Chapter 6
‘Hot Spot’ Analysis I

In this and the next chapter, we describe seven tools for identifying clusters of crime incidents. The discussion has been divided into two chapters primarily because of the length of the discussion. This chapter discusses the concept of a hot spot and four hot spot techniques: the mode, fuzzy mode, nearest neighbor hierarchical clustering, and risk-adjusted nearest neighbor hierarchical clustering. The next chapter discusses STAC, the K-means algorithm, and Anselin’s Local Moran statistics. However, the seven techniques should be seen as a continuum of approaches towards identifying hot spots.

Hot Spots

Typically called hot spots or hot spot areas, these are concentrations of incidents within a limited geographical area that appear over time. Police have learned from experience that there are particular environments that attract drug trading and crimes in larger-than-expected concentrations, so-called crime generators. Sometimes these hot spot areas are defined by particular activities (e.g., drug trading; Weisburd and Green, 1995; Sherman, Gartin and Buerger, 1989; Maltz, Gordon, and Friedman, 1989), other times by specific concentrations of land uses (e.g., skid row areas, bars, adult bookshops, itinerant hotels), and sometimes by interactions between activities and land uses, such as thefts at transit stations or bus stops (Block and Block, 1995; Levine, Wachs and Shirazi, 1986). Whatever the reasons for the concentration, they are real and are known by most police departments.

While there are some theoretical concerns about what links disparate crime incidents together into a cluster, nonetheless, the concept is very useful. Police officers patrolling a precinct can focus their attention on particular environments because they know that crime incidents will continually reappear in these places. Crime prevention units can target their efforts knowing that they will achieve a positive effect in reducing crime with limited resources (Sherman and Weisburd, 1995). In short, the concept is very useful.

Nevertheless, the concept is a perceptual construct. ‘Hot spots’ may not exist in reality, but could be areas where there is sufficient concentration of certain activities (in this case, crime incidents) such that they get labeled as being an area of high concentration. There is not a boundary around these incidents, but a gradient where people draw an imaginary line to indicate the location at which the hot spot starts. In reality, any variable that is measured, such as the density of crime incidents, will be continuous over an area, being higher in some parts and lower in others. Where a line is drawn in order to define a hot spot is somewhat arbitrary.

Statistical Approaches to the Measurement of ‘Hot Spots’

Unfortunately, measuring a hot spot is also a complicated problem. There are literally dozens of different statistical techniques designed to identify ‘hot spots’ (Everitt,
Many, but not all, of the techniques are typically known under the general statistical label of cluster analysis. These are statistical techniques aimed at grouping cases together into relatively coherent clusters. All of the techniques depend on optimizing various statistical criteria, but the techniques differ among themselves in their methodology as well as in the criteria used for identification. Because 'hot spots' are perceptual constructs, any technique that is used must approximate how someone would perceive an area. The techniques do this through various mathematical criteria.

Types of Cluster Analysis (Hot Spot) Methods

Several typologies of cluster analysis have been developed as cluster routines typically fall into several general categories (Everitt, 1974; Çan and Megbolugbe, 1996):

1. Point locations. This is the most intuitive type of cluster involving the number of incidents occurring at different locations. Locations with the most number of incidents are defined as 'hot spots'. CrimeStat includes two point location techniques: the Mode and Fuzzy Mode;

2. Hierarchical techniques (Sneath, 1957; McQuitty, 1960; Sokal and Sneath, 1963; King, 1967; Sokal and Michener, 1958; Ward, 1963; Hartigan, 1975) are like an inverted tree diagram in which two or more incidents are first grouped on the basis of some criteria (e.g., nearest neighbor). Then, the pairs are grouped into second-order clusters. The second-order clusters are then grouped into third-order clusters, and this process is repeated until either all incidents fall into a single cluster or else the grouping criteria fails. Thus, there is a hierarchy of clusters that can be displayed with a dendogram (an inverted tree diagram).

Figure 6.1 shows an example of a hierarchical clustering where there are four orders (levels) of clustering; the visualization is non-spatial in order to show the linkages. In this example, all individual incidents are grouped into first-order clusters which, in turn, are grouped into second-order clusters which, in turn, are grouped into third-order clusters which all converge into a single fourth-order cluster. Many hierarchical techniques, however, do not group all incidents or all clusters into the next highest level. CrimeStat includes two hierarchical techniques: a Nearest Neighbor Hierarchical Clustering routine in this chapter and the Spatial and Temporal Analysis of Crime module (STAC) which will be discussed in chapter 7;

3. Partitioning techniques, frequently called the K-means technique, partition the incidents into a specified number of groupings, usually defined by the user (Thorndike, 1953; MacQueen, 1967; Ball and Hall, 1970; Beale, 1969). Thus, all points are assigned to one, and only one, group. Figure 6.2 shows a partitioning technique where all points are assigned to clusters and are displayed as ellipses. CrimeStat includes one partitioning technique, a K-means partitioning technique;
Figure 6.1:
Hierarchical Clustering Technique
Figure 6.2:
Partitioning Clustering Technique

5. Clumping techniques involve the partitioning of incidents into groups or clusters, but allow overlapping membership (Jones and Jackson, 1967; Needham, 1967; Jardine and Sibson, 1968; Cole and Wishart, 1970).

6. Risk-based techniques identify clusters in relation to an underlying base ‘at risk’ variable, such as population, employment, or active targets (Jefferis, 1998; Kulldorff, 1997; Kulldorff and Nagarwalla, 1995). CrimeStat includes two risk-based technique – a Risk-adjusted Nearest Neighbor Hierarchical Clustering routine, discussed in this chapter, and a Duel Kernel Density method, discussed in chapter 8; and

7. Miscellaneous techniques are other methods that are less commonly used including techniques applied to zones, not incidents. CrimeStat includes Anselin’s Local Moran technique for identifying neighborhood discrepancies (Anselin, 1995).

There are also hybrids between these methods. For example, the Risk-adjusted Nearest Neighbor Hierarchical Clustering routine is primarily a risk-based technique but involves elements of clumping while STAC is primarily a partitioning method but with elements of hierarchical grouping (Block and Green, 1994).

Optimization Criteria

In addition to the different types of cluster analysis, there are different criteria that distinguish techniques applied to space. Among these are:

1. The definition of a cluster - whether it is a discrete grouping or a continuous variable; whether points must belong to a cluster or whether they can be isolated; whether points can belong to multiple clusters.

2. The choice of variables in addition to the X and Y coordinates - whether weighting or intensity values are used to define similarities.

3. The measurement of similarity and distance - the type of geometry being used; whether clusters are defined by closeness or not; the types of similarity measures used.

4. The number of clusters - whether there are a fixed or variable number of clusters; whether users can define the number or not.
5. The geographical scale of the clusters - whether clusters are defined by small or larger areas; for hierarchical techniques, what level of abstraction is considered optimal.

6. The initial selection of cluster locations ('seeds') - whether they are mathematically or user defined; the specific rules used to define the initial seeds.

7. The optimization routines used to adjust the initial seeds into final locations - whether distance is being minimized or maximized; the specific algorithms used to readjust seed locations.

8. The visual display of the clusters, once extracted - whether drawn by hand or by a geometrical object (e.g., an ellipse, a convex hull); the proportion of cases represented in the visualization.

This is not the place to provide a comprehensive review of cluster techniques. Nevertheless, it should be clear that with the several types of cluster analysis and with the many criteria that can be used for any particular technique provides a large number of different techniques that could be applied to an incident data base. It should be realized that there is not a single solution to the identification of hot spots, but that different techniques will reveal different groupings and patterns among the groups. A user must be aware of this variability and must choose techniques that can complement other types of analysis. It would be very naive to expect that a single technique can reveal the existence of hot spots in a jurisdiction which are unequivocally clear. In most cases analysts are not even sure why there are hot spots in the first place and, until that is solved, it would be unreasonable to expect a mathematical or statistical routine to solve that problem.

Cluster Routines in CrimeStat

Because of the variety of cluster techniques, CrimeStat includes seven techniques that cover the range of techniques that have been used:

1. The Mode
2. The Fuzzy Mode
3. Nearest neighbor hierarchical clustering
4. Risk-adjusted nearest neighbor hierarchical clustering
5. The Spatial and Temporal Analysis of Crime (STAC) module
6. K-means clustering
7. Anselin’s Local Moran statistic

These are not the only techniques, of course, and analysts should use them as complements to other types of analysis. Because of the number of routines, these routines have been allocated to two different setup tabs in CrimeStat called ‘Hot Spot’ Analysis I and ‘Hot Spot’ Analysis II. However, they should be seen as one collection of similar techniques. This chapter will discuss the first four of these. Figure 6.3 shows the ‘Hot Spot’ Analysis I page.
Figure 6.3: 'Hot Spot' Analysis I Screen
Mode

The mode is the most intuitive type of hot spot. It is the location with the largest number of incidents. The CrimeStat Mode routine calculates the frequency of incidents occurring at each unique location (a point with a unique X and Y coordinate), sorts the list, and outputs the results in rank order from the most frequent to the least frequent.

Only locations that are represented in the primary file are identified. The routine outputs a 'dbf' file that includes four variables:

1. The rank order of the location with 1 being the location with the most incidents, 2 being the location with the next most incidents, 3 being the location with the third most incidents, and so forth until those locations that have only one incident each;

2. The frequency of incidents at the location. This is the number of incidents occurring at that location;

3. The X coordinate of the location; and

4. The Y coordinate of the location.

To illustrate, table 6.1 presents the formatted output for the ten most frequent locations for motor vehicle thefts in the Baltimore region in 1996 (the rest were ignored) and figure 6.4 maps the ten locations.\(^1\) The map displays the locations with a round symbol, the size of which is proportional the number of incidents. Also, the number of incidents at the location is displayed. These vary from a high of 43 vehicle thefts at location number 1 to a low of 15 vehicle thefts at location number 10. In order to know what these locations represent, the user will have to overlay other GIS layers over the points. In the example, of the ten locations, eight are at shopping centers, one is the parking lot of a train station, and one is the parking lot of a large organization.

The mode is a very simple measure, but one that can be very useful. In the example, it’s clear that most vehicle thefts occur at institutional settings, where there are a collection of parked vehicles. In the case of the shopping centers, the Baltimore County Police Department are aware of the number of vehicles stolen at these locations and work with the shopping center management offices to try to reduce the thefts. It also turns out that shopping centers are the most frequent locations for stolen vehicle retrievals, so it works both ways.

Fuzzy Mode

The usefulness of the mode, however, is dependent on the degree of resolution for the geo-referencing of incidents. In the case of the Baltimore vehicle thefts, thefts locations
Figure 6.4: Ten Most Frequent Locations for Motor Vehicle Theft
Symbol Area Proportional to Frequency

Features
- Arterial
- Auto theft location
- 10 most frequent locations
- Baltimore County

Miles
0 2 4
Table 6.1

Mode Output for
Most Frequent Locations for Motor Vehicle Thefts
Baltimore: 1990

Mode:

--------

Sample size............: 14853
Measurement type.......: Direct
Start time.............: 12:46:15 PM, 07/15/2001
End time...............: 12:50:19 PM, 07/15/2001

Displaying 45 results(s) starting from 1 (ONLY 10 SHOWN)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Freq</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>43</td>
<td>-76.75070</td>
<td>39.31150</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>-76.47100</td>
<td>39.37410</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>-76.48800</td>
<td>39.33720</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>-76.60150</td>
<td>39.40420</td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td>-76.78770</td>
<td>39.40460</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>-76.65170</td>
<td>39.29270</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>-76.73190</td>
<td>39.28800</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
<td>-76.53630</td>
<td>39.30600</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>-76.70260</td>
<td>39.35600</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>-76.51280</td>
<td>39.29270</td>
</tr>
</tbody>
</table>

were assigned a single point at the address. Thus, all thefts occurring at any one shopping center are assigned the same X and Y coordinates. However, there are situations when the assignment of a coordinate will not be a good indicator of the hot spot location. For example, assigning the vehicle theft location to a particular stall in a parking lot will lead to few, if any, locations coming up more than once. In this case, the mode would not be a useful statistic at all. Another example is assigning the vehicle theft location for the parking lot of a multi-building apartment complex to the address of the owner. In this case, what is a highly concentrated set of vehicle thefts become dispersed because the owners live in different buildings with different addresses.

Consequently, CrimeStat includes a second point location hot spot routine called the Fuzzy Mode. This allows the user to define a small search radius around each location to include events that occur around or near that location. For example, a user can put a 50 yard or 100 meter search radius and the routine will calculate the number of incidents that occur at each location and within a 50 yard or 100 meter radius.

The aim of the statistic is to allow the identification of locations where a number of incidents may occur, but where there may not be precision in measurement.² For example,
if several apartment complexes share a parking lot, any vehicle theft in the lot may be assigned to the address of the owner, rather than to the parking lot. In this case, the measurement is imprecise. Plotting the location of the vehicle thefts will make it appear that there are multiple locations, when, in fact, there is only approximately one.

Another example would be the measurement of motor vehicle crashes that all occur at a single intersection. If the measurement of the location is very precise, the crashes could be assigned to slightly different locations when, in fact, they occurred at more or less the same location. In other words, the fuzzy mode allows a flexible classification of a location where the analyst can vary slightly the area around a location.

The fuzzy mode output file is also a ‘dbf’ file and, like the mode, also includes four output variables:

1. The rank order of the location with 1 being the location with the most incidents, 2 being the location with the next most incidents, 3 being the location with the third most incidents, and so forth until only those locations which have only one incident each;

2. The frequency of incidents at the location. This is the number of incidents occurring at that location;

3. The X coordinate of the location; and

4. The Y coordinate of the location.

Note, that allowing a search radius around a location means that incidents are counted multiple times, one for each radius they fall within. If used carefully, the fuzzy mode can allow the identification of high incident locations more precisely than the mode routine. But, because of the multiple counting of incidents that occurs, the frequency of incidents at locations will change, compared to the mode, as well as possibly the hierarchy.

To illustrate this, figure 6.5 maps the top 13 locations for vehicle thefts identified by the fuzzy mode routine using a search radius of 100 yards. Thirteen locations are included because four were tied for number 10. The 13 locations are displayed by a magenta triangle and are compared to the 10 locations identified by the mode (blue circle). Three of the locations identified by the fuzzy mode routine are at the same approximate locations as that identified by the mode, but the remaining eight locations are clustered at a place not identified by the mode.

Figure 6.6 zooms in to display the eight clustered locations. This is a small regional mall within Baltimore city that has a subway station, a Maryland state motor vehicle administration office, and a parole/probation office. There are multiple parking lots located within the mall. Within this space, approximately 29 vehicle thefts occurred in 1996.
Figure 6.5:
Most Frequent Small Zones for Motor Vehicle Theft
Search Radius of 100 Yards
Figure 6.6:
Most Frequent Small Zones for Motor Vehicle Theft
Search Radius of 100 Yards
The fuzzy mode has identified a general location where there are multiple sub-locations in which vehicle thefts occur.

In other words, the fuzzy mode allows the identification of small hot spot areas, rather than exact locations. But, because all points within the user-defined search area are counted, points are counted multiple times. Thus, any one location may not have a sufficient number of incidents to be grouped in the 'top 10' by itself, but, because it is close to other locations that have incidents occurring, it may be elevated to the 'top 10' due to its adjacency to these other incident locations.

Still, the user must be careful in the analysis. By changing the search radius, the number of incidents counted for any one location changes as well as it’s order in the hierarchy. For example, when a quarter mile search radius was used, all top locations occurred within a short distance of each other (not shown).

Nearest Neighbor Hierarchical Clustering (Nnh)

The nearest neighbor hierarchical clustering (Nnh) routine in CrimeStat identifies groups of incidents that are spatially close. It is a hierarchical clustering routine that clusters points together on the basis of a criteria. The clustering is repeated until either all points are grouped into a single cluster or else the clustering criteria fails. Hierarchical clustering methods are among the oldest cluster routines (Everitt, 1974; King, 1967; Systat, 2000). Among the clustering criteria that have been used are the nearest neighbor method (Johnson, 1967; D’andrade, 1978), farthest neighbor, the centroid method (King, 1967), median clusters (Gowers, 1967), group averages (Sokal and Michener, 1958), and minimum error (Ward, 1967).

The CrimeStat Nnh routine uses a method that defines a threshold distance and compares the threshold to the distances for all pairs of points. Only points that are closer to one or more other points than the threshold distance are selected for clustering. In addition, the user can specify a minimum number of points to be included in a cluster. Only points that fit both criteria - closer than the threshold and belonging to a group having the minimum number of points, are clustered at the first level (first-order clusters).

The routine then conducts subsequent clustering to produce a hierarchy of clusters. The first-order clusters are themselves clustered into second-order clusters. Again, only clusters that are spatially closer than a threshold distance (calculated anew for the second level) are included. The second-order clusters, in turn, are clustered into third-order clusters, and this re-clustering process is continued until either all clusters converge into a single cluster or, more likely, the clustering criteria fails.

Criteria 1: Threshold Distance

The first criteria in identifying clusters is whether points are closer than a specified threshold distance. There are two choices in selecting the threshold distance: 1) a random nearest neighbor distance (the default) and 2) a fixed distance.
Random nearest neighbor distance

The default choice to use the expected random nearest neighbor distance for first-order nearest neighbors. The user specifies a one-tailed confidence interval around the random expected nearest neighbor distance. The t-value corresponding to this probability level, t, is selected from the Student’s t-distribution under the assumption that the degrees of freedom are at least 120.³

This selection is controlled by a slide bar under the routine (see Figure 6.3). From chapter 5, the mean random distance was defined as

$$\text{Mean Random Distance} = d(\text{ran}) = 0.5 \text{ SQRT} \left[ \frac{A}{N} \right]$$

where A is the area of the region and N is the number of incidents. The confidence interval around that distance is defined as

$$\text{Confidence Interval for Mean Random Distance} = \text{Mean Random Distance} \pm t \times \text{SE}_{d(\text{ran})}$$

$$= 0.5 \text{ SQRT} \left[ \frac{A}{N} \right] \pm t \left[ \frac{0.26136}{\text{SQRT} \left[ \frac{N^2}{A} \right]} \right]$$

where A is the area of the region, N is the number of incidents, t is the t-value associated with a probability level in the Student’s t-distribution.

The lower limit of this confidence interval is

$$\text{Lower Limit of Confidence Interval for Mean Random Distance} = 0.5 \text{ SQRT} \left[ \frac{A}{N} \right] - t \left[ \frac{0.26136}{\text{SQRT} \left[ \frac{N^2}{A} \right]} \right]$$

and the upper limit of this confidence interval is

$$\text{Upper Limit of Confidence Interval for Mean Random Distance} = 0.5 \text{ SQRT} \left[ \frac{A}{N} \right] + t \left[ \frac{0.26136}{\text{SQRT} \left[ \frac{N^2}{A} \right]} \right]$$
The confidence interval defines a probability for the distance between any pair of points. For example, for a specific one-tailed probability, p, fewer than p% of the incidents would have nearest neighbor distances smaller than this selected limit if the distribution was spatially random. If the data were spatially random and if the mean random distance is selected as the threshold criteria (the default position on the slide bar), approximately 50% of the pairs will be closer than this distance. For randomly distributed data, if a p≤.05 level is taken for t (two steps to the left of the default or the fifth in from the left), then only about 5% of the pairs would be closer than the threshold distance. Similarly, if a p≤.75 level is taken for t (one step to the right of the default or the fifth in from the right), then about 75% of the pairs would be closer than the threshold distance.

In other words, the threshold distance is a probability level for selecting any two points (a pair) on the basis of a chance distribution. The slide bar has 12 levels and is associated with a probability level for a t-distribution from a sample of 120 or larger. From the left, the p-values are approximately (Table 6.2):

<table>
<thead>
<tr>
<th>Scale Bar Position</th>
<th>Probability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00001</td>
<td>Far left point of slide bar</td>
</tr>
<tr>
<td>2</td>
<td>0.0001</td>
<td>Second from left</td>
</tr>
<tr>
<td>3</td>
<td>0.001</td>
<td>Third from left</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>Fourth from left</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>Fifth from left</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>Sixth from left</td>
</tr>
<tr>
<td>7</td>
<td>0.5</td>
<td>Sixth from right (default value)</td>
</tr>
<tr>
<td>8</td>
<td>0.75</td>
<td>Fifth from right</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>Fourth from right</td>
</tr>
<tr>
<td>10</td>
<td>0.95</td>
<td>Third from right</td>
</tr>
<tr>
<td>11</td>
<td>0.99</td>
<td>Second from right</td>
</tr>
<tr>
<td>12</td>
<td>0.999</td>
<td>Far right point of slide bar</td>
</tr>
</tbody>
</table>

Taking a broader conception of this, if there is a spatially random distribution, then for all distances between pairs of points, of which there are

\[
\frac{N (N-1)}{2}
\]

combinations, fewer than p% of the pairs will be shorter than this threshold distance.

This does not mean, however, that the probability of finding a cluster is equal to this probability. It only indicates the probability of selecting two points (a pair) on the basis of a
chance distribution. If additional points are to be included in the cluster, then the probability of obtaining the cluster will be less. Thus, the probability of selecting three points or four points or more points on the basis of chance will be much smaller.

Note that it is very important that area be defined correctly for this routine to work. If the user defines the area on the measurement parameters page (see chapter 3), the Nnh routine uses that value to calculate the threshold distance. If the user does not define the area on the measurement parameters page, the routine calculates the area from the minimum and maximum X/Y values (the bounding rectangle). In either case, the routine will be able to calculate a threshold distance and run the routine.

However, if the area units are defined incorrectly on the measurement parameters page, then the routine will certainly calculate the threshold distance wrongly. For example, if data are in feet but the area on the measurement parameters page are defined in square miles, most likely the routine will not find any points that are farther apart the threshold distance since that distance is defined in miles. In other words, it is essential that the area units be consistent with the data for the routine to properly work.

Fixed distance

The second choice in selecting a threshold distance is to choose a fixed distance (in miles, nautical miles, feet, kilometers, or meters). The uses checks the “Fixed distance” box and selects a threshold distance. The main advantage in this method is that the search radius can be specified exactly. This is useful for comparing the number of clusters for different distributions (e.g., the number of robbery hot spots compared to burglary hot spots using a search radius of 0.5 miles). The main disadvantage of this method is that the choice of a threshold is subjective. The larger the distance that is selected, the greater the likelihood that clusters will be found by chance. Of course, this can be tested using a Monte Carlo simulation (see below).

Criteria 2: Minimum Number of Points

Whatever method is used for selecting a threshold distance, a second criteria is the minimum number of points that are required for each cluster. This criteria is used to reduce the number of very small clusters. With large data sets, hundreds, if not thousands, of clusters can be found if only points are selected by being closer than a threshold distance. To minimize numerous very small clusters as well as reduce the likelihood that clusters could be found by chance, the user can set a minimum number restriction. The default is 10. By decreasing this number, more clusters are produced; conversely, by increasing this number, fewer clusters are produced. The routine will only include points in the final clustering that are part of groups (or clusters) in which the minimum number is found.

First-order clustering

Using these criteria, CrimeStat constructs a first-order clustering of the points. For each first-order cluster, the center of minimum distance is output as the cluster center, which can be saved as a ‘.dbf’ file.
Second and higher-order clusters

The first-order clusters are then tested for second-order clustering. The procedure is similar to first-order clustering except that the cluster centers are now treated as ‘points’ which themselves are clustered. The process is repeated until no further clustering can be conducted, either all sub-clusters converge into a single cluster, or the threshold distance criteria fails, or there are fewer than four seeds in the higher-order cluster.

Visualizing the Cluster Output

To identify the approximate cluster location, CrimeStat III allows the cluster to be output as either an ellipse, a convex hull, or both.

Ellipse output

A standard deviational ellipse is calculated for each cluster (see chapter 4 for definition). The user can choose between 1X (the default), 1.5X, and 2X. Typically, one standard deviation will cover more than 50% of the cases, one and a half standard deviations will cover more than 90% of the cases, and two standard deviations will cover more than 99% of the cases, although the exact percentage will depend on the distribution. The user specifies the number of standard deviations to save as ellipses in ArcView ‘.shp’, MapInfo ‘.mif’ or Atlas*GIS ‘.bna’ formats.

In general, use a 1X standard deviational ellipse since 1.5X and 2X standard deviations can create an exaggerated view of the underlying cluster. The ellipse, after all, is an abstraction from the points in the cluster that may be arranged in an irregular manner. On the other hand, for a regional view, a 1X standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.

Convex hull output

A convex hull is calculated for each cluster (see chapter 4 for definition). The convex hull draws a polygon around the points in the cluster. It is a literal definition of the cluster, as opposed to the ellipse which is an abstraction. The convex hull can be saved in ArcView ‘.shp’, MapInfo ‘.mif’ or Atlas*GIS ‘.bna’ formats.

Ellipse or convex hulls

In previous versions of CrimeStat, only ellipses were used for cluster graphical output. With the addition of a convex hull, the user can visualize the cluster in two different ways. There are advantages and disadvantages of each approach. The convex hull has the advantage of being a polygon that corresponds exactly to the cluster. For neighborhood level analysis, it is probably preferable to the ellipse, which is an abstraction. On the other hand, any convex hull is based on a sample (e.g., this year’s robberies compared to last year’s robberies) and like any sample will vary from one instance to
another. It may not capture all the space associated with the hot spot. The shape also is often un-intuitive, following the outline of the incidents. The ellipse, on the other hand, is more general and will usually be more stable from year to year. It usually looks better on a map or at least users seem to understand it better: it is a more familiar graphical object than an irregular polygon. The biggest disadvantage to an ellipse is that it forces a certain shape on the data, whether there are incidents in every part of it or not. So, in extreme cases, one finds ellipses that go outside of study area boundaries or extend into reservoirs or lakes or other features which are logically impossible.

In short, the user needs to balance the generality and visual familiarity of an ellipse with the limits of the actual hot spot. Probably for a small scale, regional perspective, the ellipses are more adequate and are preferable since a viewer can quickly see where the hot spots are located. For detailed neighborhood-level work, however, the convex hull is probably better since it shows where the incidents actually occurred.

Guidelines for Selecting Parameters

In the Nnh routine, the user has to define three parameters - the threshold distance, the minimum number of points, and the visual output of the hot spots. For a fixed threshold distance, the user has to choose something that is meaningful. For crime incidents, probably the threshold distance should not be more than 0.5 miles and, preferably, smaller.

If the random nearest neighbor distance is used as a threshold, the p-value is selected with a likelihood slider bar (see figure 6.3). This bar indicates a range of p-values from 0.00001 (i.e., the likelihood of obtaining a pair by chance is 0.001%) to 0.999 (i.e., the likelihood of obtaining a pair by chance is 99.9%). The slider bar actually controls the value of $t$ in equation 6.3, which varies from -3.719 to +3.090. The smaller the $t$-value, the smaller the threshold distance. With smaller threshold distances, fewer clusters are extracted, which are typically smaller (although not always).

If only pairs of points were being grouped, then the threshold distance would be critical. Thus, if the default $p \leq .5$ value is selected, then about half the pairs would be selected by chance if the data were truly random. However, since there are a minimum number of points that are required, the likelihood of finding a cluster with the minimum number of points is much smaller. The higher the minimum number that is required, the smaller the likelihood of obtaining a cluster by chance.

Therefore, one can think of the slide bar as a filter for grouping points. One can make the filter smaller (moving the slide bar to the left) or larger (moving the slide bar to the right). There will be some effect on the final number of clusters, but the likelihood of obtaining a cluster by chance will be generally low. Statistically, there is more certainty with small threshold distances than with larger ones using this technique. Thus, a user must trade off the number of clusters and the size of an area that defines a cluster with the likelihood that the result could be due to chance.
This choice will depend on the needs of the user. For interventions around particular locations, the use of a small threshold distance may actually be appropriate; some of the ellipses seen in figure 6.7 below cover only a couple of street segments. These define micro-neighborhoods or almost pure hot spot locations. On the other hand, for a patrol route, for example, a cluster the size of several neighborhoods might be more appropriate. A patrol car would need to cover a sizeable area and having a larger area to target might be more appropriate than a ‘micro’ environment. However, there will be less precision with a larger cluster size covering this type of area.

A second criterion is the minimum number of points that are required to define a cluster. If a cluster does not have this minimum number, CrimeStat will ignore the seed location. Without this criteria, the Nnh routine could identify clusters of two or three incidents each. A hot spot of this size is usually not very useful. Consequently, the user should increase the number to ensure that the identified cluster represents a meaningful number of cases. The default value is 10, but the user can type in any other value.

The user may have to experiment with several runs to get a solution that appears right. As a rule of thumb, start with the default settings. If there appears to be too many clusters, tighten up the criteria by selecting a lower probability for grouping a pair by chance (i.e., shifting the threshold distance to the left) or increasing the minimum number of points required to be defined as a cluster (e.g., from 10 to 20). On the other hand, if there appears to be too few clusters, loosen the criteria by selecting a higher probability for grouping pairs by chance (i.e., shifting the threshold distance to the right) or decreasing the minimum number of points in a cluster (e.g., from 10 to 5). Then, once an appropriate solution has been found, the user can fine tune the results by slight changes.

In general, the minimum number of points criteria is more critical for the number of clusters than the threshold distance, though the latter can also influence the results. For example, with the 1996 Baltimore County robbery data set (N=1181 incidents), a minimum of 26 and a maximum of 28 clusters were found by changing the threshold distance from the minimum p-value (p<0.00001) to the maximum p-value (p>0.999). On the other hand, changing the minimum number of points per clusters from 10 to 20 reduced the number of clusters found (with the default threshold distance) from 26 to 11.

The third criterion is the visual display of the clusters. The convex hull is literal; it will draw a polygon around the points in the cluster. The ellipse, on the other hand, requires a decision by the user on the number of standard deviations to be displayed. The choices are 1X (the default), 1.5X and 2X standard deviations. Typically, one standard deviation will cover more than 50% of the cases, one and a half standard deviations will cover more than 90% of the cases, and two standard deviations will cover more than 99% of the cases, although the exact percentage will depend on the distribution.

In general, use a one standard deviational ellipse since 1.5X and 2X standard deviations can create an exaggerated view of the underlying cluster. On the other hand, for a regional view, a one standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.
Nnh Output Files

The Nnh routine has six outputs. First, for each cluster that is identified, the hierarchical order and the cluster number. Second, for each cluster that is calculated, CrimeStat calculates the mean center of the cluster. Only 45 of the seed locations are displayed on the screen. The user can scroll down or across by adjusting the horizontal and vertical slider bars and clicking on the Go button. This can be saved as a '.dbf' file. Third, the standard deviational ellipses of the clusters is shown, whether the graphical output is an ellipse or a convex hull. The size of the ellipses are determined by the number of standard deviations to be calculated (see above). Fourth, the number of points in the cluster. Fifth, the area of the ellipse and, sixth, the density of the cluster (number of points divided by area).

The ellipses and convex hulls can be saved in ArcView ‘.shp’, MapInfo ‘.mif’ or Atlas*GIS ‘.bna’ formats. Because there are also orders of clusters (i.e., first-order, second-order, etc.), there is a naming convention that distinguishes the order.

For the ellipses, the convention is

Nnh<O><username>

where O is the order number and username is a name provide by the user. Thus,

Nnh1robbery

are the first-order clusters for a file called ‘robbery’ and

Nnh2NightBurglaries

are the second-order clusters for a file called ‘NightBurglaries’. Within files, clusters are named

Nnh<O>Ell<N><username>

where O is the order number, N is the ellipse number and username is the user-defined name of the file. Thus,

Nnh1Ell10robbery

is the tenth ellipse within the first-order clusters for the file ‘robbery’ while

Nnh2Ell1NightBurglaries

is the first ellipse within the second-order clusters for the file ‘NightBurglaries’.
For the convex hulls, the name will be output with a ‘CNNH1’ prefix for the first-order clusters, a ‘CNNH2’ prefix for the second-order clusters, and a ‘CNNH3’ prefix for the third-order clusters. Higher-order clusters will index only the number.

In other words, names of files and features can get complicated. The easiest way to understand this, therefore, is to import the file into one of the GIS packages and display it.

Example 1: Nearest neighbor hierarchical clustering of burglaries

The Nnh routine was applied to the Baltimore County 1996 burglary data (n=6,051 incidents). A default one-tailed probability level of .05 (or 5%) was selected and each cluster was required to contain a minimum of 10 points (the default). CrimeStat returned 122 first-order clusters, 15 second-order clusters and two third-order clusters. Figure 6.7 shows the first-order clusters displayed as 1x standard deviational ellipses. Since the criteria for clustering is the lower limit of the mean random distance, the distances involved are very small, as can be seen. Note, the standard deviational ellipse is defined by the points in the cluster and includes approximately 50% of the points. Thus, the clusters actually extend a little beyond the ellipses.

Figure 6.8 shows the 20 second-order clusters (dashed lines) and the two third-order clusters (double lines). As seen, they cover much larger areas than the first-order clusters. Finally, figure 6.9 shows a part of east Baltimore County where there are 29 first-order clusters (solid line), five second-order clusters (dashed lines), and one third-order cluster (double line). The street network is presented to indicate the scale. Most first-order clusters cover an area the size of a small neighborhood while the second-order clusters cover larger neighborhoods.

To illustrate how the convex hull produces a different visualization, figure 6.10 shows the same clusters as in figure 6.9 but the clusters are displayed as convex hulls rather than ellipses. As seen, the convex hulls are irregular in shape and more limited in geographical spread; they show only the incidents that are clusters. The second-order and third-order clusters are also more defined. From a policing viewpoint, this is probably more useful in that it shows where the hot spot incidents are actually located. As mentioned above, the polygons created by the convex hulls are irregular and are, therefore, less familiar to most people. Consequently, for presentations of crime patterns at a regional level or even neighborhood-level for non-specialists, the ellipses may convey better where the hot spots are located.

Advantages of Hierarchical Clustering

There are four advantages to this technique. First, it can identify small geographical environments where there are concentrated incidents. This can be useful for specific targeting, either by police deployment or community intervention. There are clearly micro-environments that generate crime incidents (Levine, Wachs and Shirazi, 1986; Maltz, Gordon and Friedman, 1989). The technique tends to identify these small environments because the lower limit of the mean random distance is used to group the
Figure 6.7: First-Order Baltimore County Burglary 'Hot Spots': Ellipses
Using Nearest Neighbor Hierarchical Clustering Method
Figure 6.8:
Second- and Third-Order Burglary 'Hot Spots': Ellipses
Using Nearest Neighbor Hierarchical Clustering Method
Figure 6.9:
First, Second- and Third-Order Burglary 'Hot Spots': Ellipses
Using Nearest Neighbor Hierarchical Clustering Method
Figure 6.10:
First, Second- and Third-Order Burglary 'Hot Spots': Convex Hulls
Using Nearest Neighbor Hierarchical Clustering Method
clusters. The user can, of course, control the size of the grouping area by loosening or tightening either the threshold distance or the minimum number of required points. Thus, the sizes of the clusters can be adjusted to fit particular groupings of points.

Second, the technique can be applied to any entire data set, such as for Baltimore County and Baltimore City, and need not only be applied to smaller geographical areas, such as precincts. This increases the ease of use for analysts and can facilitate comparisons between different areas without having to limit arbitrarily the data set.

Third, the linkages between several small clusters can be seen through the second- and higher-order clusters. Frequently, 'hot spots' are located near other 'hot spots' which, in turn, are located near other 'hot spots'. As we’ve seen from the maps of robbery, burglary and motor vehicle thefts in Baltimore County, there are large areas within the County that have a lot of incidents. Within these large areas, there are smaller hot spots and within some of those hot spots, there are even small ones. In other words, there are different scales to the clustering of points - different geographical levels, if you will, and the hierarchical clustering technique can identify these levels.

Fourth, each of the levels imply different policing strategies. For the smallest level, officers can intervene effectively in small neighborhoods, as discussed above. Second-order clusters, on the other hand, are more appropriate as patrol areas; these areas are larger than first-order clusters, but include several first-order clusters within them. If third- or higher-order clusters are identified, these are generally areas with very high concentrations of crime incidents over a fairly large section of the jurisdiction. The areas start to approximate precinct sizes and need to be thought of in terms of an integrated management strategy - police deployment, crime prevention, community involvement, and long-range planning. Thus, the hierarchical technique allows different security strategies to be adopted and provides a coherent way of approaching these communities.

Simulating Statistical Significance

Testing the significance of clusters from the Nnh routine is difficult. Conceptually, using the random nearest neighbor distance for the threshold distance defines the probability that two points could be grouped together on the basis of chance; the test is for the confidence interval around the first-order nearest neighbor distance for a random distribution. If the probability level is p%, then approximately p% of all pairs of points would be found under a random distribution. Under this situation, we would know whether the number of clusters (pairs) that were found were significantly greater than would be expected on the basis of chance.

The problem is, however, that the routine is not just clustering pairs of points, but clustering as many points as possible that fall within the threshold distance. Further, the additional requirement is added that there be a minimum number of points, with the minimum defined by the user. The probability distribution for this situation is not known. Consequently, there is a necessity to resort to a Monte Carlo simulation of randomness under the conditions of the Nnh test (Dwass, 1957; Barnard, 1963).

James L. LeBeau
Administration of Justice
Southern Illinois University at Carbondale

Stephen Schnebly
Criminology & Criminal Justice
University of Missouri – St Louis

The CrimeStat Nearest Neighbor Hierarchical clustering routine and GIS were used for defining, comparing, analyzing, and visualizing changes in drug arrest clusters between 1997 and 1998. Using a minimum cluster size of 25 arrests some of the emerging patterns or relationships include: 1) the overlapping of secondary clusters, but those emerging during 1998 were much larger, especially in the north because of new primary clusters; 2) many primary clusters during 1997 remaining static or increasing in area during 1998; and 3) the disappearing of some 1997 primary clusters during 1998, with new clusters emerging close by implying displacement.
Using Nearest Neighbor Hierarchical Clustering to Identify High Crime Areas Along Commercial Corridors

Philip R. Canter
Baltimore County Police Department
Towson, Maryland

Robberies in Baltimore County had increased by 45% between 1990 and 1999, and by 1997, were the highest on record. In 1997, 73% of all reported robberies in Baltimore County were occurring in commercial areas. The department wanted to target commercial districts with intensive patrol and outreach programs. These high crime commercial districts were identified as Business Patrol Initiative (BPI) areas. A total of 40 police officers working two 8-hour shifts were assigned to BPI areas. Robberies in the BPI areas declined by 26.7% during the first year of the program and another 13.8% one year following the BPI program.

Police analysts used CrimeStat's Nearest Neighbor Hierarchical clustering (Nnh) method to identify high crime areas along commercial corridors. The Nnh routine was very effective in identifying commercial areas having the highest concentration of crime. The clustering also demonstrated that commercial crime was not restricted to county borders; rather, crime crossed municipal boundaries into neighboring jurisdictions. A neighboring jurisdiction was shown the crime cluster map, leading to their decision to implement a similar BPI program.
CrimeStat includes a Monte Carlo simulation routine that produces approximate confidence intervals for the first-order Nnh clusters that has been run; second- and higher-order clusters are not simulated since their structure depends on the first-order clusters. Essentially, the routine assigns N cases randomly to a rectangle with the same area as the defined study area, A, and evaluates the number of clusters according to the defined parameters (i.e., threshold distance and minimum number of points). It repeats this test K times, where K is defined by the user (e.g., 100, 1,000, 10,000). By running the simulation many times, the user can assess approximate confidence intervals for the particular first-order Nnh.

The output includes five columns and twelve rows:

**Columns:**

1. The percentile,
2. The number of first-order clusters found for that percentile,
3. The area of the cluster for that percentile,
4. The number of points in the cluster for that percentile, and
5. The density of points (per unit area) for that percentile.

**Rows:**

1. The minimum (smallest) value obtained,
2. 0.5th percentile,
3. 1st percentile,
4. 2.5th percentile,
5. 5th percentile,
6. 10th percentile,
7. 90th percentile,
8. 95th percentile,
9. 97.5th percentile,
10. 99th percentile,
11. 99.5th percentile, and
12. The maximum (largest) value obtained.

The manner in which percentiles are calculated are as follows. First, over all simulation runs (e.g., 1000), the routine calculates the number of first-order clusters obtained for each run, sorts them in order, and defines the percentiles for the list. Thus, the minimum is the fewest number of clusters obtained over all runs, the 0.5 percentile is the lowest half of a percent for the number of clusters obtained over all runs, and so forth until the maximum number of clusters obtained over all runs. The routine does not calculate second- or higher-order clusters since those are dependent on the first order clustering. Second, within each run, the routine calculates the number of points per cluster, the area of each ellipse, and the density of each ellipse. Then, it groups all clusters together, over all runs, and sorts them into a list. The percentiles for individual clusters are then calculated. Note that the points refer to the cluster whereas the area and density refer to the ellipses, which is a geometrical abstraction from the cluster.
Table 6.3 presents an example. An Nnh run was conducted on the Baltimore robbery data base (N=1181 incidents) using the default threshold distance (p<.5 for grouping a pair by chance) and a minimum number of points of at least five for each cluster. Then, 1000 Monte Carlo runs were conducted with simulated data. For the actual data, the Nnh routine identified 69 first-order clusters and 7 second-order clusters. Table 6.3 presents the parameters for the first ten first-order clusters.

In examining a simulation, one has to select percentiles as choice points. In this example, we use the 95th percentile. That is, we are willing to accept a one-tailed Type I error of only 5% since we are only interested in finding a greater number of clusters than by chance. For the simulation, let's look at each column in turn. Column 2 presents the number of clusters found in each simulation. Over the 1000 runs, there was a minimum of one cluster found (for at least one simulation) and a maximum of 7 clusters found (for at least one simulation). That is, running 1000 simulations of randomly assigned data only yielded between 1 and 7 clusters using the parameters defined in the particular Nnh run. The 95th percentile was 3. It is highly unlikely that the 69 first-order clusters that were identified would have been due to chance. That is, we would have expected at most three of them to have been due to chance. It appears that the robbery data is significantly clustered, though we have only tested significance through a random simulation.

Column 3 shows the areas of clusters that were found over the 1000 runs. For the individual clusters, the simulation showed a range from about 0.04 to 0.38. The 95th percentile was 0.31. In the actual Nnh, the area of clusters varied between 0.05 and 0.27, indicating that all first-order clusters were smaller than the smallest value found in the simulation. In other words, the real clusters are more compact than random clusters even though the random clusters are subject to the same threshold distance as the real data. This is not always true, but, in this case, it is.

Column 4 presents the number of points found per cluster. In the simulations, the numbers varied between 5 and 9 points per cluster. The 95th percentile was 7. With the actual data, the number of points varied between 5 and 40. Thus, some of the clusters could have been due to chance, at least in terms of the number of points per cluster. Analyzing the distribution (not shown), 27 of the 69 clusters had 7 or fewer points. In other words, about 39% had only as many points as might be expected on the basis of a chance distribution. Putting it another way, about 40% of the clusters had more points than would be expected on the basis of chance 95% of the time.

Finally, column 5 presents the density of points found per cluster. Since the output unit is squared miles, density is the number of points per square mile. The simulation presents a range from 15.6 points per square mile to 156.1 points per square mile. The 95th percentile was 73.4 points per square mile. The actual Nnh, on the other hand, finds a range of densities from 27.1 points per square mile to a very high number (11071821 points per square mile). Again, there is overlap between the actual clusters and what might be expected on the basis of chance: 26 out of 69 clusters have densities that are lower than the 95th percentile found in the simulation. Again, about 38% have densities are not different than would be expected on the basis of chance.
Table 6.3

Simulated Confidence Intervals for Nnh Routine
Baltimore County Robberies: N=1181

Nearest Neighbor Hierarchical Clustering:

<table>
<thead>
<tr>
<th>Sample size</th>
<th>1181</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood of grouping pair of points by chance</td>
<td>0.50000 (50.000%)</td>
</tr>
<tr>
<td>Z-value for confidence interval</td>
<td>0.000</td>
</tr>
<tr>
<td>Measurement type</td>
<td>Direct</td>
</tr>
<tr>
<td>Output units</td>
<td>Miles, Squared Miles, Points per Squared Miles</td>
</tr>
<tr>
<td>Clusters found</td>
<td>76</td>
</tr>
<tr>
<td>Simulation runs</td>
<td>1000</td>
</tr>
</tbody>
</table>

Displaying ellipse(s) starting from 1

<table>
<thead>
<tr>
<th>Order</th>
<th>Cluster</th>
<th>Mean X</th>
<th>Mean Y</th>
<th>Rotation</th>
<th>X-Axis</th>
<th>Y-Axis</th>
<th>Area</th>
<th>Points</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-76.44927</td>
<td>39.31455</td>
<td>77.09164</td>
<td>0.28303</td>
<td>0.09636</td>
<td>0.08568</td>
<td>40</td>
<td>466.828013</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-76.60219</td>
<td>39.40050</td>
<td>11.98132</td>
<td>0.11540</td>
<td>0.27452</td>
<td>0.09952</td>
<td>33</td>
<td>331.580616</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>-76.44601</td>
<td>39.30490</td>
<td>16.66988</td>
<td>0.21907</td>
<td>0.16239</td>
<td>0.11176</td>
<td>25</td>
<td>223.684859</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>-76.78123</td>
<td>39.36088</td>
<td>25.36983</td>
<td>0.27643</td>
<td>0.14530</td>
<td>0.12618</td>
<td>29</td>
<td>229.826284</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>-76.73103</td>
<td>39.34319</td>
<td>67.71617</td>
<td>0.19445</td>
<td>0.16058</td>
<td>0.08910</td>
<td>29</td>
<td>295.628310</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>-76.72945</td>
<td>39.28910</td>
<td>79.88383</td>
<td>0.16428</td>
<td>0.25957</td>
<td>0.13396</td>
<td>29</td>
<td>216.476166</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>-76.51486</td>
<td>39.25910</td>
<td>79.83833</td>
<td>0.16428</td>
<td>0.25957</td>
<td>0.13396</td>
<td>29</td>
<td>216.476166</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>-76.45374</td>
<td>39.32106</td>
<td>54.57635</td>
<td>0.15150</td>
<td>0.18261</td>
<td>0.08692</td>
<td>7</td>
<td>80.538112</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>-76.75368</td>
<td>39.31132</td>
<td>89.56994</td>
<td>0.19748</td>
<td>0.22914</td>
<td>0.14216</td>
<td>22</td>
<td>154.753006</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>-76.71641</td>
<td>39.29139</td>
<td>10.45387</td>
<td>0.15048</td>
<td>0.16879</td>
<td>0.07980</td>
<td>14</td>
<td>175.44372</td>
</tr>
</tbody>
</table>

...etc.

Distribution of the number of clusters found in simulation (percentile):

<table>
<thead>
<tr>
<th>Percentile</th>
<th>Clusters</th>
<th>Area</th>
<th>Points</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>1</td>
<td>0.03845</td>
<td>5</td>
<td>15.615111</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
<td>0.04922</td>
<td>6</td>
<td>16.608967</td>
</tr>
<tr>
<td>1.0</td>
<td>1</td>
<td>0.05603</td>
<td>6</td>
<td>17.162252</td>
</tr>
<tr>
<td>2.5</td>
<td>1</td>
<td>0.06901</td>
<td>6</td>
<td>18.570113</td>
</tr>
<tr>
<td>5.0</td>
<td>1</td>
<td>0.08243</td>
<td>6</td>
<td>19.468353</td>
</tr>
<tr>
<td>10.0</td>
<td>1</td>
<td>0.10045</td>
<td>6</td>
<td>21.256559</td>
</tr>
<tr>
<td>90.0</td>
<td>2</td>
<td>0.28706</td>
<td>7</td>
<td>61.173748</td>
</tr>
<tr>
<td>95.0</td>
<td>3</td>
<td>0.31074</td>
<td>7</td>
<td>73.463654</td>
</tr>
<tr>
<td>97.5</td>
<td>3</td>
<td>0.32442</td>
<td>7</td>
<td>87.550868</td>
</tr>
<tr>
<td>99.0</td>
<td>4</td>
<td>0.35279</td>
<td>8</td>
<td>115.460337</td>
</tr>
<tr>
<td>99.5</td>
<td>5</td>
<td>0.36489</td>
<td>8</td>
<td>122.625375</td>
</tr>
<tr>
<td>max</td>
<td>7</td>
<td>0.38424</td>
<td>9</td>
<td>156.056837</td>
</tr>
</tbody>
</table>

6.32
In other words, the simulation suggests that around 60% of the clusters are real with the other 40% being no different than might be expected on the basis of chance. There are far more clusters found in the actual Nnh than would be expected on the basis of chance and they are more compact than would be expected. On the other hand, only about half have densities that are higher than would be expected on the basis of chance.

It should be clear that testing the significance of a cluster analysis is complex. In the example, some of the criteria chosen were definitely different than a chance distribution (as evidenced by the simulation) while other criteria were not very different. In this case, the user would be wise to re-run the Nnh and simulation under tighter conditions, either lowering the threshold distance or increasing the minimum number of points per cluster. With experimentation, it is frequently possible to obtain a solution in which all the criteria are greater than would be expected on the basis of chance.

Limitation to Hierarchical Clustering

There are also limitations to the technique, some technical and others theoretical. First, the method only clusters incidents (points): a weighting or intensity variable will have no effect. Second, the size of the grouping area is dependent on the sample size when the confidence interval around the mean random distance is used as the threshold distance criteria (see equation 4.2). For crime distributions that have many incidents (e.g., burglary), the threshold distance will be a lot smaller than distributions that have fewer incidents (e.g., robbery). In theory, a hot spot is dependent on an environment, not the number of incidents. Thus, that approach does not produce a consistent definition of a hot spot area. Using a fixed distance for the threshold distance can partly overcome this. However, the fixed distance needs to be tested for randomness using the Monte Carlo simulation.

Third, there is a certain arbitrariness in the technique due to the minimum points rule. This implicitly requires the user to define a meaningful cluster size, whether the number of points are 5, 10, 15 or whatever. To some extent, this is how patterns are defined by human beings: with one or two incidents in a small area, people don’t perceive any pattern. As soon as the number of incidents increases, say to 10 or more, people perceive the pattern. This is not a statistical way for defining regularity, but it is a human way. However, it can lead to arbitrariness since two different users may interpret the size of a hot spot differently. Similarly, the selectivity of the p-value, vis-a-vis the Student’s t-distribution, can allow variability between users.

In short, the technique does produce a constant result, but one subject to manipulation by users. Hierarchical techniques are, of course, not the only clustering procedures to allow users to adjust the parameters: in fact, almost all the cluster techniques have this property. But it is a statistical weakness in that it involves subjectivity and is not necessarily consistently applied across users.

Finally, there is no theory or rationale behind the clusters. They are empirical derivatives of a procedures. Again, many clustering techniques are empirical groupings and also do not have any explanatory theory. However, if one is looking for a substantive
The Massachusetts State Police is collecting incident addresses as part of its state-level implementation of the FBI’s National Incident Based Reporting System (NIBRS). They intend to develop a regional and statewide crime mapping and analysis program. As an example of the type of analysis that can be done with the enhanced NIBRS database, the State Police’s Crime Reporting Unit analyzed year 2000 drug arrests for one city in the Commonwealth, focusing on arrests for possession of heroin and marijuana. The arrest locations were plotted, with the size of points proportionate to the amount of drugs seized. A nearest neighbor clustering analysis was done of the data. It indicates that, while there is some small amount of overlap, the arrest locations for the two drug types are generally different.

This type of analysis can be very useful for smaller police agencies that do not have the resources to conduct their own analysis of crime data. It may also prove useful for crime problems with cross-jurisdictional boundaries.
Use of CrimeStat in Crime Mapping in India: An Application for Chennai City Policing

Jaishankar Karuppannan
Department of Criminology & Criminal Justice
Manonmaniam Sundaranar University
Tamil Nadu, India

The present study was done as an implementation of GIS technology in Chennai (Madras), India. In the present study hotspot analysis was done with the help of CrimeStat. We converted the output to Arcview shape files.

When hotspot analysis examined changes over a period of time, the change seemed to be significant. There exists not only a change in the location of the hotspots, but also in their areal extent. The numbers of hotspots also differ over time. The map shows hotspots for residential burglary for both day and night. The hot spots for daytime house break-ins are confined to a smaller area in the west of the city, whereas the hot spots for nighttime residential break-ins are seen in all parts of the city. In particular, the Posh area of Anna Nagar is more prone to daytime burglaries. In this area, a higher proportion of couples work, which appears to make the homes in this neighborhood more open for burglaries.
hot spot defined by a unique constellation of land uses, activities, and targets, the
technique does not provide any insight into why the clusters are occurring or why they
could be related. I will return to this point at the end of the next chapter, but it should be
remembered that these are empirical groupings, not necessarily substantive ones.

Risk-Adjusted Nearest Neighbor Hierarchical Clustering

CrimeStat also includes a risk-adjusted nearest neighbor hierarchical clustering
routine (Rnnh), which is a variation on the Nnh routine discussed above. It combines the
hierarchical clustering capabilities of the Nnh routine with kernel density interpolation
techniques, that are discussed in chapter 8.

The Nnh routine identifies clusters of points that are close together. That is, it will
identify groups of points that are closer together than a threshold distance and in which the
minimum number of points is greater than a user-defined value. Many of these clusters,
however, are due to a high concentration of persons in the vicinity. That is, because the
population is not arranged randomly over a plane, but is, instead, highly concentrated in
population centers, there is a higher likelihood of incidents happening (whatever they are)
simply due to the higher population concentration. In the above examples, many of the
clusters for Baltimore burglaries or vehicle thefts were due primarily to a high
concentration of households and vehicles in the center of the metropolitan area. In fact, one
would normally expect a higher concentration of incidents in the center since there are
more persons residing in the center and, certainly, more persons being concentrated there
during the daytime through employment, shopping, cultural attendance, and other urban
activities.

For many police purposes, the concentration of incidents is of sufficient interest in
itself. Police have to intervene at high incidence locations irrespective of whether there is
also a larger population at those locations. The demands for policing and responding to
community emergency needs is population sensitive since there are more demands where
there are more persons. From a service viewpoint, the concentration of incidents is what is
important.

But for other purposes, the concentration of incidents relative to the baseline
population is of interest. Crime prevention activities, for example, are aimed at reducing
the number of crimes that occur for every area in which they are applied. For these
purposes, the rate of decrease in the number of crimes is the prime focus. Similarly, after-
school programs are aimed at neighborhoods where there is a high risk of crime, whether or
not there is also a large population. In other words, for many purposes, the risk of crime or
other types of incidents is of paramount importance, rather than the volume (i.e., absolute
amount) of crime by itself. If the aim is to assess where there are high risk clusters, then
the Nnh routine is not appropriate.

CrimeStat includes a Risk-adjusted Nearest Neighbor Hierarchical Clustering
routine (or Rnnh) that defines clusters of points that are closer than what would be
expected on the basis of a baseline population. It does this by dynamically adjusting the
threshold distance in the Nnh routine according to the distribution of a second, baseline

6.36
variable. Unlike the Nnh routine where the threshold distance is constant throughout the study area (i.e., it is used to pair point irrespective of where they are within the area), the Rnnh routine adjusts the threshold distance according to what would be expected on the basis of the baseline variable. It is a risk measure, rather than a volume measure.

Dynamic Adjustment of the Threshold Distance

To understand how this works, think of a simple example. In a typical metropolitan area, there are more people living towards the center than in the periphery. There are topographical and social factors that might modify this (e.g., an ocean, a mountain range, a lake), but in general population densities are much higher in the center than in the suburbs. In the next chapter, we will examine the distribution of population and how it affects incidence of crime over an entire metropolitan area. If a different baseline variable were selected than population, for example, employment, one would generally find even higher concentrations since central city employment tends to be very high relative to suburban employment. Thus, if population or employment (or another variable that is correlated with population density) is taken as the baseline, then one would expect more people and, hence, more incidents occurring in the center rather than the periphery. In other words, all other things being equal, there should be more robberies, more burglaries, more homicides, more vehicle thefts, and more of any other type of event in the center than in the periphery of an urban area. This is just a by-product of urban societies.

Using this idea to cluster incidents together, then, intuitively, the threshold distance must be adjusted for the varying population densities. In the center, the threshold must be short since one would expect there to be more persons. Conversely, in the periphery - the far suburbs, the threshold distance must be a lot longer since there are far fewer persons per unit of area. In other words, dynamic adjustment of the threshold grouping distance means changing the distance inversely proportional to the population density of the location: in the center, a high density means a short threshold distance and in the periphery, a low density means a larger threshold distance.

Kernel Adjustment of the Threshold Distance

To implement this logic, CrimeStat overlays a standard grid and uses an interpolation algorithm, based on the kernel density method, to estimate the expected number of incidents per grid cell if the actual incident file was distributed according to the baseline variable. The next chapter discusses in detail the kernel density method and the reader should be familiar with the method before attempting to use the Rnnh routine. If not, the author highly recommends that Chapter 8 be read before reading the rest of this section.

Steps in the Rnnh Routine

The Rnnh routine works as follows:

1. Both a primary and secondary file are required. The primary file are the basic incidents (e.g., robberies) while the secondary file is the baseline
variable (e.g., population of zones: all crimes as a baseline: or another baseline variable). If the baseline variable are zones, the user must define both the X and Y coordinates as well as the variable assigned to the zone (e.g., population); the latter will typically be an intensity or weight variable (see Chapter 3).

2. A grid is defined in the reference file tab of the data setup section (see Chapter 3). The Rnnh routine takes the lower-left and upper-right limits of the grid, but uses a standard number of columns (50).

3. The area of the study is defined in the measurement parameters tab of the data setup section (see Chapter 3). If no area is defined, the routine uses the area of the entire grid.

4. The user checks the Risk-adjusted box under the Nnh routine. The risk variable is estimated with the parameters defined in the Risk Parameters box. These are the kernel parameters. Without going into detail, the user must define:
   A. The method of interpolation, which is the type of kernel used: normal, uniform, quartic, triangular, or negative exponential. The normal distribution is the default.
   B. The choice of bandwidth, whether a fixed or adaptive (variable) bandwidth is used. For a fixed bandwidth, the user must define the size of the interval (e.g., 2 miles). For an adaptive bandwidth, the user must define the minimum sample size to be included in the circle that defines the bandwidth. The default is an adaptive bandwidth with a minimum sample size of 100 incidents.
   C. The output units, which are points per unit of area: squared miles, squared nautical miles, squared feet, squared kilometers, or squared meters. The default is squared miles.
   D. Also, if an intensity or weight variable is used (e.g., the centroids of zones with population being an intensity variable), the intensity or weight box should be checked (be careful about checking both if there are both an intensity and a weight variable).

Consult Chapter 8 for more detail about these parameters.

5. Once the baseline variable (the secondary file) is interpolated to the grid using the above parameters, it is converted into absolute densities (points per grid cell) and re-scaled to the same sample size as the primary incident file. This has the effect of making the interpolation of the baseline variable the same sample size as the incident variable. For example, if there are 1000 incidents in the primary file, the interpolation of the secondary file will be re-scaled so that all grid cells add to 1000 points, irrespective of how many units
the secondary variable actually represented. This creates a distribution for the primary file (the incidents) that is proportional to the secondary file (the baseline variable) if the primary file had the same distribution as the secondary file. It is then possible to compare the actual distribution of the incident variable with the expected distribution if it was similar to the baseline variable.

6. Once the risk parameters have been defined, the selection of parameters is similar to the Nnh routine with one exception.

   A. The threshold probabilities are selected with the scale bar. The probabilities are identical to those in Table 6.2.

   B. However, for each grid cell, a unique threshold distance is defined using formulas similar to 6.1 and 6.2. The difference is, however, that the formulas are applied to each grid cell with a unique distance for each grid cell (formulas 6.5-6.8):

   \[
   \text{Mean Random Distance of Grid Cell } i = d(\text{ran}) = 0.5 \text{ SQRT} \left[ \frac{A_i}{N_i} \right] \tag{6.5}
   \]

   where \( A_i \) is the area of the grid cell and \( N_i \) is the estimated number of points from the kernel density interpolation. Thus, each grid cell has its own unique expected number of points, \( N_i \), its own unique area, \( A_i \) (though, in general, all grid cells will have approximately equal areas), and, consequently, its own unique threshold distance.

   \[
   \text{Confidence Interval for Mean Random Distance of Grid Cell } i = \text{Mean Random Distance of grid cell } i \pm t^* \text{SE}_{d(\text{ran})}
   \]

   \[
   = 0.5 \text{ SQRT} \left[ \frac{A_i}{N_i} \right] \pm t \frac{0.26136}{\text{SQRT}[N_i^{\frac{2}{A_i}}]} \tag{6.6}
   \]

   where the Mean Random Distance of Grid Cell \( i \), \( A_i \) and \( N_i \) are as defined above, \( t \) is the \( t \)-value associated with a probability level in the Student’s \( t \)-distribution (defined by the scale bar).
The lower limit of this confidence interval is

\[
\text{Lower Limit of Confidence Interval for Mean Random Distance of Grid Cell } i = \frac{0.26136}{2} \sqrt{\frac{n}{A_i}} \left( \frac{t}{\sqrt{\frac{N_i}{A_i}}} \right) \] (6.7)

and the upper limit of this confidence interval is

\[
\text{Upper Limit of Confidence Interval for Mean Random Distance of Grid Cell } i = \frac{0.26136}{2} \sqrt{\frac{n}{A_i}} \left( \frac{t}{\sqrt{\frac{N_i}{A_i}}} \right) \] (6.8)

C. In addition, the user defines a minimum sample size for each cluster, as with the Nnh routine.

6. The actual incident points are then identified by the grid cell that they fall within and the unique threshold distance (and confidence interval) for that grid cell. For each pair of points that are compared for distance, there is, however, asymmetry. The unique threshold distance for point A will not necessarily be the same as that for point B. The Rnnh routine, therefore, requires the distance between each pair of points to be the shorter of the two distances between the points.

7. Once pairs of points are selected, the Rnnh routine proceeds in the same way as the Nnh routine.

In other words, points are clustered together according to two criteria. First, they must be closer than a threshold distance. However, the threshold distance varies over the study area and is inversely proportional to the baseline variable. Only points that are closer together than would be expected on the basis of the baseline variable are selected for grouping. Second, clusters are required to have a minimum number of points with the minimum being defined by the user. The result are clusters that are more concentrated than would be expected, not just from chance but, from the distribution of the baseline variable. These are high risk clusters.

Area must be defined correctly

Note that it is very important that area be defined correctly for this routine to work. If the user defines the area on the measurement parameters page (see chapter 3), the Rnnh routine uses that value to calculate the area of each grid cell and, in turn, the grid-specific threshold distance. If the user does not define the area on the measurement parameters
page, the routine calculates the total area from the minimum and maximum X/Y values (the bounding rectangle) and uses that value to calculate the area of each grid cell and, in turn, the grid-specific threshold distance. In either case, the routine will be able to calculate a threshold distance for each grid cell and run the routine.

However, if the area units are defined incorrectly on the measurement parameters page, then the routine will certainly calculate the grid cell-specific threshold distances wrongly. For example, if data are in feet but the area on the measurement parameters page are defined in square miles, most likely the routine will not find any points that are farther apart than any of the grid cell threshold distances since each distance will be defined in miles. In other words, it is essential that the area units be consistent with the data for the routine to properly work.

Use kernel bandwidths that produce stable estimates

Another concern is that the bandwidth for the baseline variable be defined as to produce a stable density estimate of the variable. Be careful about choosing a very small bandwidth. This could have the effect of creating clusters at the edges of the study area or very large clusters in low population density areas. For example, in low population density areas, there will probably be fewer persons or events than in more built-up areas. This will have the effect on the Rnnh calculation of producing a very large matching distance. Points that are quite far apart could be artificially grouped together, producing a very large cluster. Using a larger bandwidth will produce a more stable average.

Example 2: Simulated Rnnh Clustering

To illustrate the logic of the Rnnh routine, a simulated example is presented. Twenty-seven points were assigned to three groups in the Baltimore metropolitan region (Figure 6.11). The 27 points were grouped in a similar pattern, but one was placed in the center of the metropolitan region (near downtown Baltimore) while the other two were placed in less populated areas. The Nnh and Rnnh routines were compared with these data. One would expect the Nnh routine to cluster the 27 points into three groups whereas the Rnnh routine should cluster only 18 of the points into two groups. The reason for the lack of a third group is that one would expect a high number of incidents in the center; consequently, it is not high relative to the underlying baseline population. Figures 6.12 and 6.13 show exactly this solution.

In other words, the Nnh routine clusters points together irrespective of the distribution of the baseline population whereas the Rnnh routine clusters points together relative to the baseline population.

Rnnh Output Files

The output files are similar to the Nnh routine. The Rnnh routine has three outputs. First, final seed locations of each cluster and the parameters of the selected standard deviational ellipse are calculated for each cluster. These can be output to a '.dbf'
Figure 6.11:
Incidents in Relation to Population: Baltimore Region 1990

Incident Location and Persons Per Square Mile

Person Per Square Mile
- 5,000 or less
- 5,001-10,000
- 10,001-15,000
- 15,001-20,000
- 20,001-25,000
- 25,001-30,000
- 30,001-35,000
- 35,001-40,000
- 40,001-45,000
- 45,001-50,000
- 50,001-55,000
- 55,001 or more
Figure 6.12:
Nearest Neighbor Clustering of Incidents
Nnh Clusters and Incident Locations
Figure 6.13:
Risk-Adjusted Nearest Neighbor Clustering of Incidents Relative to Population

Rnnh Clusters and Incident Locations

Person Per Square Mile

- 5,000 or less
- 5,001-10,000
- 10,001-15,000
- 15,001-20,000
- 20,001-25,000
- 25,001-30,000
- 30,001-35,000
- 35,001-40,000
- 40,001-45,000
- 45,001-50,000
- 50,001-55,000
- 55,001 or more

Miles

0 2 4
file or saved as a text (‘.txt’) file. Only 45 of the seed locations are displayed on the screen. The user can scroll down or across by adjusting the horizontal and vertical slider bars and clicking on the Go button.

Second, for each order that is calculated, CrimeStat calculates the mean center of the cluster. This can be saved as a ‘.dbf’ file. Third, either standard deviational ellipses or convex hulls of the clusters can be saved in ArcView ‘.shp’, MapInfo ‘.mil’ or Atlas*GIS ‘.bna’ formats. Again, the convex hulls display polygons around the incidents whereas the ellipses are determined by the number of standard deviations to be calculated (see above). In general, use a 1X standard deviational ellipse since 1.5X or 2X standard deviations can create an exaggerated view of the underlying cluster. On the other hand, for a regional view, a one standard deviational ellipse may not be very visible. The user has to balance the need to accurately display the cluster compared to making it easier for a viewer to understand its location.

Because there are also orders of clusters (i.e., first-order, second-order, etc.), there is a naming convention that distinguishes the order.

For the ellipses, the convention is

\[ R_{nnh}^O<\text{username} > \]

where \( O \) is the order number and \( \text{username} \) is a name provide by the user. Thus,

- \( R_{nnh}^1\text{robbery} \)
  
  are the first-order clusters for a file called ‘robbery’ and

- \( R_{nnh}^2\text{burglary} \)
  
  are the second-order clusters for a file called ‘burglary’. Within files, clusters are named

\[ R_{nnh}^O\text{Ell}^N<\text{username} > \]

where \( O \) is the order number, \( N \) is the cluster number and \( \text{username} \) is the user-defined name of the file. Thus,

- \( R_{nnh}^1\text{Ell}^{10}\text{robbery} \)
  
  is the tenth cluster within the first-order clusters for the file ‘robbery’ while

- \( R_{nnh}^2\text{Ell}^1\text{burglary} \)
  
  is the first cluster within the second-order clusters for the file ‘burglary’.

For the convex hulls, the cluster numbers are the same as the ellipses but the prefix name is output with a ‘CRNNH1’ prefix for the first-order clusters, a ‘CRNNH2’ prefix for
the second-order clusters, and a ‘CRNNH3’ prefix for the third-order clusters. Higher-order clusters will index only the number.

Example 3: Rnnh Clustering of Vehicle Thefts

A second example is the clustering of 1996 Baltimore vehicle thefts relative to the 1990 population of census block groups. The test is for clusters of vehicle thefts that are more concentrated than would be expected on the basis of the population distribution. Using the default threshold probabilities and a minimum sample size per cluster of 25, the Rnnh routine identified five first-order and one second-order cluster (Figure 6.14); the incidents are not shown. As seen, there are only five clusters, most of which are peripheral to the downtown area.

Compare this distribution with the results of the Nnh on the same data, using the same parameters (Figure 6.15). The Nnh found 28 first-order clusters and two second-order clusters. As expected, they are more concentrated in the center. Note that there are far fewer clusters identified in the Rnnh routine than in the Nnh. Many of the clusters in the Nnh routine are due to a higher concentration of population. Once this is normalized, one finds that there are only a few areas of very high risk for vehicle theft. In other words, the Rnnh routine identifies areas of high risk for vehicle theft whereas the Nnh routine identifies areas of high volume for vehicle theft.

Simulating Statistical Significance

Because the sampling distribution of the clustering method is not known, the Rnnh routine allows Monte Carlo simulations to approximate confidence intervals, similar to the Nnh routine (Dwass, 1957; Barnard, 1963). The output is identical to the Nnh routine. Essentially, it produces approximate confidence intervals for the number of first-order clusters, the area of clusters, the number of points in each cluster, and the density of each cluster. Second- and higher-order clusters are not simulated since their structure depends on the first-order clusters. The user can see whether the first-order cluster structure is different than that which is produced by a random distribution. See the notes above under Nnh for more details. Table 6.4 shows the output for 1996 Baltimore County robberies with the default search threshold and a minimum sample size of 20 incidents.

The results also show those obtained from 1000 Monte Carlo simulations. There were seven first-order clusters and one second-order cluster. Looking at the Monte Carlo simulations, the two most critical parameters are the number of first-order clusters found and the density of the clusters. In the simulation, the minimum number of clusters found under these conditions (i.e., with the default threshold distance and a minimum sample size of 20 incidents) was one while the maximum number was five. The 95th percentile was four incidents. Since the Rnnh routine produced seven first-order clusters, the routine has identified more clusters than would normally be expected on the basis of chance. Looking at the density estimates from the simulation, the maximum density was 11.913282 and the 95th percentile was 7.365212. Since all seven first-order clusters had densities higher than
Figure 6.14: 
1996 Metropolitan Baltimore Vehicle Theft
Risk-Adjusted Nearest Neighbor Clusters
Figure 6.15:
1996 Metropolitan Baltimore Vehicle Theft
Nearest Neighbor Clusters
Table 6.4
Risk-adjusted Clustering of Baltimore County Robberies: 1996

Risk-Adjusted Nearest Neighbor Hierarchical Clustering:

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>1181</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Likelihood of grouping pair of points by chance</td>
<td>0.50000 (50.000%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z-value for confidence</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Measurement type</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output units</td>
<td>Miles, Squared Miles, Points per Squared Miles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clusters found</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation runs</td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Displaying 8 ellipse(s) starting from 1

<table>
<thead>
<tr>
<th>Order</th>
<th>Cluster</th>
<th>Mean X</th>
<th>Mean Y</th>
<th>Rotation</th>
<th>X-Axis</th>
<th>Y-Axis</th>
<th>Area</th>
<th>Points</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-76.44973</td>
<td>39.31523</td>
<td>73.89169</td>
<td>0.19429</td>
<td>0.09230</td>
<td>0.05634</td>
<td>31</td>
<td>550.251866</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-76.60194</td>
<td>39.40076</td>
<td>4.40641</td>
<td>0.12272</td>
<td>0.12929</td>
<td>0.04984</td>
<td>23</td>
<td>461.446220</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-76.78279</td>
<td>39.36184</td>
<td>62.61813</td>
<td>0.24605</td>
<td>0.15511</td>
<td>0.11990</td>
<td>26</td>
<td>216.852324</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-76.73157</td>
<td>39.34387</td>
<td>4.30498</td>
<td>0.08916</td>
<td>0.07321</td>
<td>0.02051</td>
<td>24</td>
<td>1170.341418</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-76.44539</td>
<td>39.30523</td>
<td>13.63299</td>
<td>0.19639</td>
<td>0.11154</td>
<td>0.06882</td>
<td>20</td>
<td>290.622622</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>-76.75368</td>
<td>39.31132</td>
<td>89.56994</td>
<td>0.19748</td>
<td>0.22914</td>
<td>0.14216</td>
<td>22</td>
<td>154.753006</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>-76.73132</td>
<td>39.28897</td>
<td>11.83419</td>
<td>0.09359</td>
<td>0.18312</td>
<td>0.05384</td>
<td>21</td>
<td>390.033756</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>-76.74984</td>
<td>39.32650</td>
<td>66.40941</td>
<td>4.19556</td>
<td>1.63703</td>
<td>21.57723</td>
<td>4</td>
<td>0.185381</td>
</tr>
</tbody>
</table>

Distribution of the number of clusters found in simulation (percentile):

<table>
<thead>
<tr>
<th>Percentile</th>
<th>Clusters</th>
<th>Area</th>
<th>Points</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>1</td>
<td>1.67880</td>
<td>20</td>
<td>1.648432</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
<td>2.36257</td>
<td>20</td>
<td>1.874836</td>
</tr>
<tr>
<td>1.0</td>
<td>1</td>
<td>2.51219</td>
<td>20</td>
<td>1.996056</td>
</tr>
<tr>
<td>2.5</td>
<td>1</td>
<td>2.67031</td>
<td>20</td>
<td>2.208136</td>
</tr>
<tr>
<td>5.0</td>
<td>1</td>
<td>2.98150</td>
<td>20</td>
<td>2.372246</td>
</tr>
<tr>
<td>95.0</td>
<td>4</td>
<td>13.57660</td>
<td>50</td>
<td>7.365212</td>
</tr>
<tr>
<td>97.5</td>
<td>4</td>
<td>13.95390</td>
<td>53</td>
<td>7.932653</td>
</tr>
<tr>
<td>99.0</td>
<td>5</td>
<td>14.34076</td>
<td>56</td>
<td>8.643887</td>
</tr>
<tr>
<td>99.5</td>
<td>5</td>
<td>14.60388</td>
<td>58</td>
<td>9.595312</td>
</tr>
<tr>
<td>max</td>
<td>5</td>
<td>15.41259</td>
<td>67</td>
<td>11.913282</td>
</tr>
</tbody>
</table>

the 95th percentile, the density of these clusters is greater than what would normally be expected on the basis of chance. In other words, the routine has identified more clusters and higher density clusters than would be expected on the basis of chance.

Guidelines for Selecting Parameters

The guidelines for selecting parameters in the Rnnh routine are similar to the Nnh except the user must also model the baseline variable using a kernel density interpolation. The process is a little like tuning a shortwave radio, adjusting the dial until the signal is
detected. We suggest that the user first develop a good density model for the baseline variable (see Chapter 8). The user has to develop a trade-off between identify areas of high and low population concentration to produce an estimate that is statistical reliable (stable).

There are two types of ‘fine tuning’ that have to go on. First, the ‘background’ variation has to be tuned (the baseline ‘at risk’ variable). This is done through the kernel density interpolation. If too narrow a bandwidth is selected, the density surface will have numerous undulations with small ‘peaks’ and ‘valleys’; this could produce unreal and unstable risk estimates. A grid cell with a very small density value could produce an extremely large threshold distance whereas a grid cell with a very low density could produce an extremely small threshold distance. Conversely, if too large a bandwidth is selected, the density surface will not differentiate very well and each grid cell will have, more or less, the same threshold distance. In this case, the Rnnh routine would yield a result not very different from the Nnh routine.

Second, there is tuning of the clusters themselves through the threshold adjustment and minimum size criteria. If a large threshold probability is selected, too many incidents may be grouped; conversely, if a small threshold probability is selected, the result may be too restrictive. Similarly, if a small minimum sample size for clusters is used, there could be too many clusters whereas the opposite will happen if a large minimum sample size is chosen (i.e., zero clusters). The user must experiment with both these types of adjustment to produce a sensible cluster solution that captures the areas of high risk, but no more.

Limitations of the Technique

There are some technical limitations that the Rnnh routine shares with the Nnh routine. First, the method only clusters incidents (points); a weighting or intensity variable will have no effect. Second, the size of the grouping area is dependent on the sample size if the confidence interval around the mean random distance is used as the threshold distance criteria. However, since the threshold distance is adjusted dynamically, this has less effect than in the Nnh since it is now a relative comparison rather than an absolute distance.

Third, there is arbitrariness in the technique due to the minimum points rule. Different users could define the minimum differently, which could lead to different conclusions about the location of high risk clusters. Finally, unique to the Rnnh, the method requires both an incident file (the primary file) and a baseline file (the secondary file. It cannot work on calculated rates (e.g., incidents per capita by zones). For the latter, the user should look at techniques such as the SatScan method (Kulldorff, 1997).

Nevertheless, the Rnnh routine is a useful technique for identifying clusters that are more concentrated than would be expected on the basis of the population distribution.
Risk Adjusted Nearest Neighbor Hierarchical Clustering of Tuberculosis Cases in Harris County, Texas: 1995 to 1998

Matthew L. Stone, MPH
Center for Health Policy Studies
University of Texas-Houston, School of Public Health-Houston, Texas

Data was collected from an ongoing, population-based, active surveillance and molecular epidemiology study of tuberculosis cases reported to the City of Houston Tuberculosis Control Office from October 1995 to September 1998. During this time, 1774 cases of tuberculosis were reported and 1480 of those who participated in this study were successfully geocoded.

CrimeStat was used to make an initial survey of potential hot spot areas of tuberculosis cases where more focused TB control efforts could be implemented. Given a .05 level of significance for grouping a pair of points by chance and a minimum of five cases per cluster, 24 first-order clusters and one second-order cluster were detected after adjusting for the underlying population. Most first-order clusters were detected in the center of Harris County, including the metropolitan downtown area. By adjusting for the underlying population, the clusters identify areas with higher than average TB incidence. Some of these clusters are homeless shelters as many homeless persons are particularly prone to TB.
Using Risk Adjusted Nearest Neighbor Hierarchical Clustering to Compare Actual and Media Hotspots of Homicide

Derek J. Paulsen
Department of Criminal Justice and Police Studies
Eastern Kentucky University

*Crimestat* offers an excellent method for determining risk adjusted hot spots of crime incidents within a jurisdiction. Risk-adjusted nearest neighbor hierarchical spatial clustering (Rnnh) is a spatial clustering routine that groups points together based on both proximity to other points and the distribution of a baseline variable. In this example two different Rnnh analyses were conducted and compared for homicides in Houston, Texas. The first involves homicide incident locations adjusted for the population of each census tract, while the second involves incidents that were covered in the newspaper adjusted for the homicide rate of each census tract. The purpose of this analysis is to determine if there are differences in the spatial clustering of actual homicide incidents and those that are covered in the newspaper.

The preferences for the analysis were the same for both Rnnh analyses. For the primary file (homicide incidents & incidents covered in the newspaper) the pair probability search radius was set at .01, with a minimum of 10 points per cluster. For the secondary file (population & homicide rate), a quartic kernel density interpolation was used with an adaptive bandwidth and a minimum sample size of 100. Importantly, the analysis showed that media hot spots and actual hot spots do not coincide. Media coverage showed homicides to be concentrated in different areas than they are actually concentrated.

Actual Homicide Hot Spots vs. Media Coverage Hot Spots in Houston Texas
1. The output in table 6.1 has been formatted. CrimeStat only outputs an Ascii file. In this case, the Ascii file was pasted into Word Perfect®, the word processing program used for this manual, and was then formatted so that the underscore was consistent with the title words and the columns lined up.

2. In the statistical literature, this type of statistic is known as a spatial scan with a fixed circular window (Kulldorff, 1997; Kulldorff and Nagarwalla, 1995). However, our emphasis here is on defining approximate point locations where there is either measurement error or very small locational differences. In this sense, the term ‘fuzzy’ is more similar to the classification literature where imprecise boundaries exist and an incident can belong to two or more groups (Bezdek, 1981; McBratney and deGruijiter, 1992; Xie and Beni, 1991).

3. This is the next highest degree of freedom in the Student’s t-table below infinity.

4. The particular steps are as follows:
   
   A. All distances between pairs of points are calculated, using either direct or indirect distance as defined on the measurements parameters page. The matrix is assumed to be symmetrical, that is the distance between A and B is assumed to be identical to the distance between B and A.
   
   B. The mean expected random distance is calculated using formula 5.2 and the threshold distance (the confidence interval for the corresponding t) is calculated using formulas 6.2 and 6.3 depending on whether it is a lower or upper confidence interval. The particular interval is selected by user on the slide bar.
   
   C. All distance pairs smaller than the threshold distance are selected for clustering.
   
   D. For each incident point, the number of distances to other points that are smaller than the threshold distance are counted and placed in a reduced matrix. Any incident point which does not have another point within the threshold distance is not clustered. Any distance that is greater than the threshold distance is not considered for clustering.
   
   E. All points in the reduced matrix are sorted in descending order of the number of distances to other points shorter than the threshold distance, and the incident point with the largest number of below threshold distances is selected for the initial seed of the first cluster.
   
   F. All other incidents that are within the threshold distance of the initial seed point are selected for cluster 1.
The number of points within the cluster are counted. If the number is equal
to or greater than the minimum specified, then the cluster is kept. If the
number is less than the minimum specified, then the cluster is dropped.

For those clusters that are kept, the center of minimum distance is calculated
for each to identify the cluster center.

The clustered points are removed from further clustering.

Of the remaining points, the incident point with the largest number of
distances to other points shorter than the threshold distance is selected for
the initial seed the second cluster.

All other points which are within the threshold distance of the first cluster
seed point are selected for cluster 2.

The mean center of these selected points is calculated to identify the cluster
center.

These points are removed from further clustering.

Steps J through M are repeated for all remaining points in the reduced
matrix until no more points are remaining in the reduced matrix or until
there are fewer than the specified minimum number of points for those
remaining in the reduced matrix.

The steps are as follows:

Using the same p-values selected in the first-order, the mean random
expected distance is calculated. However, the sample size is the number of
first-order clusters identified, not the original number of points. Thus, the
threshold distance is calculated by

\[
\text{Confidence Interval for Second-order Mean Random Distance} = 0.5 \sqrt{\frac{A}{M}} +/\ t \left[ \frac{0.26136}{\sqrt{M^2/A}} \right]
\]

where A is the area of the region and M is the number of first-order clusters
identified during first-order clustering (i.e., not N). Thus, there is a different
threshold distance for the second-order clustering. The t-value specified in
the first-order clustering is maintained for second- and higher-order
clustering.

All distances between first-order cluster centers are calculated and only those
that are smaller than the second-order threshold distance are selected for
second-order clustering.

C. If there are no distances between first-order cluster centers that are smaller than the second-order threshold distance, then the clustering process ends.

D. If there are distances between first-order cluster centers that are smaller than the second-order threshold distance, then the steps specified in endnote 3 are repeated to produce second-order clusters. A minimum of four first-order clusters is required to allow a second- or higher-order cluster.

E. If there are second-order clusters, then this process is repeated to either extract third-order clusters or to end the clustering process if no distances between second-order cluster centers are smaller than the (new) third-order threshold distance or if there are fewer than four new seeds in the cluster.

F. The process is repeated until no further clustering can be conducted, either all sub-clusters converge into a single cluster or the threshold distance criteria fails or there are fewer than four seeds in the higher-order cluster.

6. It is not an exact risk test since we are comparing 1996 vehicle thefts with 1990 population. It is an approximate risk test.