup_{i:t_i \le t \le t_i + d_i} f_i(x - x_i).$$
(2.1)

We call this function a random temporal germ-grain function, where $\{f_i\}_{i\geq 1}$ are the primary grains, $\{(x_i, t_i)\}_{i\geq 1}$ the germs and $\{d_i\}_{i\geq 1}$ the durations.

We can define a pair (f, g) of such kind of random functions by considering $\left[\{(x_i, t_i, d_i)\}_{i \ge 1}; \{(y_j, s_j, e_j)\}_{j \ge 1}\right]$ where f is defined in (2.1) and $g(x, t) = \sup_{j:s_j \le t \le s_j + e_j} g_j(x - y_j)$. We introduce dependencies between the random functions f and g by generating dependencies between the germs and the birth times.

2.1 Simulation of Temporal Boolean Functions

2.1.1 Function tbfsimulation.m

TBF = TBFSIMULATION(S,WO,T,LAMBDA,SHAPE,TYPESIZE,PARAMSIZE,TYPEDURATION,PARAMDURATION)

Description

Simulation of a Temporal Boolean Function (TBF).

Arguments

S vector with the sampling times (seconds).

W0 binary image which defines the observation window (pixels).

T length of the time interval (seconds).



Figure 2.1: A realization of a simulated bivariate temporal Boolean function . (a)-(c) Three consecutive temporal cross-sections. The red color corresponds to events of type 1, the green color to events of type 2. A linked-pairs model for the germs was used in order to generate spatial-temporal dependencies between germs types.

LAMBDA mean number of germs per unit area and unit time, i.e, intensity of the Poisson process. SHAPE shape of the grains. Possible values are: 'ball' semi-ellipsoids are generated with radii and heights specified by TYPESIZE and PARAMSIZE.

TYPESIZE the probability density function of the grains sizes and heights. Possible values are: 'uniform' sizes and heights of the grains are uniformly distributed; 'gaussian' sizes and heights of the grains are normally distributed. Truncated Gaussian.

PARAMSIZE parameters of **TYPESIZE**. If **TYPESIZE** is uniform, it is a 4-element vector with [minradius maxradius minheight maxheight]. If **TYPESIZE** is gaussian, it is a 4-element vector [meanradius stdradius meanheight stdheight].

TYPEDURATION is the distribution of the grains durations. Possible values are: 'exponential' (**PARAMDURATION** is the mean); 'uniform' (**PARAMDURATION** is a vector with the inferior and superior limit of the interval); 'gaussian' (**PARAMDURATION** is a vector with the mean and standard deviation). Truncated Gaussian.

Value

TBF. This function returns an AVI file with the frames corresponding to a realization of a Temporal Boolean Function at times given in S.

Details

See also tbmsimulation.

2.1.2 Function tbfgrains.m

[SIZESGR_N, HEIGHTSGR_N] = TBFGRAINS(N, TYPEPDF, PARAM)

Description

Generation of the grains sizes and heights of a Temporal Boolean Function.

Arguments

N number of grains.

TYPEPDF the distribution of the grains sizes and heights. Possible values are: 'uniform' (**PARAM** is a 4-element vector [minradius maxradius minheight maxheight]); 'gaussian' (**PARAM** is a 4-element vector [meanradius stdradius meanheight stdheight]). Truncated Gaussian.

Value

This function returns two vectors SIZESGR_N, HEIGHTSGR_N with N elements each. SIZESGR_N and HEIGHTSGR_N are distributed as **TYPEPDF** with parameters **PARAM**. Grains are semi-ellipsoids.

Details

See also tbmgrains.

2.1.3 Function tbfbivariatesimulation.m

BITBF = TBFBIVARIATESIMULATION(S, Wo, T, LAMBDA, SHAPE, TYPESIZE, PARAMSIZE, ... TYPEDURAT PARAMDURATION, TYPEMODEL, PARAMMODEL, TYPEDEPENDENCIES, PARAMDEPENDENCIES)

Description

Generation of a bivariate Temporal Boolean Function.

Arguments

S vector with the sampling times (seconds).

Wo binary image with the observation window (pixels).

T length of the time interval (seconds).

LAMBDA mean number of germs per unit area and unit time, i.e, intensity of the Poisson process (the parents process in the case of a cluster Poisson).

SHAPE shape of the grain. Possible values are: 'ball' semi-ellipsoids are generated with parameters specified by **TYPESIZE** and **PARAMSIZE**.

TYPESIZE probability density function of the grain sizes. Possible values are: 'uniform' grain sizes and heights are uniformly distributed; 'gaussian' grain sizes and heights are normally dis-

tributed. Truncated Gaussian.

PARAMSIZE parameters of **TYPESIZE**. If **TYPESIZE** is uniform, **PARAMSIZE** is a 8element vector [minradius maxradius mingreylevel maxgreylevel] for component-1 and component-2. If **TYPESIZE** is gaussian, **PARAMSIZE** is a 8-element vector [meanradius stdradius meanheights stdheights] for component-1 and component-2. Truncated Gaussian.

TYPEDURATION the distribution of the duration of the grains (seconds). Possible values are: 'exponential' (**PARAMDURATION** is the mean); 'uniform' (**PARAMDURATION** is a vector with the inferior and superior limits of the interval); 'gaussian' (**PARAMDURATION** is a vector with the mean and standard deviation, truncated Gaussian); 'gamma' (**PARAMDURA-TION** is a vector with the shape and scale parameter).

TYPEMODEL the type of interaction model for the germs. Possible values are: 'linked-pairs' (also known as Neymann-Scott process); 'clusterPoisson' parents are the type-1 events and the offsprings are the type-2 events.

PARAMMODEL if 'linked-pairs' **PARAMMODEL** is null [], if clusterPoisson **PARAM-MODEL** is a 2-element vector with the maximum size of a cluster and the number of offsprings.

TYPEDEPENDENCIES type of dependencies between components. **PARAMDEPENDEN-CIES** gives the parameters to uniformly shift the type-1 germs with respect to type-2 germs. Possible values are: 'spatial' (if the model is 'linked-pairs' **PARAMDEPENDENCIES** is a 2-element vector with [rmin, rmax] (in pixels) and if the model is 'clusterPoisson' **PARAMDE-PENDENCIES** is null [], since the the spatial dependencies are given in **PARAMMODEL**); 'spatial-temporal' (if the model is 'linked-pairs' **PARAMDEPENDENCIES** is a 4-element vector [rmin, rmax, tmin, tmax]) and if the model is 'clusterPoisson' **PARAMDEPENDENCIES** is a 2-element vector [tmin, tmax] since the the spatial dependencies are given in **PARAMMODEL**).

Value

This function returns a struct with three fields (three AVI movies): MOV1, MOV2, MOV12. MOV1 is type-1, MOV2 type-2 and MOV12 is the union (type-1 events are in red and type-2 events are in green).

Dependencies

R software (http://cran.r-project.org/) and spatstats package. See chapter 4 for details.

Chapter 3

Bivariate Temporal Random Sets

3.1 Bivariate Temporal Random Sets

3.1.1 Definition

If Φ_1 and Φ_2 are the temporal random sets associated with the first and second event types respectively, then the random set defined as $\mathbf{\Phi} = (\Phi_1, \Phi_2)$ is a *bivariate temporal random set*, where Φ_i is the *i*-th component. The distribution of $\mathbf{\Phi} = (\Phi_1, \Phi_2)$ is characterized by the probability $P(\Phi_1 \uparrow K_1, \Phi_2 \uparrow K_2)$, i.e. the probability that the temporal random set Φ_1 hits K_1 and the temporal random set Φ_2 hits K_2 simultaneously, being K_1 and K_2 compact subsets of $\mathbb{R}^2 \times \mathbb{R}_+$. Fig. 3.1 shows a realization of a bivariate temporal random set.

Our data consist of a pair of sequences of binary images which have been simultaneously captured, i.e. two discrete sets of equally spaced temporal cross-sections observed through a sampling time interval T and within a fixed sampling window W. Let ϕ denote the corresponding realization of Φ and $\phi(t_i)$, the area covered at time t_i . If t_1, t_2, \ldots, t_n are the sampling times, then we will observe $\phi(t_i) \cap W$, where $i = 1, \ldots, n$.

Throughout this paper we assume that (Φ_1, Φ_2) is jointly stationary, i.e. its joint distribution is invariant against spatial-temporal translations in $\mathbb{R}^2 \times \mathbb{R}_+$. This implies that Φ_1 and Φ_2 are marginally stationary. This assumption is a natural and pragmatic simplification, which is tenable in many real applications and justifies the use of relatively simple non-parametric summaries.

Under stationarity the volume fractions can be considered separately for each component $p_i = P(\mathbf{0} \in \Phi_i(0))$, for i = 1, 2, where the spatial origin $\mathbf{0}$ denotes an arbitrary location and the temporal origin 0 denotes an arbitrary time. Note that p_i gives the probability that an arbitrary point at an arbitrary time belongs to the *i*-th component of the random set and can be interpreted as the mean volume covered by the *i*-th temporal random set per area and unit time.



Figure 3.1: A realization of a simulated bivariate temporal random set. (a)-(c) Three consecutive temporal cross-sections. The white color corresponds to events of type 1, the light grey to events of type 2, and the dark grey to overlapping regions.



Figure 3.2: A spatial-temporal reconstruction: the red color corresponds to type-1 events and the green color to type-2 events

3.1.2 Summary statistics for bivariate temporal random sets

We define a generalized version of the Ripley K-function, the covariance and the pair correlation functions for a bivariate temporal random set, $\mathbf{\Phi} = (\Phi_1, \Phi_2)$. For a given $\mathbf{h} \in \mathbb{R}^2$ and a given time $t \in \mathbb{R}_+$, we define the *spatial-temporal covariance function* as

$$\mathbb{C}_{ij}(\mathbf{h},t) = P(\mathbf{0} \in \Phi_i(0), \mathbf{h} \in \Phi_j(t)), \tag{3.1}$$

where i, j = 1, 2. This is a natural and direct generalization of the covariance of a random set (details of various theoretical aspects can be found in [6,8,9]). The spatial-temporal covariance function gives us the probability that an arbitrary point observed at an arbitrary time belongs to the *i*-th component of the temporal random set and its translation by a vector **h** and a time *t* belongs to the *j*-th component.

Throughout this paper we assume that (Φ_1, Φ_2) are jointly spatial (but not temporal) isotropic, i.e. their joint distribution is invariant against spatial rotations. Under spatial stationarity and isotropy, the covariance function \mathbb{C}_{ij} only depends on the modulus of \mathbf{h} , $s = \|\mathbf{h}\|$, and on t.

The cross \mathbb{K} -function of a bivariate temporal random set can be defined as

 $p_j \mathbb{K}_{ij}(s,t) = E[$ volume covered by the *j*-th component within a cylinder of radius *s* and height 2*t* centered at a randomly chosen point of the *i*-th component],

where p_j is the volume fraction of Φ_j .

More formally, it can be expressed as $p_j \mathbb{K}_{ij}(s,t) = E[\nu_3(\Phi_j \cap B(\mathbf{0},s) \times [-t,t]) | \mathbf{0} \in \Phi_i(\mathbf{0})]$, with $s,t \ge 0$, where ν_3 stands for the volume and $B(\mathbf{0},s)$ is the disk centered at an arbitrary location of Φ_i with radius s. From the covariance function, the cross \mathbb{K} -function can be computed as

$$\mathbb{K}_{ij}(s,t) = \int_{-t}^{t} \int_{B(\mathbf{0},s)} \frac{\mathbb{C}_{ij}(\mathbf{u},v)}{p_i p_j} d\mathbf{u} dv, \qquad (3.2)$$

with i, j = 1, 2. From (3.1) and (3.2), it holds that $\mathbb{K}_{ij}(s, t) = \mathbb{K}_{ji}(s, t)$.

Finally, we define the *pair correlation function* of the bivariate temporal random set as

$$g_{ij}(s,t) = \frac{\mathbb{C}_{ij}(s,t)}{p_i p_j}.$$
(3.3)

3.1.3 Estimators

Essentially, we replace the (unknown) continuous versions of p_i , \mathbb{C}_{ij} and \mathbb{K}_{ij} with the corresponding (known) discrete versions. In order to simplify the notation, we assume that the sampling times are $t = 0, \delta, 2\delta, \ldots, n\delta$; i.e. n + 1 equally spaced frames, where δ is the temporal delay between two consecutive frames. Our sampling information consists of a sequence of sets $\{(\phi_i(k\delta) \cap W, \phi_j(k\delta) \cap W)\}_{k=0,\ldots,n}$.

A natural estimator for p_i is the mean of the area fractions observed at each frame of the sequence

$$\hat{p}_i = \frac{1}{n+1} \sum_{k=0}^n \frac{\nu_2[\phi_i(k\delta) \cap W]}{\nu_2[W]},\tag{3.4}$$

where ν_2 stands for the area. We will estimate the covariance as

$$\hat{\mathbb{C}}_{ij}(\mathbf{h}, v\delta) = \frac{1}{n-v+1} \sum_{k=0}^{n-v} \frac{\nu_2 \Big[(\phi_i(k\delta) \cap W) \cap (\phi_j((k+v)\delta) \cap W) - \mathbf{h} \Big]}{\nu_2 [W \cap W - \mathbf{h}]}.$$
(3.5)

We have adopted the minus sampling method [9] to correct edge effects. If the covariance function does not depend on time, we can estimate it as

$$\hat{\mathbb{C}}_{ij}(\mathbf{h},0) = \frac{1}{n+1} \sum_{k=0}^{n} \hat{\mathbb{C}}_{ij}(\mathbf{h},k\delta).$$
(3.6)

Likewise, if there is no spatial dependence, the following estimator can be used

$$\hat{\mathbb{C}}_{ij}(\mathbf{0},k\delta) = \frac{1}{\nu_2(W)} \int_W \hat{\mathbb{C}}_{ij}(\mathbf{h},k\delta) d\mathbf{h}.$$
(3.7)

Finally, $\mathbb{K}_{ij}(s,t)$ and $g_{ij}(s,t)$ will be estimated by replacing the volume fraction and the covariance function in (3.2) and (3.3) with their corresponding estimators given in (3.4) and (3.5).

3.2 Testing spatial and spatial-temporal dependencies

The functions \mathbb{C}_{ij} , g_{ij} and \mathbb{K}_{ij} are used to describe the joint distribution and to test different null hypotheses. Firstly, we test whether there are spatial dependencies. Secondly, if locations and times are jointly analyzed, we will test spatial-temporal interactions.

Under complete independence, the covariance function is

$$\mathbb{C}_{ij}(\mathbf{h},t) = P(\mathbf{0} \in \Phi_i(0), \mathbf{h} \in \Phi_j(t)) = P(\mathbf{0} \in \Phi_i(0))P(\mathbf{h} \in \Phi_j(t)) = p_i p_j.$$
(3.8)

In such a case, from (3.3) it holds that $g_{ij}(s,t) = 1$. Values of $g_{ij}(s,t)$ greater than one represent a positive dependence (attraction) and lesser than one represent a negative dependence (inhibition). Likewise, from (3.2) we have $\mathbb{K}_{ij}(s,t) = 2\pi s^2 t$.

For temporally independent random sets, the covariance function does not depend on time and can be computed as

$$\mathbb{K}_{ij}(s,t) = 4\pi t \int_0^s \frac{r\mathbb{C}_{ij}(r,0)}{p_i p_j} dr.$$
(3.9)

If there is temporal dependence but no spatial association, the covariance function only depends on time and is calculated as $(2 - 2)^{-1}$

$$\mathbb{K}_{ij}(s,t) = \pi s^2 \int_{-t}^t \frac{\mathbb{C}_{ij}(\mathbf{0},v)}{p_i p_j} dv.$$
(3.10)

Let us consider the following marginal function:

$$\mathbb{K}_{ij}^{(1)}(s) = \int_{B(\mathbf{0},s)} \frac{\mathbb{C}_{ij}(\mathbf{u},0)}{p_i p_j} d\mathbf{u}.$$
(3.11)

Then, under spatial independence $\mathbb{K}_{ij}^{(1)}(s) = \pi s^2$.

3.2.1 Testing null hypotheses by using toroidal shift randomizations

Testing null hypotheses can be achieved by means of toroidal shift methods, which provide a nonparametric way to test complete independence. For convenience, we assume that the sampling window W is a rectangle. For a given $\mathbf{h} \in W$, we denote by $T_{\mathbf{h}}(A)$ the toroidal shift (with respect to W) of any subset A of W. A toroidal shift is a simultaneous, parallel shift of all points in the set by the same randomly chosen shift vector \mathbf{h} . The rectangle W is treated as a torus and the set A is shifted within this torus. This method can be extended to the temporal dimension as follows. Let us consider

$$T_{(\mathbf{h},d\delta)}(\phi_i(k\delta) \cap W) = \begin{cases} T_h(\phi_i((k+d)\delta) & \text{if } d = 0,\dots, n-k, \\ T_h(\phi_i((k+d-n)\delta) & \text{if } d = n-k+1,\dots, n, \end{cases}$$
(3.12)

where d = 0, ..., n - 1, being *n* the total number of frames. Let us consider the random vector (\mathbf{H}, D) such that \mathbf{H} (distance) and D (lag) are independent; \mathbf{H} is uniform in $W, \mathbf{H} \sim Unif(W)$; and D is uniform in $\{0, ..., n - 1\}$. From the original observed image sequence $\{(\phi_1(k\delta) \cap W, \phi_2(k\delta) \cap W)\}_{k=1,...,n}$, we generate $\mathbf{H} = \mathbf{h}$ and D = d obtaining a randomly modified sequence $\{(\phi_1(k\delta) \cap W, \phi_2(k\delta) \cap W)\}_{k=1,...,n}$ in which the second component has been spatially and temporally shifted over the first one.

Let us see a short explanation of the testing procedure using the cross \mathbb{K}_{ij} -function. First, the cross \mathbb{K} -function is calculated for the original pair of binary image sequences, giving us $\hat{\mathbb{K}}_{ij,0}$. Next, for m randomized pairs in which we keep the original sequence of type 1 events and randomize type 2 events, we obtain $\hat{\mathbb{K}}_{ij,n}$ with $n = 1, \ldots, m$. A graphical analysis to test independence can be performed by displaying $\hat{\mathbb{K}}_{ij,0}$ along with the lower and upper envelopes obtained from the m functions associated with the randomizations. The region delimited by both envelopes quantifies the variability when independence between components is assumed. The function $\hat{\mathbb{K}}_{ij,0}$ should be contained within both envelopes under the null hypothesis of independence between event types. A similar approach is used to test spatial independence. In this case, the random variable D is chosen to degenerate at 0.

Apart from the graphical test, a *p*-value corresponding to a Monte Carlo test can be computed. Independence would imply that the sampling distribution of $\hat{\mathbb{K}}_{ij}$ is invariant to random toroidal shifts. We can conduct a Monte Carlo test of independence by computing a suitable test statistic and comparing the value calculated for $\hat{\mathbb{K}}_{ij,0}$ with the values obtained from the randomizations $\hat{\mathbb{K}}_{ij,n}$, with n = 1, ..., m. This methodology was proposed in [3] and applied there to point processes. Under the hypothesis of independence

$$P(\mathbb{K}_{ij,0}(s,t) > U(s,t)) = P(\mathbb{K}_{ij,0}(s,t) < L(s,t)) = \frac{1}{m+1},$$
(3.13)

where the lower envelope is defined as $L(s,t) = \min_{n=1,\dots,m} \hat{\mathbb{K}}_{ij,n}(s,t)$ and the upper envelope is $U(s,t) = \max_{n=1,\dots,m} \hat{\mathbb{K}}_{ij,n}(s,t)$.

An exact test can be performed using the test statistic $d_n = \int_0^{+\infty} (\mathbb{K}_{ij,n}(s,t) - \overline{\mathbb{K}}_{ij,n}(s,t))^2 ds dt$, where $\overline{\mathbb{K}}_{ij,n}(s,t) = \sum_{r=0,r\neq n}^m \frac{\mathbb{K}_{ij,r}(s,t)}{m}$. All the rankings of d_0 are equiprobable under the null hypothesis. If $d_{(j)}$ denotes the *j*-th largest among d_n , with $n = 0, \ldots, m$, then, $P(d_0 = d_{(j)}) = \frac{1}{m+1}$ with $j = 0, \ldots, m$, and rejection of the null hypothesis on the basis that d_0 ranks the *k*-th largest or higher gives an exact one-sided test with *p*-value equal to $1 - \frac{k}{m+1}$.

3.2.2 Estimating the spatio-temporal interval of dependencies

Let us consider the random events $\{\mathbf{0} \in \Phi_1(0)\}$ and $\{\mathbf{h} \in \Phi_2(t)\}$ and let us denote $s = \|\mathbf{h}\|$. Supposing that we have rejected the null hypothesis of complete independence, then from (3.8) it holds that there is a pair of values (s,t) such that $\mathbb{C}_{ij}(s,t) - p_i p_j \neq 0$ and therefore $\{\mathbf{0} \in \Phi_1(0)\}$ and $\{\mathbf{h} \in \Phi_2(t)\}$ are not independent. Note that this does not imply that the same statement is true for any pair (s,t). In fact in many real applications there are only local dependencies in such a way that for large s and t these two events $\{\mathbf{0} \in \Phi_1(0)\}$ and $\{\mathbf{h} \in \Phi_2(t)\}$ will be independent. For a given s it is expected that the events will be independent for t values larger than a fixed point that depends on s.

Given (s_0, t_0) , let (h_n, t_n) (with n = 1, ..., m) be a random sample uniformly distributed in $[B(0, s_0 + \delta_1) \setminus B(0, s_0 - \delta_1)] \times [t_0 - \delta_2, t_0 + \delta_2]$ where $\delta_1, \delta_2 > 0$. For every translation vector (h_n, t_n) , we perform a toroidal shift of ϕ_1 over ϕ_2 and estimate the covariance functions from these new pair of sequences, $\hat{\mathbb{C}}_{ij,n}(h_n, t_n)$. Then, a two-sided Monte Carlo test applied to the difference $\hat{\mathbb{C}}_{ij,n}(h_n, t_n) - \hat{p}_i \hat{p}_j$ provides us with a *p*-value. By applying this procedure to different values of (s_0, t_0) we have a function $p(s_0, t_0)$, a map of *p*-values. This map is a graphical representation of the *p*-values providing at which distances and temporal lags the two components show significant dependencies.

3.3 Functions for the analysis of dependencies between types in bivariate TRACS

3.3.1 Function tbmbivariatesimulation

BITBM = TBMBIVARIATESIMULATION(S,Wo,T,LAMBDA,SHAPE,TYPESIZE,PARAMSIZE,... TYPEDURATION,PARAMDURATION,TYPEMODEL,PARAMMODEL,TYPEDEPENDENCIES,PARAMDEPENDENCIES)

Description

Generation of a bivariate temporal Boolean model.

Arguments

S vector with the sampling times (seconds).

Wo binary image with the observation window (pixels).

T length of the time interval (seconds).

LAMBDA mean number of germs per unit area and time, i.e, intensity of the Poisson process (the parents process in the case of a cluster Poisson).

SHAPE shape of the primary grain. Possible values are: 'disk' disks are generated with radii distributed as **TYPESIZE**; 'square' squares are generated with side length distributed as **TYPE-SIZE**.

TYPESIZE the distribution of the grain sizes. Possible values are: 'uniform' (**PARAMSIZE** is a vector with the inferior and superior limits of the interval); 'gaussian' (**PARAMSIZE** is a vector with the mean and standard deviation, Ttruncated Gaussian).

TYPEDURATION the distribution of the duration of the grains (seconds). Possible values are: 'exponential'(**PARAMDURATION** is the mean); 'uniform' (**PARAMDURATION** is inferior and superior limits of the interval); 'gaussian' (**PARAMDURATION** is the mean and standard deviation, truncated Gaussian); 'gamma' (**PARAMDURATION** is the shape and scale parameter).

TYPEMODEL the type of interaction model for the germs. Possible values are: 'linked-pairs' (also known as Neymann-Scott process); 'clusterPoisson' parents are the type-1 events and the offsprings are the type-2 events.

PARAMMODEL if 'linked-pairs' **PARAMMODEL** is NULL [], if clusterPoisson **PARAM-MODEL** is a 2-element vector with the maximum size of a cluster and the number of offsprings.

TYPEDEPENDENCIES type of dependencies between components. **PARAMDEPENDEN-CIES** gives the parameters to uniformly shift the type-1 germs with respect to type-2 germs. Possible values are: 'spatial' (if the model is 'linked-pairs' **PARAMDEPENDENCIES** is a 2-element vector with [rmin, rmax] (in pixels) and if the model is 'clusterPoisson' PARAMDEPENDENCIES is null [], since the the spatial dependencies are given in **PARAMMODEL**), 'spatial-temporal' (if the model is 'linked-pairs' **PARAMDEPENDENCIES** is a 4-element vector [rmin, rmax, tmin, tmax]) and if the model is 'clusterPoisson', **PARAMDEPENDENCIES** is a 2-element vector [tmin, tmax] since the the spatial dependencies are given in **PARAMMODEL**).

Value

This function returns a struct with three fields (three AVI movies) denoted as: mov1, mov2, mov12. MOV1 is type-1, MOV2 type-2 and MOV12 is the union. Type-1 events are in white, type-2 events are in light-gray and overlapping areas in dark gray.

Dependencies

R software http://cran.r-project.org/) and spatstat package. See chapter 4 for details.

3.3.2 Function tracsrandomization.m

```
MOVSHIFTED = TRACSRANDOMIZATION(MOV,H,T)
```

Description

Toroidal shift randomization of a TRACS.

Arguments

MOV a movie with the realization of a TRACS.H a 2-dimensional vector with the spatial shift [hx hy] (pixels).T the temporal shift (frames).

Value

This function returns an AVI file (binary or grey-level image sequence).

3.3.3 Function tracscrossC.m

C = TRACSCROSSC(S,T,MOV1,MOV2)

Description

Cross-covariance of a TRACS.

Arguments

 \mathbf{S} vector with the radii of disks where evaluate the cross-covariance.

T vector with the frames to evaluate the cross-covariance or an scalar with value T=0 (case in which the covariance does not depend on time). T must have an odd number of elements and must take the value 0, for example T=[-10:2:10]. We assume both sequences have the same length in frames and size in pixels.

Value

This function returns the cross-covariance within a disk of radius given in S and time interval T (in frames) for movies MOV1, MOV2. C is integrated over the angles [-pi:pi:pi].

3.3.4 Function tracscrossCh0.m

C = TRACSCROSSCHO(H,MOV1,MOV2)

Description Cross-covariance of a TRACS.

Arguments

H the spatial shift in polar coordinates (pixels, radians).MOV1 movie of the first component of the bivariate TRACS.MOV2 movie of the second component of the bivariate TRACS.

Value

This function returns the cross-covariance given a vector displacement H for movies MOV1, MOV2. C is an scalar, the value of the cross-covariance at (H,0). C is integrated over the angles [-pi:pi:pi]. The estimation method depends on the type of dependencies: spatial or spatial-temporal. Here, we assume there is only spatial dependencies.

Details

We assume both sequences have the same length in frames and size in pixels.

3.3.5 Function tracscrossCht.m

C = TRACSCROSSCHT(H,T,MOV1,MOV2)

Description

Cross-covariance of a TRACS.

Arguments

H the spatial shift in polar coordinates (pixels, radians).
T the temporal shift (frames).
MOV1 movie of the first component of the bivariate TRACS.
MOV2 movie of the second component of the bivariate TRACS.

Value

This function returns the cross-covariance given a vector displacement H and time T (frames) for movies MOV1, MOV2. C is an scalar, the value of the cross-covariance at (H,T).

Details

We assume that both sequences have the same length in frames and size in pixels.

3.3.6 Function tracscrossK1.m

K1 = TRACSCROSSK1(S,C,FVI,FVJ)

Description

The K1-function within a disk of radii given in S.

Arguments

 \mathbf{S} a vector with the radii of a disk where K1 is evaluated. \mathbf{C} the cross-covariance, a vector of size $1 \times \text{length}(S)$ $\mathbf{FV1}$ the volume fraction of the 1st-component. $\mathbf{FV2}$ the volume fraction of the 2nd-component.

Value

K1, a vector of size length(S), with the values of K1 within a disk of radii given in S.

3.3.7 Function tracscrossK.m

K = TRACSCROSSK(S,T,C,FVI,FVJ)

Description

The K-function within a disk of radii given in S (pixels) and time interval given in T (frames).

Arguments

S vector with the radii of a disk (pixels) where K is evaluated.

T vector with the frames where K is evaluated, for example T = [-10:2:10].

 ${\bf C}$ the cross-covariance.

 ${\bf FV1}$ the volume fraction of the 1st-component.

 ${\bf FV2}$ the volume fraction of the 2nd-component.

Value

This function returns a matrix with K.

3.3.8 Function tracspvalueMC.m

```
pvalue = TRACSPVALUEMC(F, NREP, S)
```

Description

This function provides the p-value of the Monte Carlo test for descriptor F.

Arguments

F is a matrix with the value of the descriptors for the observed model and NREP replica. Rows correspond to the replica. The observed model is assumed to be the first row. **NREP** number of replica.

S the radii of disks (pixels) where F is evaluated.

```
pvalue = TRACSPVALUEMC(F, NREP, S,T)
```

 ${\bf T}$ a vector with the frames where K is evaluated.

Value

The p-value.

3.3.9 Function tracspvaluemapMC.m

```
pvaluemap = TRACSPVALUEMAPMC(S,T,MOV1, MOV2, NREP)
```

Description

This function calculates and plots the p-values of the Monte Carlo test by applying a bootstrap procedure to the cross-covariance.

Arguments

S vector with the radii of the disks where the cross-covariance is evaluated.
T vector with the frames where the cross-covariance is evaluated.
MOV1, MOV2 the movies.
NREP number of replica.

Value

A matrix with the maps of p-values for movies MOV1 and MOV2 evaluated within a disk with radii given in S (pixels) and frames given in T for NREP replica for the cross-covariance.

3.3.10 Demos

The scripts ScriptSpatial.m and ScriptSpatialTemporal.m show how to calculate the covariance function, the K-function, L-function and the pair correlation functions.

Calculating the p-value map

```
W = ones(256, 256);
T = 40;
fps = 2;
s = [1:1/fps:T];
lambda = 0.0001;
shape = 'disk';
typesize = 'uniform';
paramsize = \begin{bmatrix} 5 & 5 \end{bmatrix};
typeduration = 'uniform ';
paramduration = \begin{bmatrix} 6 & 6 \end{bmatrix};
typemodel = 'linked-pairs';
parammodel = [];
typedependencies = 'spatial-temporal';
paramtypedependencies = \begin{bmatrix} 0 & 6 & 0 & 4 \end{bmatrix};
bimov = tbmbivariatesimulation (s, W, T, lambda, shape, typesize, paramsize, ...
                                     typeduration, paramduration,...
                                     typemodel, parammodel, typedependencies,...
                                     paramtypedependencies);
srad = [0:2:20];
t = [0:2:30];
nrep = 40;
v = [0.01 \ 0.025 \ 0.05 \ 0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \ 0.7 \ 0.8 \ 0.9];
pvalues=tracspvaluemapMC(srad,t,bomov.mov1, bimov.mov2,nrep);
figure;
[c,h] = contour(t, srad, pvalues, v); clabel(c,h), colorbar;
ylabel('s (pixels)', 'FontSize',18);
xlabel('t (frames)', 'FontSize', 18);
dlmwrite ('pvaluesst -1.txt', pvalues, '');
```



Figure 3.3:

3.4 Visualization of a bivariate-TRACS

3.4.1 Function tracsbi3dplot.m

TRACSBI3DPLOT(MOV1,MOV2,TIME,COLOR1, COLOR2)

Description

Plot a three-dimensional reconstruction of a bivariate TRACS.

Arguments

MOV1, MOV2 movies of a realization of a bivariate TRACS.TIME observation time interval.COLOR1, COLOR2 a vector with the red, green and blue components.

Value

None.

3.5 Functions for the analysis of dependencies between types in bivariate fuzzy TRACS

3.5.1 Function tracsfuzzygetalpha.m

ALPHA = TRACSFUZZYGETALPHA(MOV,BETA)

Description

This function provides the Vorobev alpha-cuts.

Arguments

MOV the movie, MOV is a grey-level image sequence. **BETA** the Vorobev α -cuts are calculated as the mean + BETA times the standard deviation.

Value

This function returns the Vorobev α -cuts calculated as the mean + BETA times the standard deviation. ALPHA is a matrix of size length(MOVIE) × length(BETA).

Dependencies

R software and fda and splines packages ttp://cran.r-project.org/}\subsection{Demos}Te scripts ScriptSpatialFUZZY.m and ScriptSpatialTemporalFUZZY.m are the respective fuzzy versions of the covariance function, the K-function, L-function and the pair correlation function.

Chapter 4

Installation

This package consists of a set of m-files compressed into a zip file. Unzip them to a folder and then add this folder to Matlab's path.

Any type of comments or bug reports are appreciated. Please send your comments to elena.diaz@uv.es.

Functions tracsfuzzygetalpha.m, tbmbivariatesimulation.m and tbfbivariatesimulation.m depend on R package, edit the mcode and replace the string $C : Archivos de programa \ R \ R - 2.6.1 \ bin \ R.exe$ with the path where R software is installed. http://cran.r-project.org/. Or edit the file tracs.conf and modify the string named RPATH.

To obtain further details of each function of the Toolbox, you can type help functionname.m in the Matlab command window.

Bibliography

- G. Ayala, R. Sebastian, M.E Díaz, E. Díaz, R. Zoncu, and D. Toomre. Analysis of spatially and temporally overlapping events with application to image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1707–1712, 2006.
- [2] E. Díaz, R. Sebastian, G. Ayala, M.E. Díaz, E. Díaz, R. Zoncu, and D. Toomre. Measuring spatio-temporal dependencies in bivariate temporal random sets with applications to cell biology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press.
- [3] P.J. Diggle. Statistical Analysis of Spatial Point Patterns. Arnold, London, second edition, 2003.
- [4] A.M. Law and W.D. Kelton. Simulation Modeling and Analysis. McGraw Hill, third edition, 2000.
- [5] I. Molchanov. Statistics of the Boolean Model for Practitioners and Mathematicians. John Wiley and Sons, Chichester, 1997.
- [6] I. Molchanov. Theory of Random Sets (Probability and its Applications). Springer, 2005.
- [7] R. Sebastian, E. Díaz, G. Ayala, M.E. Díaz, R. Zoncu, and D. Toomre. Studying endocytosis in space and time by means of temporal boolean models. *Pattern Recognition*, 39:2175–2185, 2006.
- [8] J.P. Serra. Image Analysis and Mathematical Morphology, volume 1. Academic Press, 1982.
- [9] D. Stoyan, W.S. Kendall, and J. Mecke. Stochastic Geometry and its Applications. Wiley, Berlin, second edition, 1995.