An Introduction to Point Pattern Analysis

Spatial Statistics

M Besford, A Hunter, R Johnson, LJ Spurling & K Thorn

School of Mathematics and Statistics
University of Sheffield

Contents

Point Pattern Analysis (PPA)

- What is a point pattern?
  A set of points which are distributed in a region of space
- Where do they arise?
  Point patterns can occur in epidemiology, geography, astronomy and biology
- Why use PPA?
  To assess whether the occurence of events in a region follows a systematic pattern, rather than what would be expected if they were randomly distributed

A Motivating Example

In the 19th century, John Snow investigated incidences of cholera during an outbreak in London.

Were the deaths clustered about one pump?

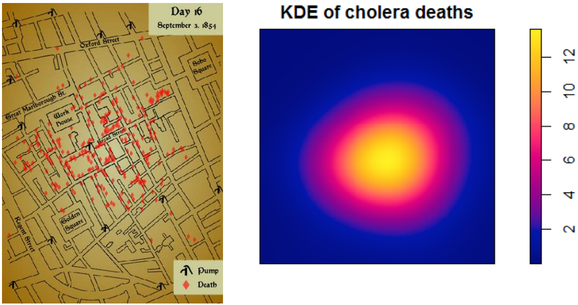
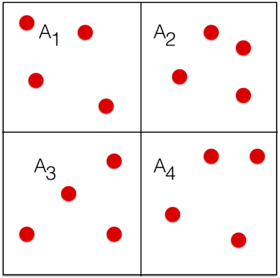


Figure: Map of London showing cholera deaths (left) and kernel density estimate plot of the cholera deaths (right)

Poisson Point Process

- Homogeneous spatial Poisson process
- Multidimensional generalisation of a Poisson process

$A = \cup_i A_i$



\# events in $A = \lambda A \sim Po(\lambda A)$
\# events in $A_i = \lambda A_i \sim Po(\lambda A_i)$
$\hat{\lambda} = n/A$

Complete Spatial Randomness

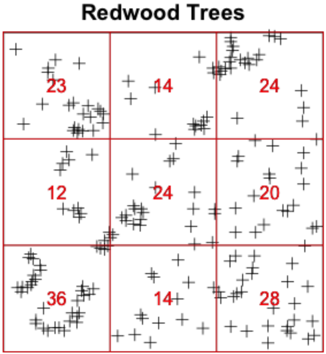
- Complete spatial randomness - homogeneous point process
- Events distributed independently, at random and uniformly over an area
- Alternatives are clustering (attraction) or competition (repulsion)
- `spatstat` package used for datasets and commands

Figure: R datasets that will be used throughout the presentation

Methods of Analysis

- First Order Methods
  Measures the intensity of the points in a region
  - Quadrat Method
  - Kernel Density Estimation
- Second Order Methods
  Measures the spatial dependence between the points in a region
  - G-function (Nearest Neighbour)
  - Ripley's K-function

Quadrat Method



n = number of observations
k = number of subregions
$\bar{x}$ = mean number of observations in a subregion
O_i = observed number of points in subregion i

$$\chi^2_{k-1} = \frac{\Sigma_{i=1}^{k}(O_i - \bar{x})^2}{\bar{x}}$$

```
quadrat.test(dataset,nx=3,ny=3)
```

Limitations:
- Dependent on number and orientation of subregions
- Doesn't consider spatial dependence

Kernel Density Estimation

- Analogous to the `density` function in `R` base package. As h increases, hemispheres sum and estimated density (total height above point) increases. Typically use normal curve rather than hemisphere (Gaussian kernel)
- Boundary problem - have to estimate density for points near the edge

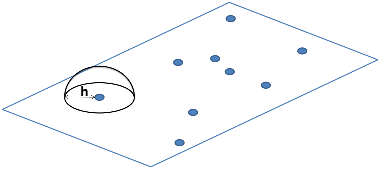


Figure: Illustration of the kernel density estimation

Kernel Density Estimation

We can examine the KDE from each of our datasets - Redwood pines (clustered), Japanese pines (complete spatial randomness) and Cells (regular).

- Plotting KDE ("heat map") gives useful exploratory data analysis tool
- Can see evidence of clustering by colour gradient

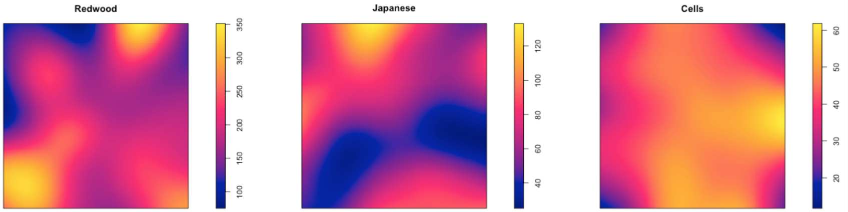


Figure: KDE "heat maps" of the three data sets

G-function (Nearest Neighbour)

- Second order method to measure spatial dependence between events
- Calculates the distance between each event and its nearest neighbouring event

The G-function is the cumulative distribution of these distances:

$$G(r) = \frac{\sum I(d_i \leq r)}{n}$$

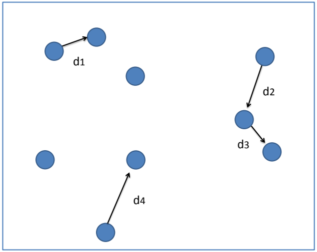


Figure: Illustration of nearest neighbour distances

G-function (Nearest Neighbour)

- The distribution of these nearest neighbour distances can be used to test the null of CSR
- Under $H_0 : CSR,\ G(r) = 1 - e^{\lambda \pi r^2}$
- Monte Carlo simulation methods are used to create envelopes
- What do we expect to see for different patterns?
  - Clustered: Large number of small distances
  - Regular: Small number of distances below a certain value, then many distances of a similar value

G-function Application to Datasets

```
plot(envelope(dataset,Gest))
```

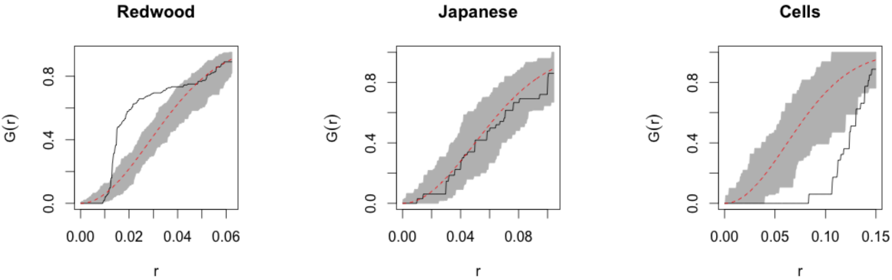


Figure: G-function applied to the three datasets

Limitations:

- Focus is only on nearest event
- Local rather than a global view of the pattern over the region

Ripley's K-function

$$K(r) = \frac{E[N(r)]}{\lambda}$$

$$E[\hat{N}(r)] = \frac{1}{n}\Sigma_{i=1}^{n} p_i, \text{ where } p_i = \Sigma_{i \neq j} I\{||x_i - x_j|| < r\}$$

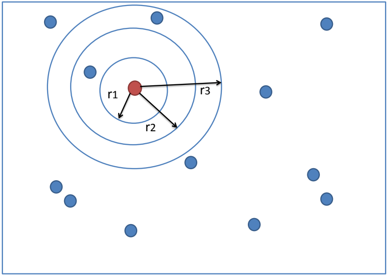


Figure: Illustration of Ripley's K-function

Ripley's K-function

- Without edge correction
  - $\widehat{E[N(r)]} = \frac{1}{n}\Sigma_{i=1}^{n} p_i$
- With edge correction
  - $\widehat{E[N(r)]} = \frac{1}{n}\Sigma_{i=1}^{n} \frac{p_i}{w_{ij}}$, where w_{ij} is the proportion of the circumference of the circle centered at i

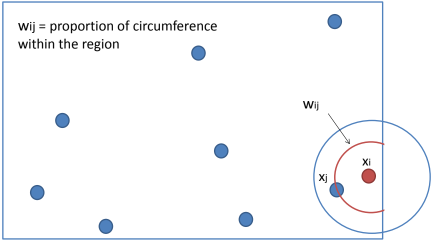


Figure: Illustration of the edge corrections used in Ripley's K-function

Interpreting the K-function

Under the assumption of CSR, the expected number of events within a distance r of an event is $\lambda\pi r^2$ and the K-function becomes
$K(r) = \frac{\lambda\pi r^2}{\lambda} = \pi r^2$

- $K(r) < \pi r^2$ for regular patterns
- $K(r) > \pi r^2$ for clustering

We usually work with $L(r) = \sqrt{\frac{K(r)}{\pi}}$ because

- $Var[L(\hat{r})]$ is approximately constant under CSR
- Under $H_0 : CSR$, $L(r) = r$

K-function Application to Datasets

```
plot(envelope(dataset,Kest))
```

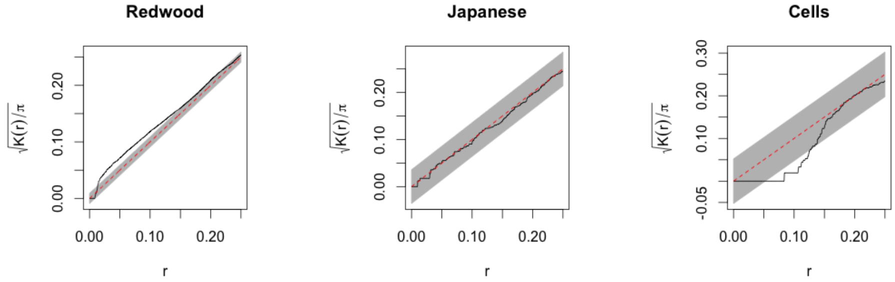


Figure: Ripley's K-function applied to the three datasets

Summary

- PPA involves the analysis of the arrangement of events in a specified region
- Patterns can be random, clustered or regular
- Testing the null hypothesis of CSR can be done in many ways, the most widely accepted being Ripley's K function