7

Spatial Point Pattern Analysis

7.1 Introduction

The analysis of point patterns appears in many different areas of research. In ecology, for example, the interest may be focused on determining the spatial distribution (and its causes) of a tree species for which the locations have been obtained within a study area. Furthermore, if two or more species have been recorded, it may also be of interest to assess whether these species are equally distributed or competition exists between them. Other factors which force each species to spread in particular areas of the study region may be studied as well. In spatial epidemiology, a common problem is to determine whether the cases of a certain disease are clustered. This can be assessed by comparing the spatial distribution of the cases to the locations of a set of controls taken at random from the population.

In this chapter, we describe how the basic steps in the analysis of point patterns can be carried out using R. When introducing new ideas and concepts we have tried to follow Diggle (2003) as much as possible because this text offers a comprehensive description of point processes and applications in many fields of research. The examples included in this chapter have also been taken from that book and we have tried to reproduce some of the examples and figures included there.

In general, a point process is a stochastic process in which we observe the locations of some *events* of interest within a bounded region A. Diggle (2003) defines a point process as a 'stochastic mechanism which generates a countable set of events'. Diggle (2003) and Möller and Waagepetersen (2003) give proper definitions of different types of a point process and their main properties. The locations of the events generated by a point process in the area of study A will be called a *point pattern*. Sometimes, additional covariates may have been recorded and they will be attached to the locations of the observed events.

Other books covering this subject include Schabenberger and Gotway (2005, Chap. 3), Waller and Gotway (2004, Chaps. 5 and 6) and O'Sullivan and Unwin (2003, Chaps. 4 and 5).

7.2 Packages for the Analysis of Spatial Point Patterns

There are a number of packages for R which implement different functions for the analysis of spatial point patterns. The **spatial** package provides functions described in Venables and Ripley (2002, pp. 430–434), and **splancs** (Rowlingson and Diggle, 1993) and **spatstat** (Baddeley and Turner, 2005) provide other implementations and additional methods for the analysis of different types of point processes. The Spatial Task View contains a complete list of all the packages available in R for the analysis of point patterns. Other packages worth mentioning include **spatialkernel**, which implements different kernel functions and methods for the analysis of multivariate point processes. Given that most of the examples included in this chapter have been computed using **splancs** and **spatstat**, we focus particularly on these packages.

These packages use different data structures to store the information of a point pattern. Given that it would be tedious to rewrite all the code included in these packages to use **sp** classes, we need a simple mechanism to convert between formats. Package **maptools** offers some functions to convert between `ppp` objects representing two-dimensional point patterns (from **spatstat**, which uses old-style classes, see p. 24) and **sp** classes. Note that, in addition to the point coordinates, `ppp` objects include the boundary of the region where the point data have been observed, whilst **sp** classes do not, and it has to be stored separately. Data types used in **splancs** are based on a two-column matrix for the coordinates of the point pattern plus a similar matrix to store the boundary; the package was written before old-style classes were introduced. Function `as.points` is provided to convert to this type of structure. Hence, it is very simple to convert the coordinates from **sp** classes to use functions included in **splancs**.

Section 2.4 describes different types of **sp** classes to work with point data. They are `SpatialPoints`, for simple point data, and `SpatialPointsDataFrame`, when additional covariates are recorded. More information and examples can be found in the referred section. Hence, it should not be difficult to have the data available in the format required for the analysis whatever package is used.

To illustrate the use of some of the different techniques available for the analysis of point patterns, we have selected some examples from forest ecology, biology, and spatial epidemiology. The point patterns in Fig. 7.1 show the spatial distribution of cell centres (left), California redwood trees (right), and Japanese black pine (middle). All data sets have been re-scaled to fit into a one-by-one square. These data sets are described in Ripley (1977), Strauss (1975), Numata (1961) and all of them have been re-analysed in Diggle (2003).

These data sets are available in package **spatstat**. This package uses `ppp` objects to store point patterns, but package **maptools** provides some functions to convert between `ppp` objects and `SpatialPoints`, as shown in the following example. First we take the Japanese black pine saplings example, measured in a square sampling region in a natural forest, reading in the data provided with **spatstat**.

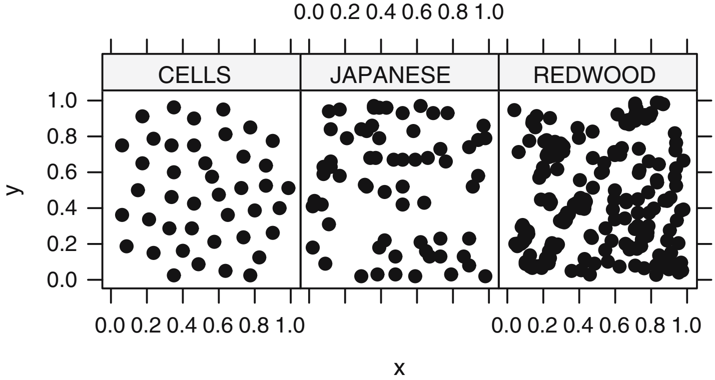

Fig. 7.1. Example of three point patterns re-scaled to fit in the unit square. On the left, spatial distribution of the location of cell centres (Ripley, 1977); in the middle, Japanese black pine saplings (Numata, 1961); and on the right, saplings of California redwood trees (Strauss, 1975)

```
> library(spatstat)
> data(japanesepines)

> summary(japanesepines)

Planar point pattern: 65 points
Average intensity 65 points per square unit (one unit = 5.7 metres)

Window: rectangle = [0, 1] x [0, 1] units
Window area =  1 square unit
Unit of length: 5.7 metres
```

The summary shows the average intensity in the region of interest; this region, known as a window, is also reported in the summary; windows are stored in objects of class `owin`. In this case, the points have been scaled to the unit square already, but the size of the sampling square can be used to retrieve the actual measurements. Note that **spatstat** windows may be of several forms, here the window is a rectangle. When we coerce a `ppp` object with a rectangular window to a `SpatialPoints` object, the point coordinates will by default be re-scaled to their original values.

```
> library(maptools)

> spjpines <- as(japanesepines, "SpatialPoints")
> summary(spjpines)

Object of class SpatialPoints
Coordinates:
     min max
[1,]   0 5.7
[2,]   0 5.7
```

```
Is projected: NA
proj4string : [NA]
Number of points: 65
```

We can get back to the unit square using the `elide` methods discussed in Chap. 5 as the summary of the output object shows.

```
> spjpines1 <- elide(spjpines, scale = TRUE, unitsq = TRUE)
> summary(spjpines1)
```

```
Object of class SpatialPoints
Coordinates:
     min max
[1,]   0   1
[2,]   0   1
Is projected: NA
proj4string : [NA]
Number of points: 65
```

Getting back to a `ppp` object is also done by coercing, but if we want to preserve the actual dimensions, we have to manipulate the `owin` object belonging to the `ppp` object directly. We return later to see how `SpatialPolygons` objects may be coerced into `owin` objects, and how **spatstat** `im` objects can interface with `SpatialGrid` objects.

```
> pppjap <- as(spjpines1, "ppp")
> summary(pppjap)
```

```
Planar point pattern: 65 points
Average intensity 65 points per square unit

Window: rectangle = [0, 1] x [0, 1] units
Window area =  1 square unit
```

These point patterns have been obtained by sampling in different regions, but it is not rare to find examples in which we have different types of events in the same region. In spatial epidemiology, for example, it is common to have two types of points: *cases* of a certain disease and *controls*, which usually reflect the spatial distribution of the population. In general, this kind of point pattern is called a *marked* point pattern because each point is assigned to a group and labelled accordingly.

The Asthma data set records the results of a case–control study carried out in 1992 on the incidence of asthma in children in North Derbyshire (United Kingdom). This data set has been studied by Diggle and Rowlingson (1994), Singleton et al. (1995), and Diggle (2003) to explore the relationship between asthma and the proximity to the main roads and three putative pollution sources (a coking works, chemical plant, and waste treatment centre). In the study, a number of relevant covariates were also collected by means of a questionnaire that was completed by the parents of the children attending 10 schools in the region. Children having suffered from asthma will act as cases whilst the remainder of the children included in the study will form the set of controls. Although this data set is introduced here, the spatial analysis of case–control data is described in the final part of this chapter.

The data set is available from Prof. Peter J. Diggle's website and comes in anonymised form. Barry Rowlingson provided some of the road lines. The original data were supplied by Dr. Joanna Briggs (University of Leeds, UK). To avoid computational problems in some of the methods described in this section, we have removed a very isolated point, which was one of the cases, and we have selected an appropriate boundary region.

The next example shows how to display the point pattern, including the boundary of the region (that we have created ourselves) and the location of the pollution sources using different **sp** layouts and function `spplot` (see Chap. 3 for more details). Given that the data set is a marked point pattern, we have converted it to a `SpatialPointsDataFrame` to preserve the type (case or control) of the events and all the other relevant information. In addition, we have created a `SpatialPolygons` object to store the boundary of the region and a `SpatialPointsDataFrame` object for the location of the three pollution sources. Given that the main roads are available, we have included them as well using a `SpatialLines` object. The final plot is shown in Fig. 7.2.

```
> library(rgdal)
> spasthma <- readOGR(".", "spasthma")
> spbdry <- readOGR(".", "spbdry")
> spsrc <- readOGR(".", "spsrc")
> sproads <- readOGR(".", "sproads")
```

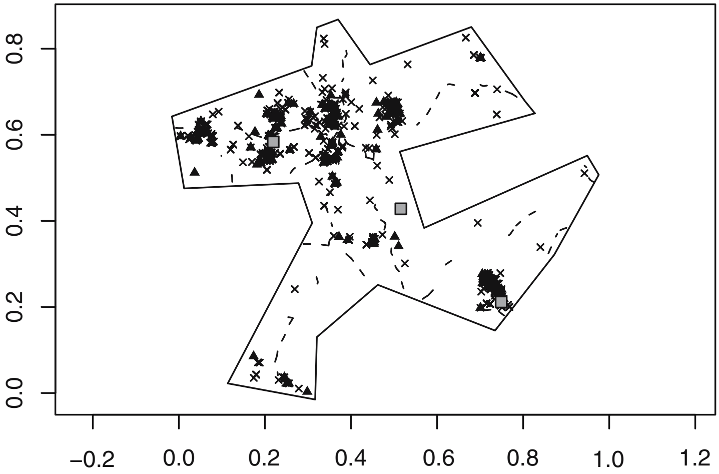

Fig. 7.2. Locations of the residence of asthmatic (cases, *filled triangle*) and non-asthmatic (controls, *cross*) in North Derbyshire, 1992 (Diggle and Rowlingson, 1994). The boundary has been taken to contain all points in the data set. The map shows the pollution sources (*grey filled square*) and the main roads (*dashed lines*)

7.3 Preliminary Analysis of a Point Pattern

The analysis of point patterns is focused on the spatial distribution of the observed events and making inference about the underlying process that generated them. In particular, there are two main issues of interest: the distribution of events in space and the existence of possible interactions between them. For a merely descriptive analysis, we would represent the locations of the point pattern in the study area. This will give us an idea of the distribution of the points, and it can lead to possible hypothesis about the spatial distribution of the events. Further statistical analyses can be done and they are described in this section.

7.3.1 Complete Spatial Randomness

When studying a point process, the most basic test that can be performed is that of *Complete Spatial Randomness* (CSR, henceforth). Intuitively, by CSR we mean that the events are distributed independently at random and uniformly over the study area. This implies that there are no regions where the events are more (or less) likely to occur and that the presence of a given event does not modify the probability of other events appearing nearby.

Informally, this can be tested by plotting the point pattern and observing whether the points tend to appear in clusters or, on the contrary, they follow a regular pattern. In any of these cases, the points are not distributed uniformly because they should be distributed filling all the space in the study area. Usually, clustered patterns occur when there is attraction (i.e. 'contagion') between points, whilst regular patterns occur when there is inhibition (i.e. 'competition') among points.

Figure 7.1 shows three examples of point patterns that have been generated by different biological mechanisms and seem to have different spatial distributions. In particular, plot 7.1 of the Japanese pine trees (middle) seems neither clustered nor regularly distributed, whilst the redwood seeds (right) show a clustered pattern and the cells (left) a regular one. Hence, only the spatial distribution of Japanese pine trees seems to be compatible with CSR.

To measure the degree of accomplishment of the CSR, several functions can be computed on the data. These are described in the following sections, together with methods to measure the uncertainty related to the observed pattern.

Testing for CSR is covered in Waller and Gotway (2004, pp. 118–126), O'Sullivan and Unwin (2003, pp. 88–92, including a discussion on pp. 108–112), and Schabenberger and Gotway (2005, pp. 86–99, including other methods not presented here).

7.3.2 *G* Function: Distance to the Nearest Event

The G function measures the distribution of the distances from an arbitrary event to its nearest event. If these distances are defined as $d_i = \min_j\{d_{ij}, \forall j \neq i\}$, $i = 1, \dots, n$, then the G function can be estimated as

$$\hat{G}(r) = \frac{\#\{d_i : d_i \leq r, \forall i\}}{n},$$

where the numerator is the number of elements in the set of distances that are lower than or equal to d and n is the total number of points. Under CSR, the value of the G function is

$$G(r) = 1 - \exp\{-\lambda \pi r^2\},$$

where λ represents the mean number of events per unit area (or *intensity*).

The compatibility with CSR of the point pattern can be assessed by plotting the empirical function $\hat{G}(d)$ against the theoretical expectation. In addition, point-wise envelopes under CSR can be computed by repeatedly simulating a CSR point process with the same estimated intensity $\hat{\lambda}$ in the study region (Diggle, 2003, p. 13) and check whether the empirical function is contained inside. The next chunk of code shows how to compute this by using **spatstat** functions `Gest` and `envelope`. The results have been merged in a data frame in order to use conditional Lattice graphics.

```
> r <- seq(0, sqrt(2)/6, by = 0.005)
> envjap <- envelope(as(spjpines1, "ppp"), fun = Gest,
+     r = r, nrank = 2, nsim = 99)
> envred <- envelope(as(spred, "ppp"), fun = Gest, r = r,
+     nrank = 2, nsim = 99)
> envcells <- envelope(as(spcells, "ppp"), fun = Gest,
+     r = r, nrank = 2, nsim = 99)
> Gresults <- rbind(envjap, envred, envcells)
> Gresults <- cbind(Gresults, DATASET = rep(c("JAPANESE",
+     "REDWOOD", "CELLS"), each = length(r)))
```

Figure 7.3 shows the empirical function $\hat{G}(r)$ against $G(r)$ together with the 96% pointwise envelopes (because `nrank=2`) of the same point pattern examined using the G function. The plot is produced by taking the pairs $(G(r), \hat{G}(r))$ for a set of reasonable values of the distance r, so that in the x-axis we have the values of the theoretical value of $G(r)$ under CSR and in the y-axis the empirical function $\hat{G}(r)$. The results show that only the Japanese trees seem to be homogeneously distributed, whilst the redwood seeds show a clustered pattern (values of $\hat{G}(r)$ above the envelopes) and the location of the cells shows a more regular pattern (values of $\hat{G}(r)$ below the envelopes).

`envelope` is a very flexible function that can be used to compute Monte Carlo envelopes of a certain type of functions. Basically, it works by randomly simulating a number of point patterns so that the summary function is computed for all of them. The resulting values are then used to compute point-wise

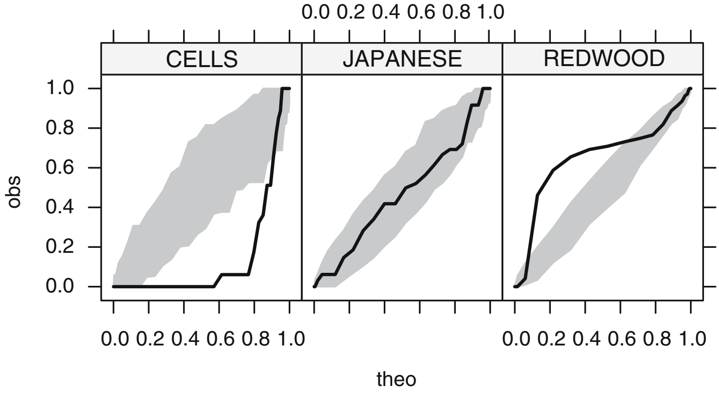

Fig. 7.3. Envelopes and observed values of the G function for three point patterns

(i.e. at different distances) or global Monte Carlo envelopes. `envelope` can be passed the way the point patterns are generated (by default, CSR). The reader is referred to the manual page for more information.

7.3.3 *F* Function: Distance from a Point to the Nearest Event

The F function measures the distribution of all distances from an arbitrary point of the plane to its nearest event. This function is often called the *empty space* function because it is a measure of the average space left between events. Under CSR, the expected value of the F function is

$$F(r) = 1 - \exp\{-\lambda \pi r^2\}.$$

Hence, we can compare the estimated value of the F function to its theoretical value and compute simulation envelopes as before.

```
> Fenvjap <- envelope(as(spjpines1, "ppp"), fun = Fest,
+     r = r, nrank = 2, nsim = 99)
> Fenvred <- envelope(as(spred, "ppp"), fun = Fest, r = r,
+     nrank = 2, nsim = 99)
> Fenvcells <- envelope(as(spcells, "ppp"), fun = Fest,
+     r = r, nrank = 2, nsim = 99)
> Fresults <- rbind(Fenvjap, Fenvred, Fenvcells)
> Fresults <- cbind(Fresults, DATASET = rep(c("JAPANESE",
+     "REDWOOD", "CELLS"), each = length(r)))
```

Figure 7.4 shows the empirical F functions and their associated 96% envelopes (because `nrank=2`) for the three data sets presented before. The Japanese data are compatible with the CSR hypothesis, whereas the cells point pattern shows a regular pattern ($\hat{F}(r)$ is above the envelopes) and the redwood points seem to be clustered, given the low values of $\hat{F}(r)$.

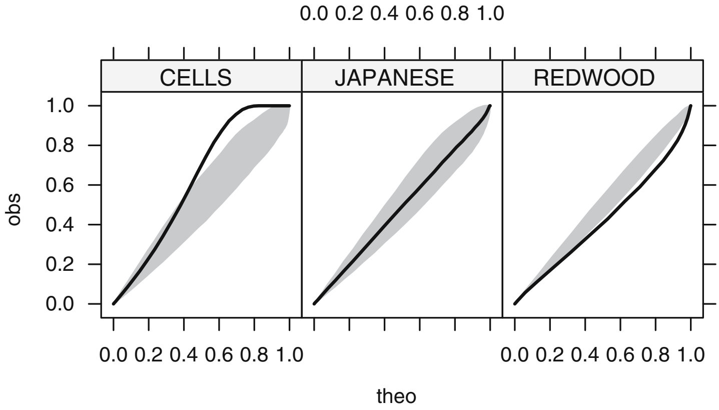

Fig. 7.4. Envelopes and observed values of the F function for three point patterns

7.4 Statistical Analysis of Spatial Point Processes

A first description of the point pattern can be done by estimating the spatial statistical density from the observed data. The spatial density has the same properties as a univariate density, but its domain is the study area where the point process takes place.

As an alternative function to measure the spatial distribution of the events, we can work with the *intensity* $\lambda(x)$ of the point process, which is proportional to its spatial density. The constant of proportionality is the expected number of events of the point process in the area A. That is, for two point processes with the same spatial density but different intensities, the number of events observed will be higher for the process with the highest intensity.

The intensity and spatial density are part of the *first-order* properties because they measure the distribution of events in the study region. Note that neither the intensity nor the spatial density give any information on the interaction between two arbitrary points. This is measured by *second-order properties*, which reflect any tendency of the events to appear clustered, independently, or regularly spaced.

First- and second-order properties are properly defined in, for example, Diggle (2003, p. 43) and Möller and Waagepetersen (2003, Chap. 4). We focus on the estimation of the intensity and the assessment of clustering, as explained in the following sections. Waller and Gotway (2004, pp. 130–146) and Schabenberger and Gotway (2005, 90–103, 110–112) discuss the estimation of the intensity of a point pattern and the assessment of clustering as well.

The separation between first- and second-order properties can be difficult to disentangle without further assumptions. For example, do groups of events appear at a specific location because the intensity is higher there or because events are clustered? In general, it is assumed that interaction between points occurs at small scale, while large-scale variation is reflected on the intensity

(Diggle, 2003, p. 143). Waller and Gotway (2004, 146–147) also discuss the roles of first and second-order properties.

In the remainder of this chapter, we focus on Poisson processes because they offer a simple approach to a wide range of problems. Loosely, we can distinguish between homogeneous and inhomogeneous Poisson point processes (HPP and IPP, respectively). Both HPP and IPP assume that the events occur independently and are distributed according to a given intensity. The main difference between the two point processes is that the HPP assumes that the intensity function is constant, while the intensity of an IPP varies spatially. In a sense, the IPP is a generalisation of the HPP or, inversely, the HPP can be regarded as an IPP with constant intensity. Poisson processes are also described in Schabenberger and Gotway (2005, pp. 81–86, 107–110) and Waller and Gotway (2004, pp. 126–130).

Note that other spatial processes may be required when more complex data sets are to be analysed. For example, when events are clustered, points do not occur independently of each other and a clustered process would be more appropriate. See Diggle (2003, Chap. 5) and Möller and Waagepetersen (2003) for a wider description of other spatial point processes. `spatstat` provides a number of functions to fit some of the models described therein.

7.4.1 Homogeneous Poisson Processes

A homogeneous Poisson process is characterised as representing the kind of point process in which all events are independently and uniformly distributed in the region A where the point process occurs. This means that the location of one point does not affect the probabilities of other points appearing nearby and that there are no regions where events are more likely to appear.

More formally, Diggle (2003) describes an HPP in a region A as fulfilling:

1. The number of events in A, with area $|A|$, is Poisson distributed with mean $\lambda|A|$, where λ is the constant intensity of the point process.
2. Given n observed events in region A, they are uniformly distributed in A.

The HPP is also *stationary* and *isotropic*. It is stationary because the intensity is constant and the second-order intensity depends only on the relative positions of two points (i.e. direction and distance). In addition, it is isotropic because the second-order intensity is invariant to rotation. Hence, the point process has constant intensity and its second-order intensity depends only on the distance between the two points, regardless of the relative positions of the points.

These constraints reflect that the intensity of the point process is constant, that is $\lambda(x) = \lambda > 0, \forall x \in A$, and that events appear independently of each other. Hence, the HPP is the formal definition of a point process which is CSR.

7.4.2 Inhomogeneous Poisson Processes

In most cases assuming that a point process under study is homogeneous is not realistic. Clear examples are the distribution of the population in a city or the location of trees in a forest. In both cases, different factors affect the spatial distribution. In the case of the population, it can be the type of housing, neighbourhood, etc., whilst in the case of the trees, it can be the environmental factors such as humidity, quality of the soil, slope and others.

The IPP is a generalisation of the HPP, which allows for a non-constant intensity. The same principle of independence between events holds, but now the spatial variation can be more diverse, with events appearing more likely in some areas than others. As a result, the intensity will be a generic function $\lambda(x)$ that varies spatially.

7.4.3 Estimation of the Intensity

As stated previously, the intensity of an HPP point process is constant. Hence, the problem of estimating the intensity is the problem of estimating a constant function λ such as the expected number of events in region A ($\int_A \lambda \, \mathrm{d}x$) is equal to the observed number of cases. This is the volume under the surface defined by the intensity in region A. Once we have observed the (homogeneous) point process, we have the locations of a set of n points. So, an unbiased estimator of the intensity is $n/|A|$, where $|A|$ is the area of region A. This ensures that the expected number of points is, in fact, the observed number of points.

For IPP, the estimation of the intensity can be done in different ways. It can be done non-parametrically by means of kernel smoothing or parametrically by proposing a specific function for the intensity whose parameters are estimated by maximising the likelihood of the point process. If we have observed n points $\{x_i\}_{i=1}^n$, the form of a kernel smoothing estimator is the following (Diggle, 1985; Berman and Diggle, 1989):

$$\hat{\lambda}(x) = \frac{1}{h^2} \sum_{i=1}^{n} \kappa\Big(\frac{||x - x_i||}{h}\Big)/q(||x||), \qquad (7.1)$$

where $\kappa(u)$ is a bivariate and symmetrical kernel function. $q(||x||)$ is a border correction to compensate for the missing observations that occur when x is close to the border of the region A. Bandwidth h measures the level of smoothing. Small values will produce very peaky estimates, whilst large values will produce very smooth functions.

Silverman (1986) gives a detailed description of different kernel functions and their properties. In the examples included in this chapter, we have used the *quartic* kernel (also known as *biweight*), whose expression in two dimensions is

$$\kappa(u) = \begin{cases} \frac{3}{\pi}(1 - \|u\|^2)^2 & \text{if } u \in (-1, 1) \\ 0 & \text{Otherwise} \end{cases},$$

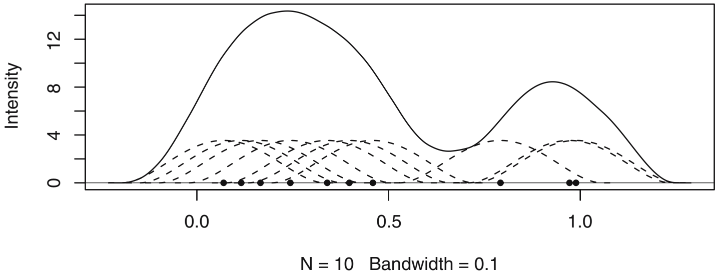

Fig. 7.5. Example of the contribution of the different points to the estimate of the intensity. Dashed lines represent the kernel around each observation, whilst the solid line is the estimate of the intensity

where $\|u\|^2$ denotes the squared norm of point $u = (u_1, u_2)$ equal to $u_1^2 + u_2^2$. Figure 7.5 shows an example of estimation of the intensity by kernel smoothing in a one-dimensional setting, but the same ideas are used in a spatial framework.

Methods for the selection of the bandwidth of kernel smoothers in a general setting are given by Silverman (1986). In the context of spatial analysis, a few proposals have been made so far, but it is not clear how to choose an optimal value for the bandwidth in the general case. It seems reasonable to use several values depending on the process under consideration, and choose a value that seems plausible.

Diggle (1985) and Berman and Diggle (1989) propose a criterion based on minimising the Mean Square Error (MSE) of the kernel smoothing estimator when the underlying point process in a stationary Cox process (see, e.g. p. 68 of Diggle (2003) for details). However, it can still be used as a general exploratory method and a guidance in order to choose the bandwidth. Kelsall and Diggle (1995a,b, 1998) propose and compare different methods for the selection of the bandwidth when a case–control point pattern is used. Clark and Lawson (2004) have compared these and other methods for disease mapping, including some methods for the automatic selection of the bandwidth.

We have applied the approach proposed by Berman and Diggle (1989), which is implemented in function `mse2d` to the redwood data set.

```
> library(splancs)
> mserw <- mse2d(as.points(coordinates(spred)), as.points(list(x = c(0,
+     1, 1, 0), y = c(0, 0, 1, 1))), 100, 0.15)
> bw <- mserw$h[which.min(mserw$mse)]
```

Figure 7.6 shows different values of the bandwidth and their associated values of the MSE. The value that minimises it is 0.039, but it should be noted that the curve is very flat around that point, which means that many other values of the bandwidth are plausible. This is a common problem in the analysis of real data sets.

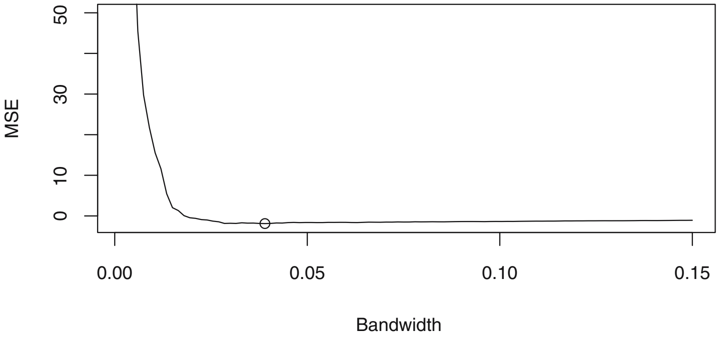

Fig. 7.6. Values of the mean square error for several values of the bandwidth using the redwood data set. The value that minimises it is 0.039 but many other values seem plausible, given the flatness of the curve

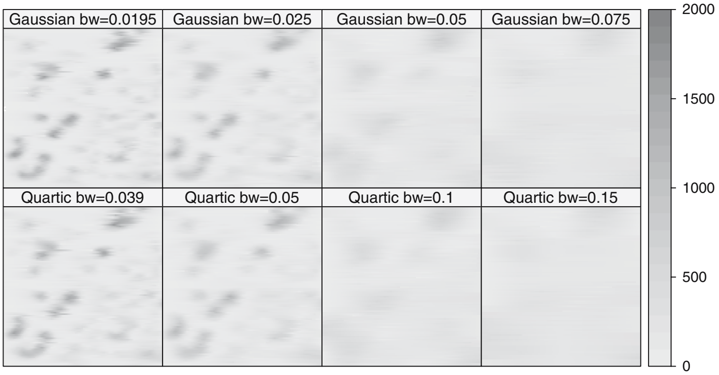

Fig. 7.7. Different estimates of the intensity of the redwood data set using a quartic kernel and different values of the bandwidth

It must be noted that when estimating the intensity by kernel smoothing, the key choice is not that of the kernel function but the bandwidth. Different kernels will produce very similar estimates for equivalent bandwidths, but the same kernel with different bandwidths will produce dramatically different results. An example of this fact is shown in Fig. 7.7, where four different bandwidths have been used to estimate the intensity of the redwood data.

```
> poly <- as.points(list(x = c(0, 0, 1, 1), y = c(0, 1,
+     1, 0)))
> sG <- Sobj_SpatialGrid(spred, maxDim = 100)$SG
> grd <- slot(sG, "grid")
> summary(grd)
> k0 <- spkernel2d(spred, poly, h0 = bw, grd)
> k1 <- spkernel2d(spred, poly, h0 = 0.05, grd)
> k2 <- spkernel2d(spred, poly, h0 = 0.1, grd)
> k3 <- spkernel2d(spred, poly, h0 = 0.15, grd)
> df <- data.frame(k0 = k0, k1 = k1, k2 = k2, k3 = k3)
> kernels <- SpatialGridDataFrame(grd, data = df)
> summary(kernels)
```

Package **spatstat** provides similar functions to estimate the intensity by kernel smoothing using an isotropic Gaussian kernel. We have empirically adjusted the value of the bandwidth to make the kernel estimates comparable. See Härdle et al. (2004, Sect. 3.4.2) for a full discussion. When calling `density` on a `ppp` object (which in fact calls `density.ppp`), we have used the additional arguments `dimxy` and `xy` to make sure that the grid used to compute the estimates is compatible with that stored in `kernels`. Finally, the kernel estimate is returned in an `im` class that is converted into a `SpatialGridDataFrame` and the values incorporated into `kernels`.

```
> xy <- list(x = coordinates(kernels)[, 1], y = coordinates(kernels)[,
+     2])
> k4 <- density(as(spred, "ppp"), 0.5 * bw, dimyx = c(100,
+     100), xy = xy)
> kernels$k4 <- as(k4, "SpatialGridDataFrame")$v
> k5 <- density(as(spred, "ppp"), 0.5 * 0.05, dimyx = c(100,
+     100), xy = xy)
> kernels$k5 <- as(k5, "SpatialGridDataFrame")$v
> k6 <- density(as(spred, "ppp"), 0.5 * 0.1, dimyx = c(100,
+     100), xy = xy)
> kernels$k6 <- as(k6, "SpatialGridDataFrame")$v
> k7 <- density(as(spred, "ppp"), 0.5 * 0.15, dimyx = c(100,
+     100), xy = xy)
> kernels$k7 <- as(k7, "SpatialGridDataFrame")$v
> summary(kernels)
```

7.4.4 Likelihood of an Inhomogeneous Poisson Process

The previous procedure to estimate the intensity is essentially non-parametric. Alternatively, a specific parametric or semi-parametric form for the intensity may be of interest (e.g. to include available covariates). Standard statistical techniques, such as the maximisation of the likelihood, can be used to estimate the parameters that appear in the expression of the intensity.

The expression of the likelihood can be difficult to work out for many point processes. However, in the case of the IPP (and, hence, the HPP) it has a very simple expression. The log-likelihood of a realisation of n independent events of an IPP with intensity $\lambda(x)$ is (Diggle, 2003, p. 104)

$$L(\lambda) = \sum_{i=1}^{n} \log \lambda(x_i) - \int_A \lambda(x)\, \mathrm{d}x,$$

where $\int_A \lambda(x)\, \mathrm{d}x$ is the expected number of cases of the IPP with intensity $\lambda(x)$ in region A.

When the intensity of the point process is estimated parametrically, the likelihood can be maximised to obtain the estimates of the parameters of the model. Diggle (2003, p. 104) suggests a log-linear model

$$\log \lambda(x) = \sum_{j=1}^{p} \beta_j z_j(x)$$

using covariates $z_j(x),\ j = 1, \ldots, p$ measured at a location x. These models can be fit using standard numerical integration techniques.

The following example defines the log-intensity (loglambda) at a given point $x = (x_1, x_2)$ using the parametric specification given by

$$\log \lambda(x) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 * x_2. \tag{7.2}$$

This expression is in turn used to construct the likelihood of an IPP (L). Function adapt (from the package with the same name) is used to compute numerically the integral that appears in the expression of the likelihood.

```
> loglambda <- function(x, alpha, beta) {
+     l <- alpha + sum(beta * c(x, x * x, prod(x)))
+     return(l)
+ }
> L <- function(alphabeta, x) {
+     l <- apply(x, 1, loglambda, alpha = alphabeta[1],
+         beta = alphabeta[-1])
+     l <- sum(l)
+     intL <- adapt(2, c(0, 0), c(1, 1), functn = function(x,
+         alpha, beta) {
+         exp(loglambda(x, alpha, beta))
+     }, alpha = alphabeta[1], beta = alphabeta[-1])
+     l <- l - intL$value
+     return(l)
+ }
```

The following example uses the locations of maple trees from the Lansing Woods data set (Gerard, 1969) in order to show how to fit a parametric intensity using (7.2). The parameters are estimated by maximising the likelihood using function optim.

```
> library(adapt)
> data(lansing)
> x <- as.points(lansing[lansing$marks == "maple", ])

> optbeta <- optim(par = c(log(514), 0, 0, 0, 0, 0), fn = L,
+     control = list(maxit = 1000, fnscale = -1), x = x)
```

The values of the coefficients $\alpha, \beta_1, \ldots, \beta_5$ are 5.53, 5.64, –0.774, –5.01, –1.2, 0.645, for a value of the (maximised) likelihood of 2778.4. Figure 7.8 shows the location of the maple trees and the estimated intensity according

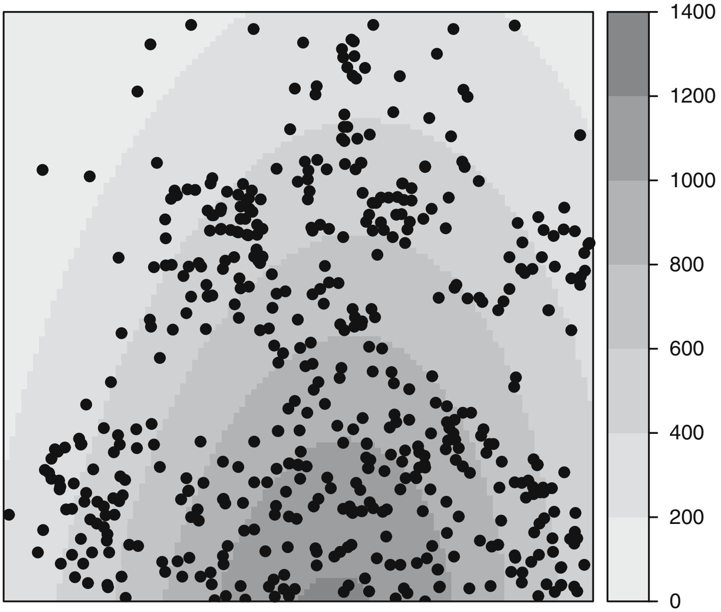

Fig. 7.8. Location of maple trees from the Lansing data set and their estimated parametric intensity using model (7.2)

to parametric model in (7.2). See Diggle (2003, Chap. 7) for a similar analysis using all the tree species in the Lansing Woods data set.

The same example can be run using function ppm from **spatstat** as follows (x and y representing the coordinates of the point pattern):

```
> lmaple <- lansing[lansing$marks == "maple", ]
> ppm(Q = lmaple, trend = ~x + y + I(x^2) + I(y^2) + I(x *
+     y))

Nonstationary multitype Poisson process
Possible marks:
blackoak hickory maple misc redoak whiteoak

Trend formula: ~x + y + I(x^2) + I(y^2) + I(x * y)

Fitted coefficients for trend formula:
(Intercept)           x           y      I(x^2)      I(y^2)
  3.7310742   5.6400643  -0.7663636  -5.0115142  -1.1983209
   I(x * y)
  0.6375824
```

As the authors mention in the manual page, ppm can be compared to glm because it can be used to fit a specific type of point process model to a particular point pattern. In this case, the family argument used in glm to define the model is substituted by `interaction`, which defines the point process to be fit. By default, a Poisson point process is used, but many other point processes can be fitted (see manual page for details).

7.4.5 Second-Order Properties

Second-order properties measure the strength and type of the interactions between events of the point process. Hence, they are particularly interesting if we are keen on studying clustering or competition between events.

Informally, the *second-order intensity* of two points x and y reflects the probability of any pair of events occurring in the vicinities of x and y, respectively. Diggle (2003, p. 43) and Möller and Waagepetersen (2003, Chap. 4) give a more formal description of the second-order intensity. Schabenberger and Gotway (2005, pp. 99–103) and Waller and Gotway (2004, pp. 137–147) also discuss second-order properties and the role of the K-function.

An alternative way of measuring second-order properties when the spatial process is HPP is by means of the K-function (Ripley, 1976, 1977). The K-function measures the number of events found up to a given distance of any particular event and it is defined as

$$K(s) = \lambda^{-1} E[N_0(s)],$$

where $E[.]$ denotes the expectation and $N_0(s)$ represents the number of further events up to a distance s around an arbitrary event. To compute this function, Ripley (1976) also proposed an unbiased estimate equal to

$$\hat{K}(s) = (n(n-1))^{-1}|A| \sum_{i=1}^{n} \sum_{j \neq i} w_{ij}^{-1} |\{x_j : d(x_i, x_j) \leq s\}|, \qquad (7.3)$$

where w_{ij} are weights equal to the proportion of the area inside the region A of the circle centred at x_i and radius $d(x_i, x_j)$, the distance between x_i and x_j.

The value of the K-function for an HPP is $K(s) = \pi s^2$. By comparing the estimated value $\hat{K}(s)$ to the theoretical value we can assess what kind of interaction exists. Usually, we assume that these interactions occur at small scales, and so will be interested in relatively small values of s. Values of $\hat{K}(s)$ higher than πs^2 are characteristic of clustered processes, whilst values smaller than that are found when there exists competition between events (regular pattern).

```
> Kenvjap <- envelope(as(spjpines1, "ppp"), fun = Kest,
+     r = r, nrank = 2, nsim = 99)
> Kenvred <- envelope(as(spred, "ppp"), fun = Kest, r = r,
+     nrank = 2, nsim = 99)
> Kenvcells <- envelope(as(spcells, "ppp"), fun = Kest,
+     r = r, nrank = 2, nsim = 99)
> Kresults <- rbind(Kenvjap, Kenvred, Kenvcells)
```

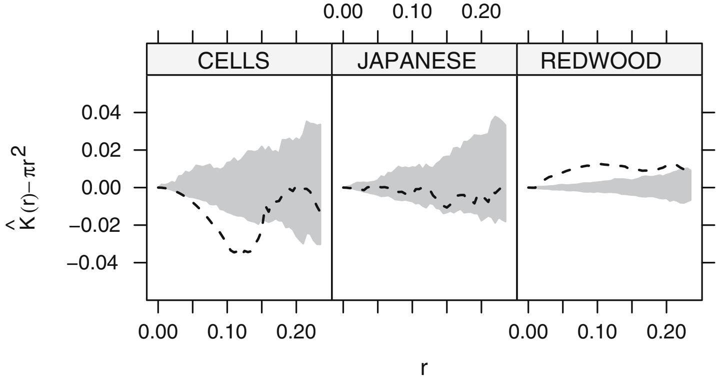

Fig. 7.9. Envelopes and observed values of Ripley's K-function for three point patterns

```
> Kresults <- cbind(Kresults, DATASET = rep(c("JAPANESE",
+     "REDWOOD", "CELLS"), each = length(r)))
```

Figure 7.9 shows the estimated K-function minus the theoretical value under CSR of the three point patterns that we have considered before. Note that the biological interpretations must be made cautiously because the underlying mechanisms are quite different and the scale of the interactions (if any) will probably be different for each point pattern. This is reflected in two ways: the width of the envelopes, which reflects the variability of the process under the null hypothesis of CSR, and the scale of the interaction. This seems to exist only for the cells, which follow a regular pattern, and the redwood seeds, which seem to be clustered. The Japanese trees point pattern is compatible with CSR because the estimated K-function is contained within the envelopes.

Inhomogeneous *K*-Function

Baddeley et al. (2000) propose a version of the K-function for non-homogeneous point processes, in particular, for the class of point processes which are second-order reweighted-stationary, which includes IPPs. This means that the second-order intensity of two points, divided by their respective intensities, is stationary. The inhomogeneous K-function is used in Sect. 7.5.5 in the analysis of case–control point patterns.

7.5 Some Applications in Spatial Epidemiology

In this section we focus on different applications of the analysis of point patterns in Spatial Epidemiology. Gatrell et al. (1996) and Diggle (2003) describe most of the methods contained here, but a comprehensive description of spatial methods for the analysis of epidemiological data can be found in

Elliott et al. (2000) and Waller and Gotway (2004). Furthermore, Chap. 11 describes the analysis of epidemiological data when they are aggregated.

The distribution of the cases of a certain disease can be regarded as the realisation of a point process, which reflects the underlying distribution of the population (which usually is not homogeneous) plus any other risk factors related to the disease and that are likely to depend on the subjects. Hence, we need to have accurate records of the locations of the disease cases, which can also include additional information on the individuals such as age, gender, and others.

In a spatial setting, the primary interest is on the spatial distribution of the cases, but any underlying risk factor that affects this spatial distribution should be taken into account. It is clear that looking solely at the spatial distribution of the cases in order to detect areas of high incidence is useless because the distribution of the cases will reflect that of the population. To overcome this problem, it would be necessary to have an estimate of the spatial distribution of the population so that it can be compared to that of the cases. For this reason, a set of controls can be randomly selected from the population at risk so that its spatial variation can be estimated (see, e.g. Prince et al., 2001).

Different authors have approached this problem in different ways. Diggle and Chetwynd (1991), for example compute the difference of the homogeneous K-function of cases and controls. Kelsall and Diggle (1995a,b) use non-parametric estimates of the distribution of the ratio between the intensities of cases and controls (i.e. the *relative* risk). Kelsall and Diggle (1998) propose a similar model and the use of binary regression and additive models to account for covariates and a smoothing term to model the residual spatial variation. More recently, Diggle et al. (2007) use the inhomogeneous K-function to compare the spatial distribution of cases and controls after accounting for the effect of relevant covariates.

Many of these methods are also covered, including new examples, and discussed in Schabenberger and Gotway (2005, pp. 103–122), Waller and Gotway (2004, Chap. 6) and O'Sullivan and Unwin (2003, see the discussion in Chap. 5).

7.5.1 Case–Control Studies

As we need to estimate the spatial distribution of the population, a number of individuals can be taken at random to make a set of controls. Controls are often selected using the population register or, if it is not available, the events of another non-related disease (Diggle, 1990). Furthermore, some strategies, such as stratification and matching (Jarner et al., 2002), can be done in order to account for other sources of confounding, such as age and sex. As discussed by Diggle (2000) when matching is used in the selection of the controls, the hypothesis of random selection from the population is violated and specific methods to handle this are required (Diggle et al., 2000; Jarner et al., 2002).

In general, we have a set of n_1 cases and n_0 controls. Conditioning on the number of cases and controls, we can assume that they are realisations of two IPP with intensities $\lambda_1(x)$ and $\lambda_0(x)$, respectively. In this setting, assuming that the distribution of cases and controls is the same means that the intensities $\lambda_1(x)$ and $\lambda_0(x)$ are equal up to a proportionality constant, which is equal to the ratio between n_1 and n_0: $\lambda_1(x) = \frac{n_1}{n_0}\lambda_0(x)$. Note that the ratio between cases and controls is determined only by the study design.

Spatial Variation of the Relative Risk

Kelsall and Diggle (1995a,b) consider the estimator of the disease risk given by the ratio between the intensity of the cases and controls $\rho(x) = \lambda_1(x)/\lambda_0(x)$ in order to assess the variation of the risk. Under the null hypothesis of equal spatial distribution, the ratio is a constant $\rho_0 = n_1/n_0$.

Alternatively, a risk estimate $r(x)$ can be estimated by working with the logarithm of the ratio of the densities of cases and controls:

$$r(x) = \log(f(x)/g(x)), \tag{7.4}$$

$f(x) = \lambda_1(x)/\int_A \lambda_1(x)\,\mathrm{d}x$ and $g(x) = \lambda_0(x)/\int_A \lambda_0(x)\,\mathrm{d}x$, respectively. In this case, the null hypothesis of equal spatial distributions becomes $r(x) = 0$. The advantage of this approach is that 0 is the reference value for equal spatial distribution without regarding the number of cases and controls. Unfortunately, this presents several computational problems because the intensity of the controls may be zero at some points, as addressed by, for example, Waller and Gotway (2004, pp. 165–166).

Kelsall and Diggle (1995a) propose the use of a kernel smoothing to estimate each intensity and evaluate different alternatives to estimate the optimum bandwidth for each kernel smoothing. They conclude that the best option is to select the bandwidth by cross-validation and use the same bandwidth in both cases.

They choose the bandwidth that minimises the following criterion:

$$\begin{aligned} CV(h) = &-\int_A \hat{r}_h(x)^2\,\mathrm{d}x - 2n_1^{-1}\sum_{i=1}^{n_1}\hat{r}_h^{-i}(x_i)/\hat{f}_h^{-i}(x_i) \\ &+ 2n_0^{-1}\sum_{i=n_1+1}^{n_1+n_0}\hat{r}_h^{-i}(x_i)/\hat{g}_h^{-i}(x_i), \end{aligned}$$

where the superscript $-i$ means that the function is computed by removing the ith point.

This criterion is not currently implemented, but it can be done easily by using function `lambdahat` in package **spatialkernel**, which allows for the computing of the intensity at a set of point using a different set of points. Our implementation of the method (which does not use border correction to avoid computational instability) gave an optimal bandwidth of 0.275. However, as discussed in Diggle et al. (2007), these automatic methods should be used as a guidance when estimating the bandwidth. In this case, the value of 0.275 seems a bit high given the scale of the data and we have set it to 0.125.

```
> bwasthma <- 0.125
```

To avoid computational problems, we use the risk estimator $\rho(x)$. First of all, we need to create a grid over the study region where the risk ratio will be estimated, using the helper function `Sobj_SpatialGrid`.

```
> library(maptools)
> sG <- Sobj_SpatialGrid(spbdry, maxDim = 50)$SG
> gt <- slot(sG, "grid")
```

The risk ratio can be computed easily by estimating the intensity of cases and controls first, and then taking the ratio (as shown below) after using the `spkernel2d` function from **splancs**.

```
> pbdry <- slot(slot(slot(spbdry, "polygons")[[1]], "Polygons")[[1]],
+     "coords")
```

After unpacking the boundary coordinates of the study area, the point locations are divided between cases and controls and the intensities of each subset calculated for grid cells lying within the study area, using the chosen bandwidth. The **splancs** package uses a simple form of single polygon boundary, while **spatstat** can use multiple separate polygons (`SpatialPolygons` objects can be coerced to suitable `owin` objects).

```
> library(splancs)
> cases <- spasthma[spasthma$Asthma == "case", ]
> ncases <- nrow(cases)
> controls <- spasthma[spasthma$Asthma == "control", ]
> ncontrols <- nrow(controls)
> kcases <- spkernel2d(cases, pbdry, h0 = bwasthma, gt)
> kcontrols <- spkernel2d(controls, pbdry, h0 = bwasthma,
+     gt)
```

The results contain missing values for grid cells outside the study area, and so we first construct a `SpatialGridDataFrame` object to hold them and coerce to a `SpatialPixelsDataFrame` to drop the missing cells. The ratio is calculated, setting non-finite values from division by zero to missing.

```
> df0 <- data.frame(kcases = kcases, kcontrols = kcontrols)
> spkratio0 <- SpatialGridDataFrame(gt, data = df0)
> spkratio <- as(spkratio0, "SpatialPixelsDataFrame")
> spkratio$kratio <- spkratio$kcases/spkratio$kcontrols
> is.na(spkratio$kratio) <- !is.finite(spkratio$kratio)
> spkratio$logratio <- log(spkratio$kratio) - log(ncases/ncontrols)
```

To assess departure from the null hypothesis, they propose the following test statistic:

$$T = \int_A (\rho(x) - \rho_0)^2 \, \mathrm{d}x.$$

This integral can be estimated up to a proportionality constant by computing $\rho(x)$ on a regular grid of points $\{s_i, i = 1, \ldots, p\}$ and computing the sum of the values $\{(\rho(s_i) - \rho_0)^2, i = 1, \ldots, p\}$. Hence, an estimate of T is given by

$$\hat{T} = |c| \sum_{i=1}^{p} (\hat{\rho}(s_i) - \hat{\rho}_0)^2,$$

where $|c|$ is the area of the cells of the grid, $\hat{\rho}_0$ is n_1/n_0, and $\hat{\rho}(x)$ the estimate of the risk ratio.

Note that the former test is to assess whether there is constant risk all over the study region. However, risk is likely to vary spatially and another appropriate test can be done by substituting ρ_0 for $\hat{\rho}(x)$ (Kelsall and Diggle, 1995a). Now we are testing for significance of risk given that we assume that its variation is not homogeneous (i.e. equal to $\hat{\rho}(x)$) and the test statistic is

$$T = \int_A (\rho(x) - \hat{\rho}(x))^2 \, \mathrm{d}x.$$

Significance of the observed value of the test statistic can be computed by means of a Monte Carlo test (Kelsall and Diggle, 1995b). In this test, we compute k values of the test statistic T by re-labelling cases and controls (keeping n_1 and n_0 fixed) and calculating a new risk ratio $\hat{\rho}_i(x)$ $i = 1, \ldots, n$ for each new set of cases and controls. This will provide a series of values $T^1, \ldots, T^k$ under the null hypothesis. If we call T^0 the value of T for the observed data set, the significance (p-value) can be computed by taking $(t+1)/(k+1)$, where t is the number of values of $T^i, i = 1, \ldots, n$ greater than T^0.

The Monte Carlo test is based on the fact that cases and controls are equally distributed under the null hypothesis. In that case, if we change the label of a case to be a control (or viceversa), the new set of cases (or controls) still have the same spatial distribution and will have the same risk function $\rho(x)$. If that is not the case, then the re-labelling of cases and controls will produce different risk functions.

```
> idxinbdry <- overlay(sG, spbdry)
> idxna <- !is.na(idxinbdry)
```

We use the `overlay` method to find grid cells within the study area boundary, and use the number of included grid cells to set up objects to hold the results for the re-labelled cases and controls:

```
> niter <- 99
> ratio <- rep(NA, niter)
> pvaluemap <- rep(0, sum(idxna))
> rlabelratio <- matrix(NA, nrow = niter, ncol = sum(idxna))
```

The probability map is calculated by repeating the re-labelling process `niter` times, and tallying the number of times that the observed kernel density ratio is less than the re-labelled ratios. In the loop, the first commands carry out the re-labelling from the full set of points, and the remainder calculate the ratio and store the results:

```
> for (i in 1:niter) {
+     idxrel <- sample(spasthma$Asthma) == "case"
+     casesrel <- spasthma[idxrel, ]
+     controlsrel <- spasthma[!idxrel, ]
+     kcasesrel <- spkernel2d(casesrel, pbdry, h0 = bwasthma,
+         gt)
+     kcontrolsrel <- spkernel2d(controlsrel, pbdry, h0 = bwasthma,
+         gt)
+     kratiorel <- kcasesrel[idxna]/kcontrolsrel[idxna]
+     is.na(kratiorel) <- !is.finite(kratiorel)
+     rlabelratio[i, ] <- kratiorel
+     pvaluemap <- pvaluemap + (spkratio$kratio < kratiorel)
+ }
```

Figure 7.10 shows the kernel ratio of cases and controls, using a bandwidth of 0.125, as discussed before. We may have computational problems when estimating the intensity at points very close to the boundary of the study area and obtain `NA` instead of the value of the intensity. To avoid problems with this, we have filtered out these points using a new index called `idxna2`.

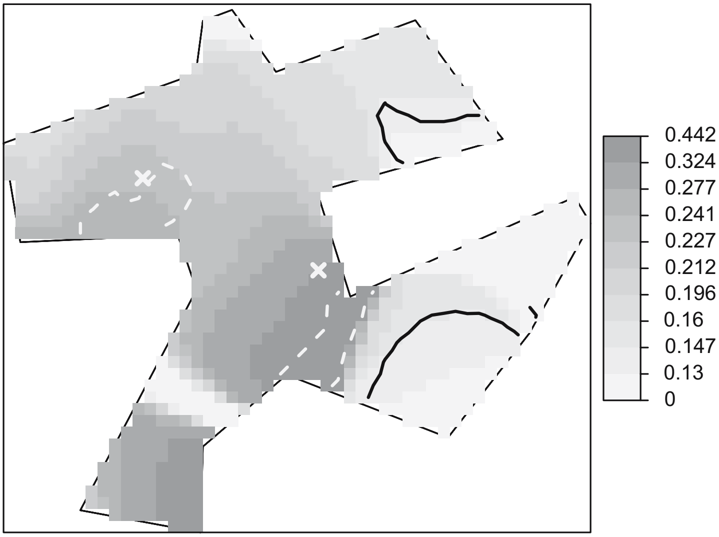

Fig. 7.10. Kernel ratio of the intensity of cases and controls. The continuous and dashed lines show the surfaces associated with 0.95 and 0.05 p-values, respectively, grey crosses mark the pollution sources. The value of $\hat{\rho}_0$ which marks a flat constant risk is 0.2

This will ensure that we use the same number of points when we estimate the value of the test statistic for the observed data and the permuted re-labelled sets of cases and controls.

```
> idxna2 <- apply(rlabelratio, 2, function(x) all(is.finite(x)))
> rhomean <- apply(rlabelratio[, idxna2], 2, mean)
> c <- prod(slot(gt, "cellsize"))
> ratiorho <- c * sum((spkratio$kratio[idxna2] - ncases/ncontrols)^2)
> ratio <- c * apply(rlabelratio[, idxna2], 1, function(X,
+     rho0) {
+     sum((X - rho0)^2)
+ }, rho0 = ncases/ncontrols)
> pvaluerho <- (sum(ratio > ratiorho) + 1)/(niter + 1)
```

The results for the test with null hypothesis $\rho = \hat{\rho}_0$ turned out to be non-significant (p-value of 0.69), which means that the observed risk ratio is consistent with a constant risk ratio. In principle, this agrees with the fact that Diggle and Rowlingson (1994) did not find a significant association with distance from main roads or two of the pollution sources and only a possible association with the remaining site, which should be further investigated. However, they found some relationship with other risk factors, but these were not of a spatial nature and, hence, this particular test is unable to detect it.

Had the p-value of the test been significant, 90% point confidence surfaces could be computed in a similar way to the envelopes shown before, but considering the different values of the estimates of $\rho(x)$ under random labelling and computing the p-value at each point. The procedure computes, for each point x_j in the grid, the proportion of values $\hat{\rho}_i(x_j)$ that are lower than $\hat{\rho}(x_j)$, where the $\hat{\rho}_i(x_j), i = 1, \ldots, R$ are the estimated ratios obtained by re-labelling cases and controls. Finally, the 0.05 and 0.95 contours of the p-value surface can be displayed on the plot of $\hat{\rho}(x)$ to highlight areas of significant low and high risk, respectively. This is shown in Fig. 7.10.

The contour lines at a given value can be obtained using function `contourLines`, which takes an `image` object. This will generate contour lines that can be converted to `SpatialLinesDataFrame` objects so that they can be added to a plot as a layout.

```
> spkratio$pvaluemap <- (pvaluemap + 1)/(niter + 1)
> imgpvalue <- as.image.SpatialGridDataFrame(spkratio["pvaluemap"])
> clpvalue <- contourLines(imgpvalue, levels = c(0, 0.05,
+     0.95, 1))
> cl <- ContourLines2SLDF(clpvalue)
```

7.5.2 Binary Regression Estimator

Kelsall and Diggle (1998) propose a binary regression estimator to estimate the probability of being a case at a given location, which can be easily extended to allow for the incorporation of covariates. In principle, the probabilities can be estimated by assuming that we have a variable Y_i, which labels cases ($y_i = 1$) and controls ($y_i = 0$) in a set of $n = n_1 + n_2$ events. Conditioning on the point locations, Y_i is a realisation of a Bernoulli variable Y_i with probability

$$P(Y_i = 1|X_i = x_i) = p(x_i) = \frac{\lambda_1(x_i)}{\lambda_0(x_i) + \lambda_1(x_i)}.$$

In practise, the following Nadaraya–Watson kernel estimator can be used:

$$\hat{p}_h(x) = \frac{\sum_{i=1}^n h^{-2}\kappa_h((x - x_i)/h)y_i}{\sum_{i=1}^n h^{-2}\kappa_h((x - x_i)/h)}, \tag{7.5}$$

where $\kappa_h(u)$ is a kernel function. Note that $p(x)$ is related to the log-ratio relative risk $r(x)$ as follows:

$$\text{logit}(p(x)) = \log\left(\frac{p(x)}{1 - p(x)}\right) = \log\left(\frac{\lambda_1(x)}{\lambda_0(x)}\right) = r(x) + \log(n_1/n_0).$$

$\hat{p}_h(x)$ can be estimated as

$$\hat{p}_h(x) = \frac{\hat{\lambda}_1(x)}{\hat{\lambda}_1(x) + \hat{\lambda}_0(x)}.$$

To estimate the bandwidth that appears in this new estimator, Kelsall and Diggle (1998) suggest another cross-validation criterion based on the value of h that minimises

$$CV(h) = \left[\prod_{i=1}^n \hat{p}_h^{-i}(x_i)^{y_i}(1 - \hat{p}_h^{-i}(x_i))^{1-y_i}\right]^{-1/n}.$$

Using the new criterion we obtained a bandwidth of 0.225, which is very similar to the one obtained with the previous cross-validation criterion. However, we believe that this value would over-smooth the data and we have set it to 0.125. The estimator for $p(x)$ can be computed easily, as is shown below. Figure 7.11 shows the resulting estimate.

```
> bwasthmap <- 0.125

> lambda1 <- spkernel2d(cases, pbdry, h0 = bwasthmap, gt)
> lambda0 <- spkernel2d(controls, pbdry, h0 = bwasthmap,
+     gt)
> lambda1 <- lambda1[idxna]
> lambda0 <- lambda0[idxna]
> spkratio$prob <- lambda1/(lambda1 + lambda0)
> is.na(spkratio$prob) <- !is.finite(spkratio$prob)
```

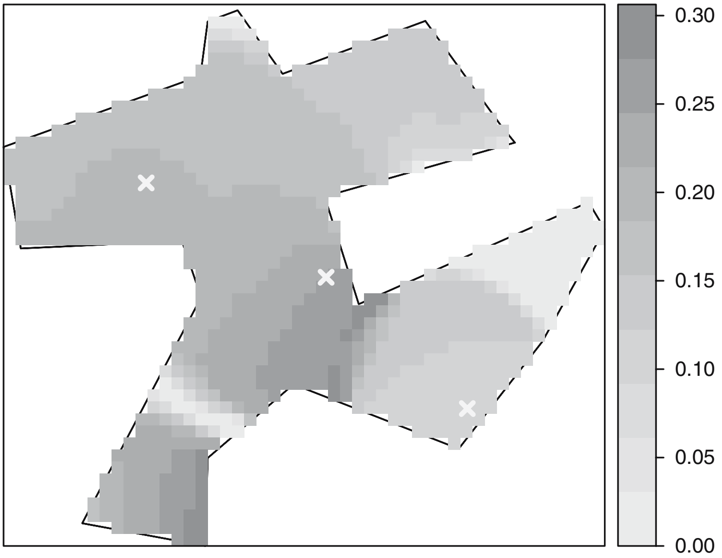

Fig. 7.11. Binary regression estimator using the probability of being a case at every grid cell in the study region

7.5.3 Binary Regression Using Generalised Additive Models

This formulation allows the inclusion of covariates in the model by means of standard logistic regression. In addition, the residual spatial variation can be modelled by including a smooth spatial function. In other words, if u is a vector of covariates observed at location x and $g(x)$ is a smooth function not dependent on the covariates, the formulation is

$$\text{logit}(p(x)) = u'\beta + g(x).$$

If the covariates are missing, the former expression is just another way of estimating the probability surface. Kelsall and Diggle (1998) estimate $g(x)$ using a kernel weighted regression. We have used package **mgcv** (Wood, 2006) to fit the Generalised Additive Model (GAM) models but, given that this package lacks the same non-parametric estimator used in Kelsall and Diggle (1998), we have preferred the use of a penalised spline instead.

The following example shows how to fit a GAM using the distance of the events to the pollution sources and main roads, and controlling for known and possible risk factors such as gender, age, previous events of hay fever, and having at least one smoker in the house. Rows have been filtered so that only children with a valid value of Gender (1 or 2) are used. We have included the distance as a proxy of the actual exposure to any risk factor caused by the pollution sources or the roads. Other models that consider a special modelling for the distance are considered later.

```
> spasthma$y <- as.integer(!as.integer(spasthma$Asthma) -
+     1)
> ccasthma <- coordinates(spasthma)
> spasthma$x1 <- ccasthma[, 1]
> spasthma$x2 <- ccasthma[, 2]
> spasthma$dist1 <- sqrt(spasthma$d2source1)
> spasthma$dist2 <- sqrt(spasthma$d2source2)
> spasthma$dist3 <- sqrt(spasthma$d2source3)
> spasthma$droads <- sqrt(spasthma$roaddist2)
> spasthma$smoking <- as.factor(as.numeric(spasthma$Nsmokers >
+     0))
> spasthma$Genderf <- as.factor(spasthma$Gender)
> spasthma$HayFeverf <- as.factor(spasthma$HayFever)
```

```
> library(mgcv)
> gasthma <- gam(y ~ 1 + dist1 + dist2 + dist3 + droads +
+     Genderf + Age + HayFeverf + smoking + s(x1, x2),
+     data = spasthma[spasthma$Gender == 1 | spasthma$Gender ==
+         2, ], family = binomial)
```

```
> summary(gasthma)
```

```
Family: binomial
Link function: logit

Formula:
y ~ 1 + dist1 + dist2 + dist3 + droads + Genderf + Age + HayFeverf +
    smoking + s(x1, x2)

Parametric coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.0326784  0.9195177  -2.211   0.0271 *
dist1        0.9822575  6.0714999   0.162   0.8715
dist2       -9.5790621  5.7708614  -1.660   0.0969 .
dist3       11.2247321  7.8724979   1.426   0.1539
droads       0.0001479  0.0001717   0.861   0.3890
Genderf2    -0.3476861  0.1562020  -2.226   0.0260 *
Age         -0.0679031  0.0382349  -1.776   0.0757 .
HayFeverf1   1.1881331  0.1875414   6.335 2.37e-10 ***
smoking1     0.1651210  0.1610362   1.025   0.3052
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
           edf Est.rank Chi.sq p-value
s(x1,x2) 2.001        2  7.004  0.0301 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0403   Deviance explained = 4.94%
UBRE score = -0.12348  Scale est. = 1         n = 1283
```

The results show that the significant variables are the presence of reported hay fever (p-value 2.4e-10) and gender (p-value 0.026). The coefficient of the second pollution source is marginally significant (p-value 0.097). The smoothed residual term using splines is significant (p-value 0.0301), which suggests that there may have been some residual spatial variation unexplained in the generalised linear model.

7.5.4 Point Source Pollution

In the previous model, we have shown how to consider the exposure to a number of pollution sources by including the distance as a covariate in the model. However, this approach does not allow for a more flexible parametric modelling of the exposure according to the distance to a pollution source. Diggle (1990) proposed the use of an IPP for the cases in which their intensity accounts for the distance to the pollution sources. In particular, the intensity is as follows:

$$\lambda_1(x) = \rho\lambda_0(x) f(x - x_0; \theta),$$

ρ measures the overall number of events per unit area, $\lambda_0(x)$ is the spatial variation of the underlying population (independent of the effect of the source), and $f(x - x_0; \theta)$ is a function of the distance from point x to the location of the source x_0 and has parameters θ. Diggle (1990) uses a decaying function with distance

$$f(x - x_0; \alpha, \beta) = 1 + \alpha \ \exp(-\beta||x - x_0||^2).$$

Parameters ρ, α, and β of $\lambda_1(x)$ can be estimated by maximising the likelihood of the IPP, assuming that $\lambda_0(x)$ is estimated by kernel smoothing taking a certain value h_0 of the bandwidth. That is, the value of h_0 is not obtained by the maximisation procedure, but choosing a reasonable value for h_0 can be difficult and it can have an important impact on the results.

A slightly different approach that does not require the choice of a bandwidth is considered in Diggle and Rowlingson (1994). It is based on the previous scenario, but conditioning on the location of cases and controls to model the probability of being a case at location x:

$$p(x) = \frac{\lambda_1(x)}{\lambda_1(x) + \lambda_0(x)} = \frac{\rho f(x - x_0; \alpha, \beta)}{1 + \rho f(x - x_0; \alpha, \beta)}.$$

As in the previous scenario, the remaining parameters of the model can be estimated by maximising the log-likelihood:

$$L(\rho, \theta) = \sum_{i=1}^{n_1} \log(p(x_i)) + \sum_{j=1}^{n_0} \log(1 - p(x_j)).$$

This model can be fitted using function tribble from package splancs. Given that $\lambda_0(x)$ vanishes we only need to pass the distances to the source and the labels of cases and controls.

To compare models that may include different sets of pollution sources or covariates, Diggle and Rowlingson (1994) compare the difference of the log-likelihoods by means of a chi-square test. The following example shows the results for the exposure model with distance to source two and another model with only the covariate hay fever.

```
> D2_mat <- as.matrix(spasthma$dist2)
> RHO <- ncases/ncontrols
> expsource2 <- tribble(ccflag = spasthma$y, vars = D2_mat,
+     rho = RHO, alphas = 1, betas = 1)

> print(expsource2)

Call:
tribble(ccflag = spasthma$y, vars = D2_mat, alphas = 1, betas = 1,
    rho = RHO)
Kcode = 2

Distance decay parameters:
        Alpha     Beta
[1,] 1.305824 25.14672

rho parameter : 0.163395847627903

     log-likelihood : -580.495955916672
null log-likelihood : -581.406203518987

        D = 2(L-Lo) : 1.82049520462942

> Hay_mat <- as.matrix(spasthma$HayFever)
> exphay <- tribble(ccflag = spasthma$y, rho = RHO, covars = Hay_mat,
+     thetas = 1)

> print(exphay)

Call:
tribble(ccflag = spasthma$y, rho = RHO, covars = Hay_mat, thetas = 1)
Kcode = 2

Covariate parameters:
[1] 1.103344

rho parameter : 0.163182953009353

     log-likelihood : -564.368250327802
null log-likelihood : -581.406203518987

        D = 2(L-Lo) : 34.0759063823702
```

As the output shows, the log-likelihood for the model with exposure to source 2 is −580.5, whilst for the model with the effect of hay fever is only −564.4. This means that there is a significant difference between the two models and that the model that accounts for the effect of hay fever is preferable. Even though the second source has a significant impact on the increase of the cases of asthma, its effect is not as important as the effect of having suffered from hay fever. However, another model could be proposed to account for both effects at the same time.

```
> expsource2hay <- tribble(ccflag = spasthma$y, vars = D2_mat,
+     rho = RHO, alphas = 1, betas = 1, covars = Hay_mat,
+     thetas = 1)
```

This new model (output not shown) has a log-likelihood of −563, with two more parameters than the model with hay fever. Hence, the presence of the second source has a small impact on the increase of cases of asthma after adjusting for the effect of hay fever, which can be regarded as the main factor related to asthma, and the model with hay fever only should be preferred. The reader is referred to Diggle and Rowlingson (1994) and Diggle (2003, p. 137) for more details on how the models can be compared and results for other models.

These types of models are extended by Diggle et al. (1997), who consider further options for the choice of the function $f(x - x_0, \alpha, \beta)$ to accommodate different spatial variants of the risk around the source.

In our experience, these models can be very sensitive to the initial values for certain data sets, especially if they are sparse. Hence, it is advised to fit the model using different values for the initial values to ensure that the algorithm is not trapped in a local maximum of the likelihood.

Assessment of General Spatial Clustering

As discussed by Diggle (2000), it is important to distinguish between spatial variation of the risk and clustering. Spatial variation occurs when the risk is not homogeneous in the study region (i.e. all individuals do not have the same risk) but cases appear independently of each other according to this risk surface, whilst clustering occurs when the occurrence of cases is not at random and the presence of a case increases the probability of other cases appearing nearby.

The former methods allow us to inspect a raised incidence in the number of cases around certain pre-specified sources. However, no such source is identified a priori, and a different type of test is required to assess clustering in the cases.

Diggle and Chetwynd (1991) propose a test based on the homogeneous K-function to assess clustering of the cases as compared to the controls. The null hypothesis is as before, that is cases and controls are two IPP that have the same intensities up to a proportionality constant. Hence, they will produce the same K-functions. Note that the inverse is not always true, that is two point processes with the same homogeneous K-function can be completely different (Baddeley and Silverman, 1984). Diggle and Chetwynd (1991) take the difference of the two K-functions to evaluate whether the cases tend to cluster after considering the inhomogeneous distribution of the population: $D(s) = K_1(s) - K_0(s)$, where $K_1(s)$ and $K_0(s)$ are the homogeneous K-functions of cases and controls, respectively.

The test statistic is

$$D = \int_A \frac{D(s)}{\mathrm{var}[D(s)]^{1/2}} \, \mathrm{d}s,$$

where $\mathrm{var}[D(s)]$ is the variance of $D(s)$ under the null hypothesis. Diggle and Chetwynd (1991) compute the value of this variance under random labelling of cases and controls so that the significance of the test statistic can be assessed. Note that under the null hypothesis the expected value of the test statistic D is zero. Finally, the integral is approximated in practice by a discrete sum at a set of finite distances, as the T statistic was computed before.

Significant departure from 0 means that there is a difference in the distribution of cases and controls, with clustering occurring at the range of those distances for which $D(s) > 0$. Furthermore, pointwise envelopes can be provided for the test statistic by the same Monte Carlo test so that the degree of clustering can be assessed. Function `Kenv.label` also provides envelopes for the difference of the K-functions but it does not carry out any test of significance.

```
> s <- seq(0, 0.15, by = 0.01)

> khcases <- khat(coordinates(cases), pbdry, s)
> khcontrols <- khat(coordinates(controls), pbdry, s)
> khcov <- khvmat(coordinates(cases), coordinates(controls),
+     pbdry, s)
> T0 <- sum(((khcases - khcontrols))/sqrt(diag(khcov)))


> niter <- 99
> T <- rep(NA, niter)

> khcasesrel <- matrix(NA, nrow = length(s), ncol = niter)
> khcontrolsrel <- matrix(NA, nrow = length(s), ncol = niter)
> for (i in 1:niter) {
+     idxrel <- sample(spasthma$Asthma) == "case"
+     casesrel <- coordinates(spasthma[idxrel, ])
+     controlsrel <- coordinates(spasthma[!idxrel, ])
+     khcasesrel[, i] <- khat(casesrel, pbdry, s)
+     khcontrolsrel[, i] <- khat(controlsrel, pbdry, s)
+     khdiff <- khcasesrel[, i] - khcontrolsrel[, i]
+     T[i] <- sum(khdiff/sqrt(diag(khcov)))
+ }

> pvalue <- (sum(T > T0) + 1)/(niter + 1)
```

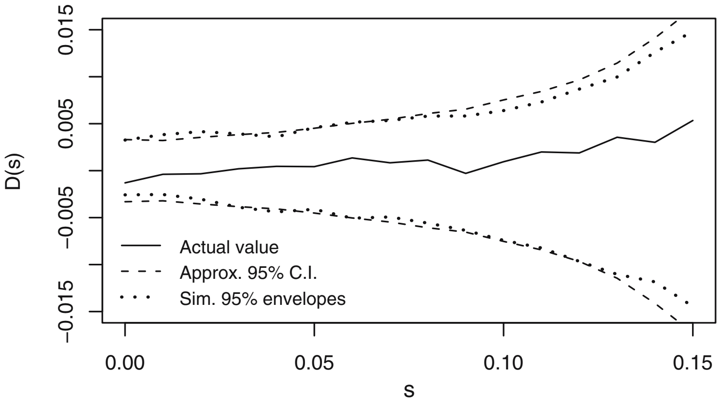

Fig. 7.12. Actual value of $D(s)$ with approximate 95% confidence intervals and 95% envelopes

The p-value for this data set is 0.36, meaning that there is no significant difference between the distribution of cases and controls.The outcome is consistent with the fact that the observed K-function is contained by the simulation envelopes and approximated 95% confidence intervals, as shown in Fig. 7.12.

7.5.5 Accounting for Confounding and Covariates

Diggle et al. (2007) propose a similar way of assessing clustering by means of the inhomogeneous K-function $K_{I,\lambda}(s)$ (Baddeley et al., 2000). For an IPP with intensity $\lambda(x)$, it can be estimated as

$$\hat{K}_{I,\lambda}(s) = |A|^{-1} \sum_{i=1}^{n} \sum_{j \neq i} w_{ij}^{-1} \frac{|\{x_j : d(x_i, x_j) \leq s\}|}{\lambda(x_i)\lambda(x_j)}.$$

Note that this estimator is a generalisation of the estimator of the homogeneous K-function from expression (7.3) and that in fact reduces to it when instead of an IPP we have an HPP (the intensity becomes $\lambda(x) = \lambda$). Similarly, the value of $K_{I,\lambda}(s)$ for an IPP with intensity $\lambda(s)$ is πs^2.

In practise the intensity $\lambda(x)$ needs to be estimated either parametrically or non-parametrically, so that the estimator that we use is

$$\hat{K}_{I,\hat{\lambda}}(s) = |A|^{-1} \sum_{i=1}^{n} \sum_{j \neq i} w_{ij}^{-1} \frac{|\{x_j : d(x_i, x_j) \leq s\}|}{\hat{\lambda}(x_i)\hat{\lambda}(x_j)}.$$

Values of $\hat{K}_{I,\hat{\lambda}}(s)$ higher than πs^2 will mean that the point pattern shows more aggregation than that shown by $\lambda(x)$ and values lower than πs^2 reflect more relative homogeneity.

To be able to account for confounding and risk factors, Diggle et al. (2007) propose the use of a semi-parametric estimator of the intensity in a case–control setting. The basic assumption is that controls are drawn from an IPP with spatially varying intensity $\lambda_0(x)$. The cases are assumed to appear as a result of the inhomogeneous distribution of the population, measured by $\lambda_0(x)$, plus other risk factors, measured by a set of spatially referenced covariates $z(x)$. Hence, the intensity of the cases is modelled as

$$\lambda_1(x) = \exp\{\alpha + \beta z(x)\}\lambda_0(x),$$

where α and β are the intercept and covariate coefficients of the model, respectively. When there are no covariates, the intensity of the cases reduces to

$$\lambda_1(x) = \frac{n_1}{n_0}\lambda_0(x).$$

Note that it is possible to use any generic non-negative function $f(z(x);\theta)$ to account for other types of effects

$$\lambda_1(x) = \lambda_0(x) f(z(x);\theta).$$

This way it is possible to model non-linear and additive effects.

To estimate the parameters that appear in the intensity of the cases, we can use the same working variables Y_i that we have used before (see the binary regression estimator in Sect. 7.5.2), with values 1 for cases and 0 for controls. Conditioning on the locations of cases and controls, Y_i is a realisation of a Bernoulli process with probability

$$P(Y_i = 1|x_i, z(x)) = p(x_i) = \frac{\lambda_1(x)}{\lambda_0(x) + \lambda_1(x)} = \frac{\exp\{\alpha + \beta z(x)\}}{1 + \exp\{\alpha + \beta z(x)\}}. \quad (7.6)$$

Hence, conditioning on the locations of cases and controls, the problem is reformulated as a logistic regression and α and β can be estimated using function `glm`.

Baddeley et al. (2000) estimate the intensity non-parametrically and use the same data to estimate both the intensity and the inhomogeneous K-function, but Diggle et al. (2007) show that this can give poor performance in detecting clustering. This problem arises from the difficulty of disentangling inhomogeneous spatial variation of process from clustering of the events (Cox, 1955). Another problem that appears in practise is that the intensities involved must be bounded away from zero. If kernel smoothing is used, a good alternative to the quartic kernel is a Gaussian bivariate kernel.

The following piece of code shows how to estimate the inhomogeneous K-function both without covariates and accounting for hay fever.

```
> glmasthma <- glm(y ~ HayFeverf, data = spasthma, family = "binomial")
> prob <- fitted(glmasthma)
> weights <- exp(glmasthma$linear.predictors)
> library(spatialkernel)
> setkernel("gaussian")
> lambda0 <- lambdahat(coordinates(controls), bwasthma,
+     coordinates(cases), pbdry, FALSE)$lambda
> lambda1 <- weights[spasthma$Asthma == "case"] * lambda0
> ratiocc <- ncases/ncontrols
> kihnocov <- kinhat(coordinates(cases), ratiocc * lambda0,
+     pbdry, s)$k
> kih <- kinhat(coordinates(cases), lambda1, pbdry, s)$k
```

To assess for any residual clustering left after adjusting for covariates, Diggle et al. (2007) suggest the following test statistic:

$$D = \int_0^{s_0} \frac{\hat{K}_{I,\hat{\lambda}_1}(s) - E[s]}{\mathrm{var}(K_{I,\lambda}(s))^{1/2}}\,\mathrm{d}s,$$

$E[s]$ is the expectation of $\hat{K}_{I,\hat{\lambda}_1}(s)$ under the null hypothesis. In principle, it should be πs^2, but when kernel estimators are used in the computation of the intensity, the estimate of $K_{I,\lambda}(s)$ may be biased. $E[s]$ can be computed as the average of all the estimates $\hat{K}_{I,\hat{\lambda}_1}(s)$, which have been obtained during the Monte Carlo simulations (as explained below). $\mathrm{var}(K_{I,\lambda}(s))$ can be computed in a similar way.

The Monte Carlo test proposed by Diggle et al. (2007) is similar to the one that we used in the homogeneous case (see Sect. 7.5.4), with the difference that the re-labelling must be done taking into account the effects of the covariates. That is, when we relabel cases and controls, the probability of being a case will not be the same for all points but it will depend on the values of $z(x)$. In particular, these probabilities are given by (7.6). The values of the covariates are fixed to the values obtained by fitting the model with the observed data set (i.e. they are not re-estimated when the points are re-labelled) because we are only interested in testing for the spatial variation and not that related to the estimation of the coefficients of the covariates.

```
> niter <- 99
> kinhomrelnocov <- matrix(NA, nrow = length(s), ncol = niter)
> kinhomrel <- matrix(NA, nrow = length(s), ncol = niter)
> for (i in 1:niter) {
+     idxrel <- sample(spasthma$Asthma, prob = prob) ==
+         "case"
+     casesrel <- coordinates(spasthma[idxrel, ])
+     controlsrel <- coordinates(spasthma[!idxrel, ])
+     lambda0rel <- lambdahat(controlsrel, bwasthma, casesrel,
+         pbdry, FALSE)$lambda
+     lambda1rel <- weights[idxrel] * lambda0rel
+     kinhomrelnocov[, i] <- kinhat(casesrel, ratiocc *
+         lambda0rel, pbdry, s)$k
+     kinhomrel[, i] <- kinhat(casesrel, lambda1rel, pbdry,
+         s)$k
+ }

> kinhsdnocov <- apply(kinhomrelnocov, 1, sd)
> kihmeannocov <- apply(kinhomrelnocov, 1, mean)
> D0nocov <- sum((kihnocov - kihmeannocov)/kinhsdnocov)
> Dnocov <- apply(kinhomrelnocov, 2, function(X) {
+     sum((X - kihmeannocov)/kinhsdnocov)
+ })
> pvaluenocov <- (sum(Dnocov > D0nocov) + 1)/(niter + 1)

> kinhsd <- apply(kinhomrel, 1, sd)
> kihmean <- apply(kinhomrel, 1, mean)
> D0 <- sum((kih - kihmean)/kinhsd)
> D <- apply(kinhomrel, 2, function(X) {
+     sum((X - kihmean)/kinhsd)
+ })
> pvalue <- (sum(D > D0) + 1)/(niter + 1)
```

Figure 7.13 shows the estimated values of the inhomogeneous K-function plus 95% envelopes under the null hypothesis. In both cases there are no signs of spatial clustering. The p-values are 0.14 (no covariates) and 0.18 (with hay fever). The increase in the p-value when hay fever is used to modulate the intensity shows how it accounts for some spatial clustering. This is consistent with the plots in Fig. 7.13.

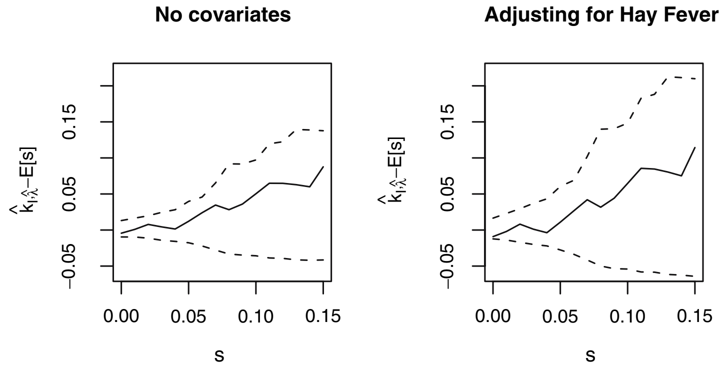

Fig. 7.13. Results of the test based on the inhomogeneous K-function for the asthma data set. The intensity has been modulated to account for the effect of suffering from hay fever

7.6 Further Methods for the Analysis of Point Patterns

In this chapter we have just covered some key examples but the analysis of point patterns with R goes beyond this. Other important problems that we have not discussed here are the analysis of marked point processes (Schabenberger and Gotway 2005, pp. 118–122; Diggle 2003, pp. 82–85), spatio-temporal analysis (see Schabenberger and Gotway 2005, pp. 442–445; Diggle 2006), and complex model fitting and simulation from different point processes (as extensively discussed in Möller and Waagepetersen, 2003). Baddeley et al. (2005) provide a recent compendium of theoretical problems and applications of the analysis of point patterns, including a description of package **spatstat**. Some of the examples described therein should be reproducible using the contents of this chapter.

The Spatial Task View contains a list of other packages for the analysis and visualisation of point patterns. The reader is referred there for updated information.