Geoinformatica (2011) 15:49–74
DOI 10.1007/s10707-010-0116-1

Crime analysis through spatial areal aggregated density patterns

Peter Phillips · Ickjai Lee

Received: 22 July 2008 / Revised: 31 May 2009 /
Accepted: 14 September 2010 / Published online: 3 October 2010

Abstract Intelligent crime analysis allows for a greater understanding of the dynamics of unlawful activities, providing possible answers to where, when and why certain crimes are likely to happen. We propose to model density change among spatial regions using a density tracing based approach that enables reasoning about large areal aggregated crime datasets. We discover patterns among datasets by finding those crime and spatial features that exhibit similar spatial distributions by measuring the dissimilarity of their density traces. The proposed system incorporates both localized clusters (through the use of context sensitive weighting and clustering) and the global distribution trend. Experimental results validate and demonstrate the robustness of our approach.

Keywords Crime analysis · Spatial distribution · Density tracing · Areal aggregated data

1 Introduction

Crime analysis allows for a greater understanding of the dynamics of unlawful activities, providing possible answers to where, when and why certain crimes are likely to happen. This analysis is of great importance to a number of people and agencies such as regional planners, politicians, police and residents themselves.

The distribution of crime in time and space is non-random. Because criminal behavior is dependent upon situational factors, crime is patterned according to the

P. Phillips (✉) · I. Lee
School of Business, Discipline of IT, James Cook University, Townsville, Australia
e-mail: peter.phillips@jcu.edu.au

I. Lee
e-mail: ickjai.lee@jcu.edu.au

location of criminogenic environments. Crime will be concentrated around crime opportunities and other environmental features that facilitate criminal activity. The purpose of crime analysis is to identify and describe these crime patterns [33].

Environmental criminology is a branch of criminological theory that can guide crime analysis and crime prevention efforts. The goal of environmental criminology is to understand the various aspects of a criminal event in order to identify patterns of behaviors and environmental factors that create opportunities for crime [3]. Discovering crime and spatial features that exhibit a similar spatial distribution (co-distribution) is a key component to environmental criminology and allows a deeper insight into the complex nature of criminal behavior.

As crime activities are geospatial phenomena, they must be interpreted and analyzed in conjunction with various factors that can contribute to the formulation of crime. Many of these datasets, such as those provided by the Queensland Police Service, are areal aggregated due to limited environmental circumstances and ethical issues. Areal aggregated datasets are region based datasets that have aggregate data values (densities) for regions, e.g. a particular suburb has recorded five assaults. The Australian Bureau of Statistics also releases sociodemographic information only in aggregated form to protect the privacy of individuals. It is necessary for crime analysis tools to discover co-patterning, that is, patterns happening together in multiple themes or datasets. These co-patterning relationships can comprise of point-to-point association (co-location capturing the relationship within a pair of points belonging to different geographical themes at each location), spatial dependence (capturing the relationship between distinct pairs belonging to the same geographical theme), and spatial co-distribution (modeling the relationship between pairs belonging to different geographical themes across locations) [17]. Pearson's correlation coefficient is a typical measure for point-to-point association while Moran's I is a typical measure for spatial dependence. There is no widely accepted measure for spatial co-distribution.

Several crime data mining techniques have been developed over recent years [5, 12, 20, 24], however reasoning about crime data has received less attention [4, 21]. Most of these reasoning approaches are based on clustering *point crime data* and reasoning with those clusters. The drawback of such approaches is that reasoning based on surrogate clusters can be imprecise and is heavily cluster-dependent. These techniques can also only discover positive associative features whereas negative associative features can also be informative. Several works using spatial Association Rules Mining (ARM) have also been proposed in order to mine spatial associations in geospatial databases [15, 16, 29]. The main drawback of these approaches is that they capture point-to-point association (with no consideration for neighboring regions), but ignore spatial dependence.

The increasing availability of heterogeneous data, such as socio-economic and socio-demographic factors, geospatial features and crime datasets, has brought with it an increasing need for intelligent analysis. To discover interesting patterns in these areal aggregated datasets the co-patterning relationship between different spatial themes across locations needs to be modeled and quantified.

We propose to capture these co-patterning relationships by modeling the spatial distribution of areal aggregated datasets using a density trace. Unlike point-to-point association methods, we are able to account for local neighborhood information (capturing nearby neighbors) and the global distribution (modeling the whole study

region) within the density trace. Figure 1 shows three hexagon shaped areal aggregated datasets (shaded regions indicate density: black = 3, grey = 2, white = 1). The three pairs of datasets, A–B, B–C and C–A show identical point-to-point association as Pearson's correlation coefficient is the same (0.422)(also each dataset has 5 black, 17 grey and 13 white regions). However, it can be seen that the pair A–B shows a higher level of spatial co-patterning than those of B–C and C–A. By modeling the spatial co-distribution, we are able to account for this spatial neighborhood information. We are able to discover co-patterning relationships that can be either positive (causing crimes) or negative (preventing crimes). The resulting patterns can then be used by domain specialists to further investigate and target the cause of these specific patterns.

How we model the density trace is important as we must retain as much spatial information as possible. We examine four popular locational ordering methods to determine the spatial ordering of the areal units in the study region: Guided Local Search (GLS), Depth First Search (DFS), Breadth First Search (BFS) and a Nearest Neighbor (NN) technique. We show that in general, the GLS technique is best able to capture neighbors (i.e. those regions that share a border). The distance (dissimilarity) between two density traces is calculated using a modified *Locality In-between Polylines (LIP)* distance measure [23]. Intuitively, two density traces are considered spatially similar when they move close (i.e., their traces approximate each other) at the same place. To the best of our knowledge, this is the first attempt at using density traces/routes to represent the spatial distribution of areal aggregated datasets for crime data mining. Our density tracing approach efficiently and robustly reveals the top-*k* positive and negative co-distribution relationships. Density tracing observes the first law of Geography [30] and can consider not only localized clusters but the global trend (density trace).

The next section provides a review of existing areal aggregated crime reasoning techniques. In Section 3 we detail our proposed technique and the special properties

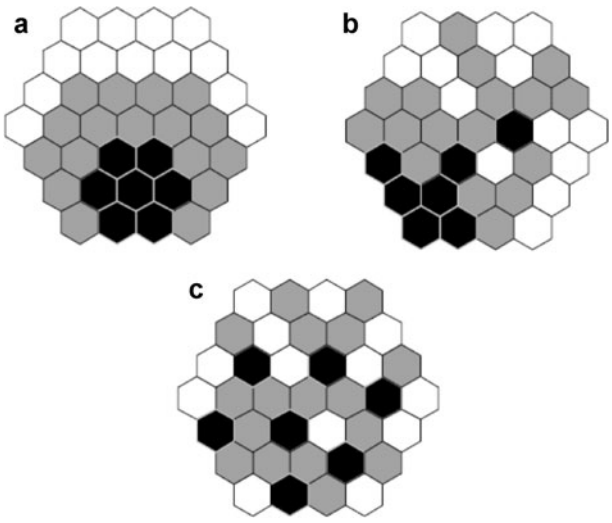


Fig. 1 Drawback of point-to-point association co-patterning: **a**–**c** Datasets *A*–*C*

of areal aggregated data and density tracing. Section 4 provides experimental evaluation and comparison with other approaches. We conclude with final remarks and ideas for future work in Section 5.

2 Reasoning about areal aggregated crime

There exist three general approaches for reasoning with areal aggregated datasets: choropleth mapping (visualization), spatial statistics, and geographical data mining.

Choropleth mapping is a common technique for representing aggregated data in data-poor environments. In crime analysis, this is generally known as crime mapping and is used by analysts to visually search for trends or patterns of a particular crime type in specific areas [20, 25]. The problem faced when using this mapping is the choice of data classification method: the user must select an appropriate technique that best depicts the spatial properties of the data [8]. Choosing the 'best' classification method is heavily dependent on the user's domain knowledge and even then may require a number of iterations to determine the most suitable method. Visualization techniques are not a scalable solution to the crime reasoning problem and further, reasoning is based on subjective visual inspection.

Geographic Data Mining (GDM) is data mining applied to georeferenced datasets. GDM must consider the peculiar characteristics of geoinformation that makes space special in order to detect geographically interesting patterns [19]. Crime activities are geospatial phenomena and as such techniques for their analysis must take into account the special properties of geospatial data such as spatial autocorrelation and spatial heterogeneity. Two core techniques within GDM are spatial clustering and association mining. Spatial clustering is closely related to intensity measurement while association mining is generally related to dependency measurement.

Spatial association is the degree to which a set of observations are similarly arranged over space. Two different communities have focused on two different measures to model these spatial relationships. The geoinformatics community have focused on using spatial statistic-based approaches to measure spatial dependence (autocorrelation) based on cross-k function with Monte Carlo simulation, spatial chi-square tests, Moran's I and Geary's c statistics [2, 6]. These measures quantify the relationship between distinct observations belonging to the same theme across locations, that is, they measure the spatial relationship a variable has with itself. For example, they can model the likelihood that theft will occur near other regions where theft is high. However, these measures are limited to univariate measurement and are computationally expensive and as such are not suited to data-rich environments [19].

The spatial data mining community has focused on variants [14–16, 18] of traditional ARM [1] to model spatial association. These approaches typically capture point-to-point associations and mainly focus on frequent patterns, and can thus be dominated by uninteresting frequent co-occurring patterns, such as traffic lights are co-located with roads. Approaches that overcome this drawback by mining rare events have also been developed [13], however they are still focused on capturing point-to-point association. These algorithms are scalable and applicable to mining co-patterning relationships, however, the main criticism of these approaches lies in their inability to model spatial dependence. Lee [17] explored the combination of

point-to-point association and spatial dependence, but it is still limited to bivariate analysis and remains computationally expensive requiring quadratic time for bivariate analysis.

Most existing GDM approaches for reasoning about areal aggregated datasets use clustering, Association Rules Mining (ARM) or a combination of the two [9, 14–16]. Estivill-Castro and Lee [9] enable exploratory analysis of geospatial patterns by utilizing a clustering technique and then reason based on those clusters using ARM. Spatial clustering methods are often based on point data, so before they can be applied to areal aggregated datasets data transformation must take place. This transformation from an areal dataset to a point dataset may introduce artificial patterns depending on the technique used. Clusters are dense spatial aggregations and as such any patterns based on these clusters may not depict the trend of the data; only the trend of the dense clusters [10].

Recent geospatial reasoning techniques have also used Co-Location Rules Mining [34] to discover point features that are frequently located together in a geographic space. It extends traditional ARM by providing a transaction free approach using the concept of neighborhoods without having to define a reference feature. Typically, traditional ARM interest measures are not used and this approach introduces two new interest measures (*prevalence* and *conditional probability*) that can be used in a dynamic situation where transactions are not fixed to a constant. However, as with ARM an overwhelming amount of uninteresting patterns are typically discovered.

To successfully enable crime analysis and decision making using areal aggregated datasets, patterns that happen together in multiple themes or datasets must be discovered. We propose to discover these co-patterning relationships by modeling the spatial distribution of datasets as density traces. Current techniques that discover co-patterning relationships by modeling point-to-point association do not take into consideration neighborhood information, we overcome this draw back by utilizing both local neighborhood information and the global distribution within the density trace. We compare and contrast our approach to ARM in Section 4.

3 Density tracing for crime reasoning

3.1 Problem statement and motivation

The major drawback of current reasoning techniques that are based on clustering and ARM is that they may miss important global trends that are not part of the local cluster aggregations [10]. A vast number of spatial clustering methods have been proposed [11], with each method likely to give a different set of clusters, any patterns based on those clusters are heavily cluster-dependent. An example of this is given in Fig. 2, where we have three datasets that show the center points and density values of each region.

If we wish to find possible associative patterns for $dataset_a$, a typical cluster based reasoning approach would determine that $dataset_b$ is highly correlated with $dataset_a$ while $dataset_c$ is not. Depending on the chosen clustering method, the algorithm may only detect the densest regions of the three datasets (shaded regions), and in this case $dataset_a$ and $dataset_b$ have the same dense region.

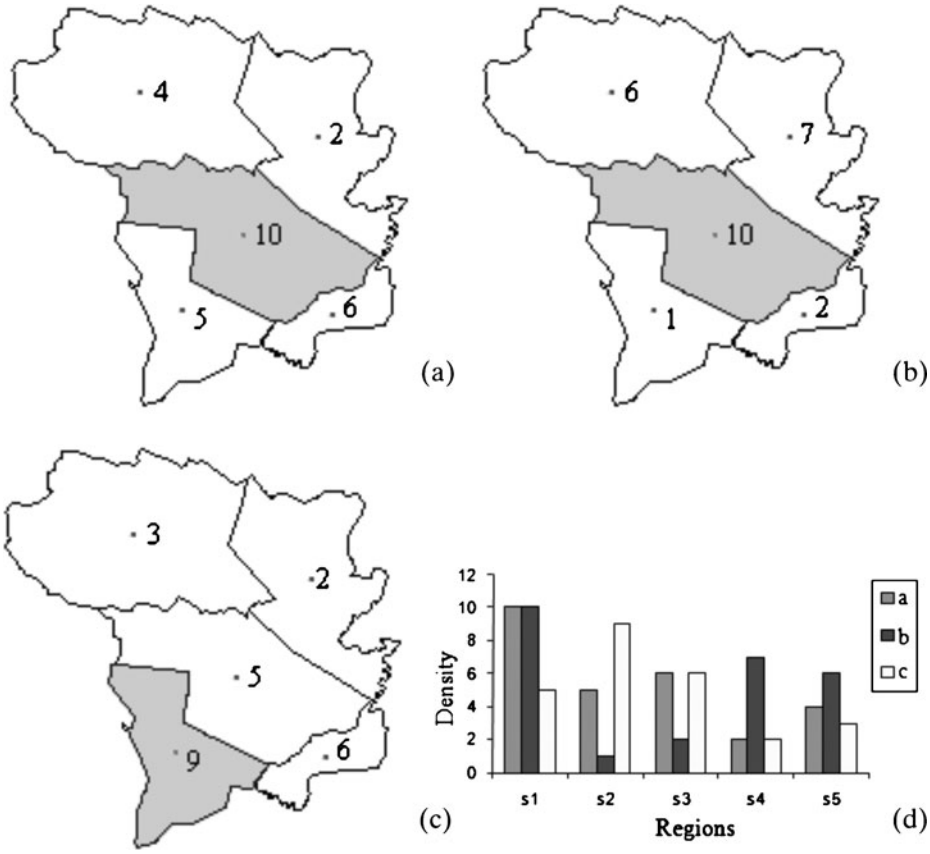


Fig. 2 Drawback of cluster based reasoning: **a–c** Datasets $dataset_{a-c}$; **d** Histogram of density values

From visual analysis of the histogram in Fig. 2d we argue that $dataset_c$ has a higher similarity than $dataset_b$ because the global density trend is more similar to that of $dataset_a$. The clustering approach fails to detect the global trend. Our framework returns the following results with these datasets demonstrating that $dataset_c$ has a higher similarity than $dataset_b$:

```
Reference Feature selected: dataset_a
Feature: dataset_b Dissimilarity: 0.4269240
Feature: dataset_c Dissimilarity: 0.0101308
```

We propose to model the spatial distribution by using the density change between regions to discover co-patterning relationships between certain types of crime. Crime and spatial features that exhibit similar spatial distributions warrant further investigation by domain experts. Our approach is able to overcome the drawbacks of frequent pattern mining approaches that are combined with clustering techniques by considering not only localized clusters but also the global spatial distribution.

3.2 Working principle

We explain the working principle of our framework with the example synthetic datasets shown in Fig. 3. We have three datasets that show the center points and density values of each region. We use the spatial distribution of density values to model the co-patterning relationship between regions belonging to different datasets across the study region. $Dataset_a$ and $dataset_b$ show a similar density distribution: higher than normal density in the shaded regions. $Dataset_c$ however has a more even density pattern with all regions showing a similar density.

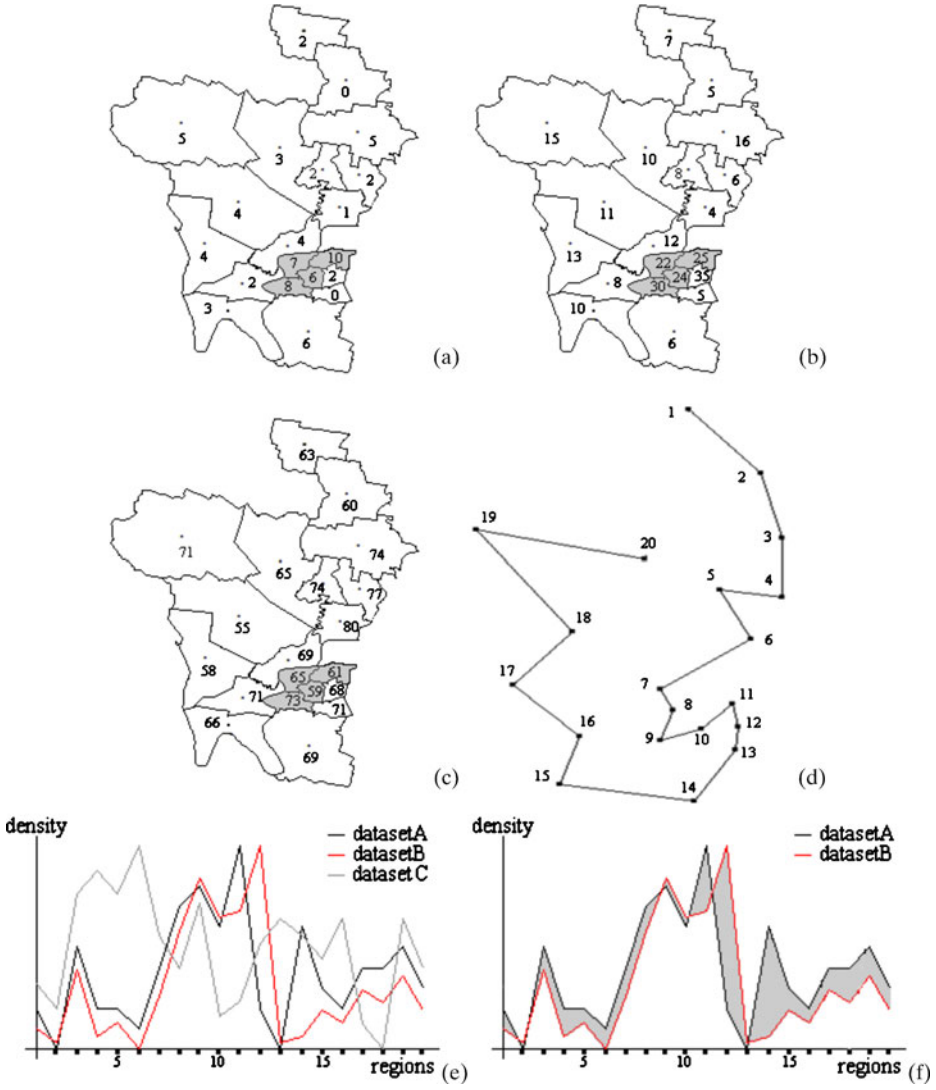


Fig. 3 Working principle of our framework: **a**–**c** Datasets $dataset_{a-c}$ before normalization; **d** GLS region ordering; **e** Density trace; **f** Density trace with area highlighted

We will now present an overview of how our framework discovers co-patterning relationships, details of each step will be given in Section 3.3. Initially, we load the three areal aggregated datasets $dataset_{a-c}$ and calculate the center point of each region. We assume all datasets share a common base map, thus the center points of each region only need to be calculated once. The density values are then normalized into the range [0, 1]. The next step is to determine the spatial ordering of regions. In this example we use the GLS method with a random starting location, Fig. 3d shows the resulting order with the starting region labeled 1. Once the spatial ordering of regions is determined, the density traces for each dataset can be calculated. Figure 3e depicts these density traces with each line representing one dataset. These density traces represent the spatial distribution of the density values projected onto the Cartesian plane. In this example we wish to find patterns involving $dataset_a$, thus we select $dataset_a$ as our reference feature f. The similarity value between f and each dataset is then computed using our modified *Locality In-between Polylines* technique, with the basic idea being to calculate the area of the shape formed by the two 2D density lines. Figure 3f highlights the areas that need to be computed to calculate the similarly between $dataset_a$ and $dataset_b$. We can prune the results with a user supplied minimum similarity *min_sim* or simply retrieve the *k Most Similar* and/or *k Least Similar* results. The output using this example dataset is as follows:

```
Reference Feature selected: dataset_a
Feature: dataset_b Dissimilarity: 0.735477
Feature: dataset_c Dissimilarity: 1.476240
```

3.3 Algorithm

Our approach is detailed in Algorithm 1. To calculate a density trace for our areal aggregated crime and feature datasets, they must first be preprocessed. Step 3 determines the center point of each region, which can then be used to determine the density trace and neighboring locations (we define neighbor as those that share a boundary). We assume all datasets share a common base map, thus the center points of each region only needs to be calculated once. The center of a region is determined by taking the center point of the region's bounding box. If the center does not fall inside the region itself the point is moved in the X direction until it enters into the region. The point is then moved in the same direction, along the X axis, until it exits the polygon. The centroid is calculated to be halfway between the two points, on the same X axis. For complex polygons that have more than one pair of polygon outlines that cross the X-axis, each pair of outlines is compared to see which pair creates the widest length along the X-axis. Then, the centroid is calculated to be halfway between the points where this pair of outlines crosses the X-axis. This is the same as the standard GIS technique employed by ArcView GIS for calculating the center point of a polygon [26].

Step 18 of the density tracing framework is to normalize the density values so that a meaningful similarity can be measured. We must normalize the datasets as the density traces are projected onto the Cartesian plane, with density as the Y axis. To measure similarity we compute the area formed between two density traces so to successfully compare between pairs of datasets the density values must be in a common range. We do this by using the *min-max normalization* [28] technique which

Algorithm 1 Density Tracing for Crime Reasoning

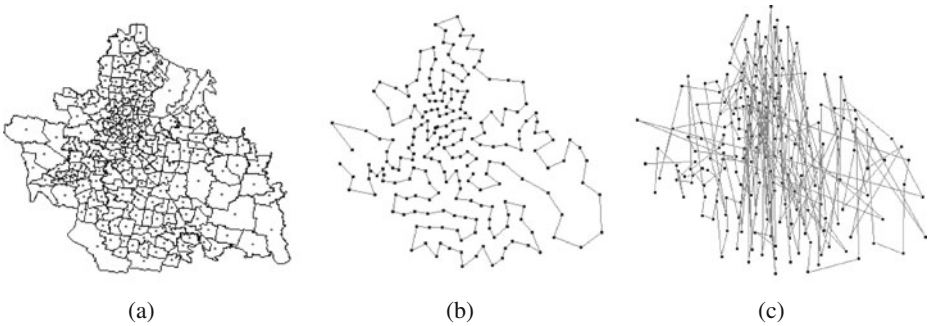
Input: A set $D = \{d_1, d_2, \ldots, d_n\}$ of spatial areal aggregated datasets, a common base map $B = \{b_1, b_2, \ldots, b_l\}$, the required spatial ordering algorithm *orderAlg*, a reference feature f, and a minimum similarity *minSimilarity*;
Output: A set $S = \{s_1, s_2, \ldots, s_m\}$ of ordered features with $dissimilarity_f < minSimilarity$;

```
LoadDatasets(D);

CalcCenters(B):
for each region r in B do
    cp ← centerPointOfBoundingBox(r)
    if cp is not within BoundingBox(r) then
        point1: Move cp in the X direction until it enters into r
        point2: Move cp in the X direction until it exits r
        Centroid cp is calculated to be halfway between point1 and point2 on the same X axis
        if More than one pair of polygon outlines cross the X-axis then
            Compare each pair of outlines to see which pair creates the widest length along the X-axis
            Calculate centroid cp as halfway between the points where this pair of outlines crosses the X-axis
        end if
    end if
    return cp
end for

NormalizeDensity(D):
for all d_i ∈ D do
    upperLimit ← 1
    lowerLimit ← 0
    Hi ← d_i.max()
    Lo ← d_i.min()
    fact ← (upperLimit − lowerLimit)/(Hi − Lo)
    for all x ∈ d_i do
        d_i[x] ← (d_i[x] − Lo) * fact + lowerLimit
    end for
end for

RegionOrdering(B, orderAlg);

DT ← DensityTrace(D):
We have to project the normalised density onto the Cartesian plane according to the region ordering
for all d_i ∈ D do
    for iterator r ← regionOrder.begin() do
        Get the corresponding density value (d1) for region r from d_i
        point(X, Y) ← p1((*r), d1)
        Get the next region r from regionOrder: r++
        if r! = RouteSet.end() AND r! = NULL then
            Get the corresponding density value (d2) for region r from d_i
            point(X, Y) ← p2((*r), d2)
            Make segment between p1, p2
            Store segment in list: DT
        end if
    end for
end for

for all d_i ∈ D do
    list similarity S
    localS ← CalcSimilarity(f, d_i, minSimilarity)
    if localS! = −1 then
        S.append(localS)
    end if
end for
S.sort()
return S
{Note: Library of Efficient Data types and Algorithms (LEDA) utilised to provide geometric algorithms and data types such as intersection of lines, area of polygons, etc.}
```

retains the original distribution of scores except for a scaling factor and transforms all the density values into the common range [0, 1]. Step 30 (detailed in Algorithm 2) of the algorithm is to determine the spatial ordering of the regions so that we can determine a density trace. This ordering can have a large impact on the overall framework: we must preserve the spatial information contained within the data so that the discovered patterns describe the spatial distribution. We are concerned with how density changes from location to location, from neighbor to neighbor, and our region ordering should reflect this. Again, as we assume all datasets share a common base map, this spatially aware region ordering needs to be calculated only once for all datasets.

We investigate four popular linear techniques that can be used to determine this spatially aware region ordering; greedy GLS, DFS, BFS and a NN technique. As we can see from Fig. 4, the GLS and NN-5 orders are better able to preserve spatial neighborhood information than both DFS and BFS. Visually, we can see that the GLS route is the only one that does not have overlapping edges (that is, edges that cross over another edge), also Fig. 4f shows that compared with GLS and NN-5, DFS and BFS exhibit a significantly larger edge length variance.

There are a number of different techniques that can be used to determine spatial neighbors and orders. Neighborhood graphs such as the Relative Neighborhood Graph (RNG), Gabriel Graph (GG) and the Delaunay Triangulation (DT) [22] cannot be used directly with our approach as they produce a nonlinear ordering. The density tracing approach outlined in this paper projects the normalized density



Order	Variance
GLS	0.000082
DFS	0.007476
BFS	0.006869
NN-5	0.000766

(f)

Fig. 4 Spatially aware region ordering: **a** Base map including center points; **b** GLS; **c** DFS; **d** BFS; **e** NN-5; **f** Variance of edge lengths

Algorithm 2 Density Tracing: RegionOrdering

Require: Centre point for each region in the base map has been calculated;
Input: A common base map B and the required spatial ordering algorithm *orderAlg*;
Output: A list of regions *order*, ordered by the spatial ordering algorithm *orderAlg*;

```
1: Starting region can either be random or user supplied; polygon p = randomRegion()
2: if orderAlg = GLS then
3:     Use Fast Local Search, the 2-Opt heuristic as the move operator, no construction heuristic and Alpha ←
       0.167
4:     Invoke GLS algorithm using a random starting solution
5:     Refer to Voudouris [31] for complete algorithm
6: else if orderAlg = DFS then
7:     list order
8:     stack < polygon > S
9:     S.push(p)
10:    while !S.empty() do
11:        polygon local ← S.pop()
12:        order.append(local)
13:        for Each neighbouring region n of local ordered by distance do
14:            if n is not already in order or S then
15:                S.push(n)
16:            end if
17:        end for
18:    end while
19:    return order
20: else if orderAlg = BFS then
21:    list order
22:    queue < polygon > Q
23:    Q.append(p)
24:    while !Q.empty() do
25:        polygon local ← Q.pop()
26:        order.append(local)
27:        for Each neighbouring region n of local ordered by distance do
28:            if n is not already in order or Q then
29:                Q.append(n)
30:            end if
31:        end for
32:    end while
33:    return order
34: else if orderAlg = NN then
35:    list order
36:    order.append(p)
37:    local base map lb ← B
38:    for all region r ← p ∈ lb do
39:        list polygon nnList ← nearestNeighbors(r, 5)
40:        for all region n ∈ nnList do
41:            if n is not already in order then
42:                order.append(n)
43:                lb.remove(n)
44:            end if
45:        end for
46:    end for
47:    return order
48: end if
```

onto the Cartesian plane following a defined spatial ordering. The ordering must be linear as the algorithm computes a dissimilarity score by calculating the area formed between two density traces when overlaid. Each region can only be connected to a maximum of two other regions in the ordering. Similarly, space filling curves such

Algorithm 3 Density Tracing: CalcSimilarity

Require: Density Trace for all datasets stored in DT;
Input: A reference feature f, dataset d and minimum similarity *minSimilarity*;
Output: The dissimilarity *dissimilarity* between traces of f and d;

segment list $referenceSegment \leftarrow DT_f$
segment list $datasetSegment \leftarrow DT_d$
length $l \leftarrow 0$
point list *areaPoints*
last intersection segment/point *lastIntersect*, *lastIntersectPoint*
number regions in polygon $numR \leftarrow 0$
region weight for polygon $rw \leftarrow 0$
for $s1 \in referenceSegment, s2 \in datasetSegment$ **do**
  if $INTERSECTION(s1, s2)$ **then**
    Save the intersection point: $areaPoints.append(INTERSECTION(s1, s2))$
    Work back from intersection point of other line (*datasetSegment*):
    iterator $it \leftarrow s2$
    while $it != NULL$ **do**
      if $it = lastIntersect$ **then**
        Save the intersect point: $areaPoints.append(lastIntersectPoint)$
        Save the length of the segment
        break
      else
        Save the start point: $areaPoints.append((*it).source())$
        Save the length of the segment
      end if
      $it--$
    end while
    Save the length of the segments up to the intersection point
    Increase num regions covered by polygon: $numR \leftarrow numR + 1$
    Update region weight covered by polygon: $rw \leftarrow rw + RegionWeight(s1.source())$
    Make polygon: $polygonP(areaPoints)$
    weight $w \leftarrow l/(totalSegmentLength(f) + totalSegmentLength(d))$
    avg region weight $rw \leftarrow rw/numR$
    $dissimilarity \leftarrow dissimilarity + areaPoints * w * rw$
    if $dissimilarity > minSimilarity$ **then**
      return -1
    end if
    Reset variables: $areaPoints.clear(), l \leftarrow 0, numR \leftarrow 0, rw \leftarrow 0$
    $lastIntersect \leftarrow s2, lastIntersectPoint \leftarrow INTERSECTION(s1, s2)$
  else if $s1 = referenceSegment.end()$ **then**
    Make a polygon from intersection point to end of segment
    Save the length of the segments up to the end of segment
    Increase num regions covered by polygon: $numR \leftarrow numR + 1$
    Update region weight covered by polygon: $rw \leftarrow rw + RegionWeight(s1.source())$
    Make polygon: $polygonP(areaPoints)$
    weight $w \leftarrow l/(totalSegmentLength(f) + totalSegmentLength(d))$
    avg region weight $rw \leftarrow rw/numR$
    $dissimilarity \leftarrow dissimilarity + areaPoints * w * rw$
    if $dissimilarity > minSimilarity$ **then**
      return -1
    end if
    Reset variables: $areaPoints.clear(), l \leftarrow 0, numR \leftarrow 0, rw \leftarrow 0$
  else
    Continue checking for intersection until last segment
    Save the length of these segments: $l \leftarrow l + s1.length() + s2.length()$
    Add reference segment start/end point:
    $areaPoints.append(s1.source()), areaPoints.append(s1.target())$
  end if
end for
if $dissimilarity > minSimilarity$ **then**
  return -1
else
  return $dissimilarity$
end if

as Morton-order or Hilbert curve [27] also cannot be used directly as the base map contains irregular regions that are not well represented by these curves. Section 5 discusses possible future work extending density tracing to neighborhood graphs.

Determining region ordering for use in our framework is similar to the symmetric Travelling Salesman Problem (TSP) where the cost between regions is the Euclidean distance. Starting from $Region_x$ we wish to order all regions in the study area so that the next region visited is a spatial neighbor (sharing a border). $Region_x$ can be chosen at random or, perhaps more usefully, it can be chosen based on some real world property (for example the Central Business District of a city). GLS is an intelligent search strategy for combinatorial optimization problems. The technique sits on top of local search procedures and has as a main aim to guide these procedures for exploring efficiently and effectively (the complete algorithm is out of the scope of this paper, please refer to [32] for details). We generate a random ordering starting from $Region_x$ and then apply GLS to generate the ordering that minimizes the Euclidean distance between regions (i.e. ideally we wish to move from neighbor to neighbor). We use the suggested GLS parameters from [32] (Fast Local Search, 2-Opt heuristic move operator, no construction heuristic and $Alpha = 0.167$).

DFS starts from $Region_x$ and pushes each neighboring region onto a stack. The closest neighbor is then popped from the stack and added to our region ordering. We continue in this fashion before backtracking to $Region_x$. The process then repeats for the remaining neighbors of $Region_x$. BFS is similar to DFS except we use a queue instead of a stack. It starts at $Region_x$ and adds all neighboring regions to the region ordering (in order of distance) and then continues the search outwards.

The NN technique that we use first adds the 5 (NN-5) closest neighbors of $Region_x$ to the region ordering. We then select the next closest neighbor (i.e. the 6th closest to $Region_x$) which becomes the new starting point. We repeat the process of adding the closest 5 neighbors until there are no regions left. We choose five closest regions as for our suburb dataset it is noted that suburbs have on average five neighbors.

Note that the ordering approaches adopted in this paper utilize the combination of topological information (neighboring regions sharing boundaries) and geometric information. Other approaches such as the Minimum Spanning Tree (MST) and cumulative distance ordering could be used with modification, however the former is not linear whilst the latter only uses topological information. We experimentally investigate the effect of region ordering in Section 4.

Once we determine the spatial ordering of regions in the study area, Step 32 of the algorithm is to calculate the density traces of each dataset. This trace projects the normalized density onto the Cartesian plane (following the defined ordering from Step 30) and depicts the spatial distribution of the density values within the study region.

To discover patterns of similar spatial distribution we need to query the set of density traces for similarity. To measure similarity we need to quantify the distance between two density traces. We modify the Spatial Trajectory Similarity Search technique [23] to incorporate region weights. Given a reference feature f, the most similar trace in D with respect to f is the one that minimizes the distance measure *Region Weighted Locality In-between Polylines* (RWLIP). Intuitively, two traces are considered spatially similar when they move close (i.e., their traces approximate each other) at the same place. As such RWLIP defines a distance function upon the traces (projected on the Cartesian plane) where the idea is to calculate the area of the

shape formed by the two traces. Note that this distance measure is equivalent to dissimilarity, i.e. a lower distance measure equates to higher similarity.

The distance measure between two traces Q and S is defined as follows:

$$RWLIP(Q, S) = \sum_{\forall polygon_i} Area_i \cdot w_i \cdot regionw_i, \tag{1}$$

where $polygon_i$ is a member of the set of polygons formed between intersection points I created when Q and S are overlaid in the 2D plane, $w_i \in [0, 1]$ is a contribution weight and $regionw_i \in [0, 1]$ is a region weight. Figure 5 illustrates the respective areas that contribute in $RWLIP(Q, S)$. Let $Length_Q(I_i, I_{i+1})$ and $Length_S(I_i, I_{i+1})$ be the length of the trace that participates in the construction of a given polygon. The contribution weight can then be defined as follows:

$$w_i = \frac{Length_Q(I_i, I_{i+1}) + Length_S(I_i, I_{i+1})}{Length_Q + Length_S}, \tag{2}$$

that is, the numerator is the perimeter of the polygon in question, while the denominator is the sum of the total length of the routes. It is a weight of how much a certain polygon contributes to the whole trace.

Each region of the reference feature f can have a user specified weight. It is designed to enable the combination of clustering results into the density trace similarity algorithm. The $regionw_i$ is defined as the average region weight of all regions that contribute to the polygon i. If no region weight is specified, then each areal unit is equally important and the global trend (the entire set of areal units) is used for the similarity calculation. On the other hand, the user can assign different weights to regions so that high peak areal units (clusters) have more effect on the similarity score. This enables the user to incorporate both context sensitive weighting and clustering into our system.

As the RWLIP algorithm traverses the spatial ordering, if there are no intersections between Q and S (the traces are parallel) then the algorithm detects this and closes the segments by connecting the initial points of Q and S and the final points of Q and S. The algorithm can then proceed with this one area. The range

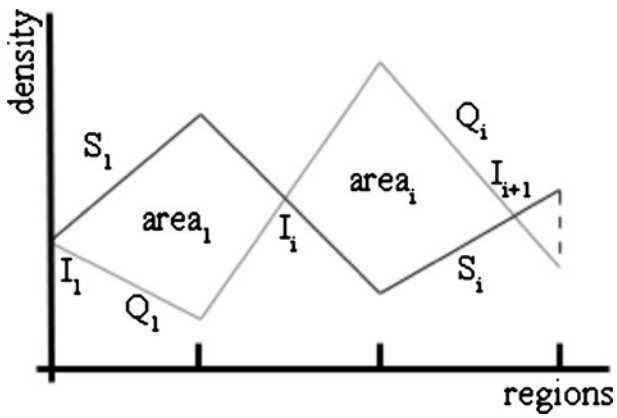


Fig. 5 Region weighted locality in-between polylines

of the similarity measure $RWLIP$ is dependent on the density range (Y axis) and number of regions (X axis). A lower dissimilarity measure equates to higher similarity between the density traces.

The last step of the algorithm is to save any features that show a similarity to our reference feature f. We can either use a user supplied minimum similarity *min_sim* or simply retrieve the *k Most Similar* and/or *k Least Similar* results.

3.4 Time complexity analysis

To analyze the time complexity of our approach we analyze each function of Algorithm 1 separately. Given n as the number of spatial areal aggregated datasets in D and l as the number of spatial regions in the base map B, `LoadDatasets`(D) is linear to n. The time complexity of `CalcCenters`(B) is dependent on the regions of B. Typically `CalcCenters`(B) is linear to l, however if B contains convex polygon regions then the complexity is $O(\log l)$ [26]. To normalize D, for each dataset d_n we must examine l values, and thus `NormalizeDensity`(D) has a time complexity of $O(n \times l)$. To generate the spatial ordering of our base map B using the default GLS method, `RegionOrdering`(B, *orderAlg*) typically requires $O(l \log l)$ [32]. To generate the density trace of each dataset d_n we need to examine l regions, thus `DensityTrace`(D) requires $O(n \times l)$. Note that the original $LIP(Q, S)$ computation requires $O(l \log l)$ [23], thus our $RWLIP$ extension also has a time complexity of $O(l \log l)$. `CalSimilarity`(f, d_i, *minSimilarity*) requires the comparison of n datasets using the $RWLIP$ algorithm and thus has a time complexity of $O(n \times l \log l)$.

4 Experimental results

This section provides experimental evaluation and comparison of our approach. The base region map used for our experiments are the 216 urban suburbs of Brisbane, the capital city of Queensland, Australia. Sections 4.1 and 4.3 use the whole study region while Section 4.2 uses a small subset of the base map.

4.1 Experiments with synthetic datasets

The examples in Sections 3.1 and 3.2 are illustrative experiments that are designed to explain our technique of using spatial distribution to find co-patterning relationships between datasets. We conduct a number of experiments with synthetic datasets to evaluate and justify our approach. The synthetic datasets are produced using a MATLAB program which generates random point data drawn from a mixture of multivariate guassians. We then convert this point set to an areal aggregated dataset by assigning points to regions of the given base map. For the figures in this section we display the point data overlaid onto the base map instead of the density aggregates to aid readability.

We first evaluate our approach with synthetic datasets that show increasing dissimilarity. Figure 6a–d show the synthetic datasets used in this experiment. We start with $n = 4$ clusters in $dataset_a$, and for each subsequent dataset we remove k clusters ($0 < k < n$). We assert that the linear decrease in the number of clusters

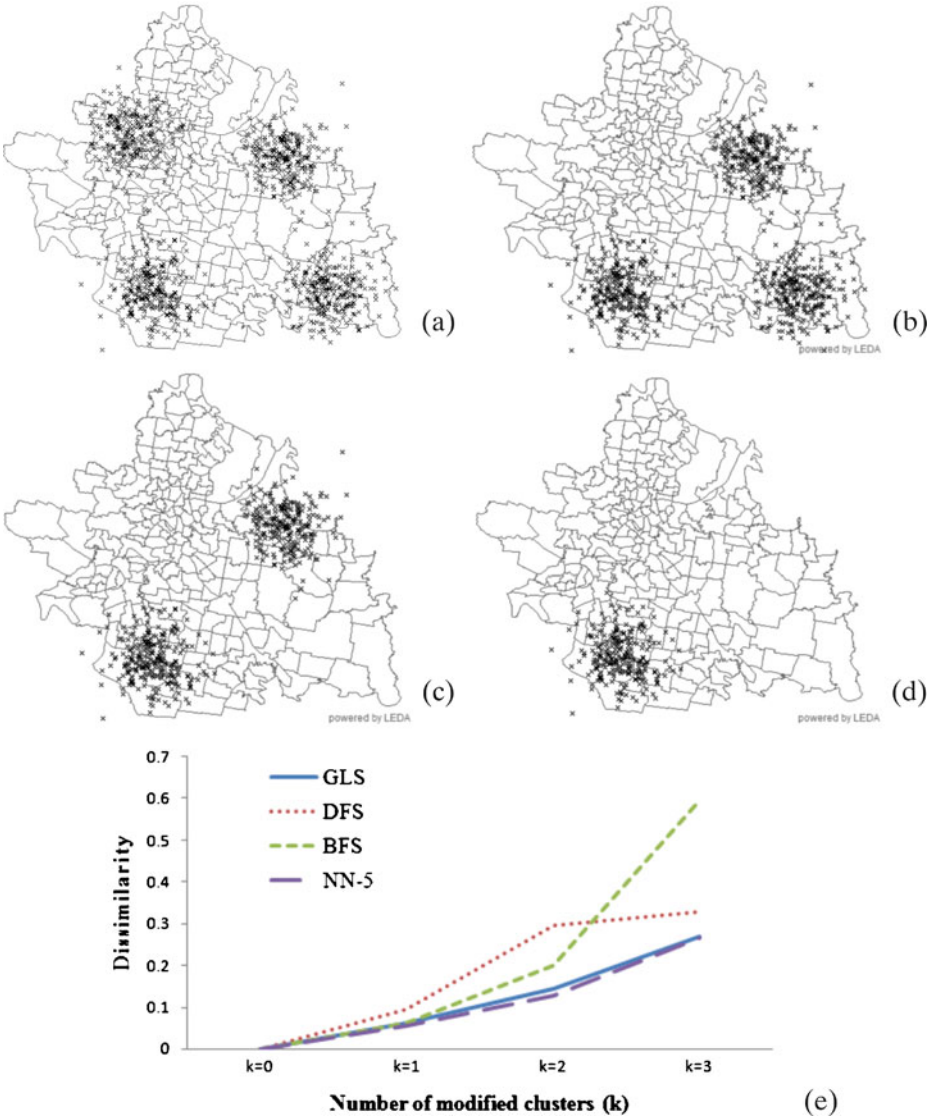


Fig. 6 Synthetic experiment removing clusters: **a–d** Datasets $dataset_{a-d}$; **e** Increasing dissimilarity with respect to number of removed clusters

results in these datasets showing increasing dissimilarity, that is, as clusters are removed, $dataset_{b-d}$ become less similar to the original $dataset_a$. Figure 6e illustrates this; when we select $dataset_a$ as the reference feature the dissimilarity increases with the number of clusters removed. For these synthetic datasets all four region orderings identify the increasing dissimilarity as we remove clusters from the datasets.

The second experiment we present is a modification of the first synthetic experiment. Figure 7 starts with $n = 4$ clusters in disjoint regions of the feature space and for each subsequent dataset we move the k^{th} cluster to a new location within

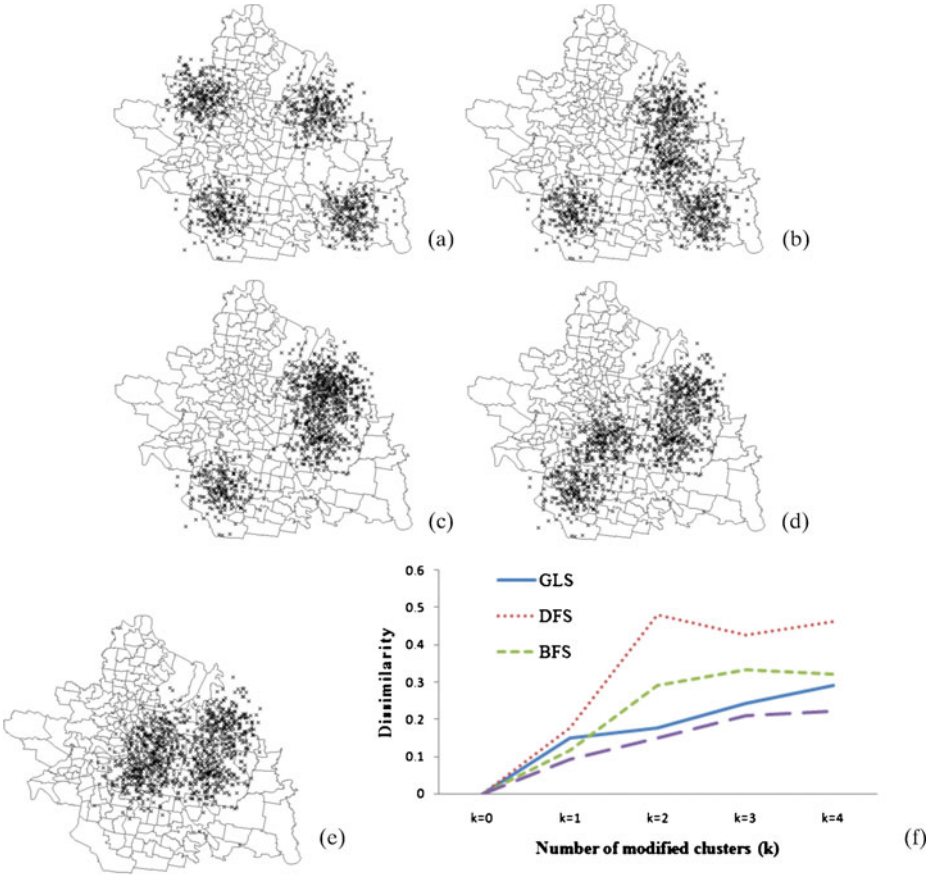


Fig. 7 Synthetic experiment moving clusters: **a**–**e** Datasets $dataset_{a-e}$; **f** Increasing dissimilarity with respect to number of moved clusters

the space. The main difference between this and the previous experiment is that n does not vary between datasets. We assert that the linear decrease of clusters in their original locations will result in an increase in dissimilarity as we move clusters. Figure 7f shows that when we select $dataset_a$ as the reference feature, only the GLS and NN-5 orders give this expected result for all datasets, that is, they show an increase in dissimilarity. When $k = 2$ ($dataset_c$) the DFS order records a peak in dissimilarity, calculating that $dataset_d$ and $dataset_e$ are more similar to the reference feature than $dataset_c$. A similar result can be seen for the BFS ordering when $k = 3$.

To measure similarity between datasets we compare the spatial distribution, that is, how the density changes between regions. For these synthetic datasets, the GLS and NN-5 spatial ordering are better able to preserve the spatial neighborhood information, as reflected in the fact that the DFS and BFS orders give unexpected results for part of this experiment. We use these synthetic datasets to evaluate and justify our approach as there is no baseline measure that can be used to measure

similarity, we rely on the linear decrease in the number of clusters to confirm that these datasets show increasing dissimilarity.

4.2 Optimal spatially aware ordering

To further investigate the effect of the four linear spatial ordering methods, we generate all possible orderings for a given base map and find the optimal solution that minimizes the dissimilarity score for a given control dataset. This enables us to compare the optimal linear ordering to the orders generated by our techniques.

We choose a small subset of eight regions from the Brisbane base map as our study area. For this study area there are 8! possible spatial orders. We generate all possible orders to determine the optimal spatial order that will minimize the dissimilarity score for the datasets shown in Fig. 8a–d. We assert that $dataset_a$ is similar to $dataset_b$ as they show a similar high density distribution in the three top left regions and low density distribution in the remaining regions. It can also be seen that $dataset_a$ is very dissimilar to $dataset_{c-d}$. Figure 9a–d show the GLS, DFS, BFS and NN-5 orders for the study region respectively. To enable comparison we use the same starting region for each order. Constraining the optimal ordering to the same starting region as our three techniques reduces the number of possible orders to 7!.

Figure 10 shows a comparison of the results obtained by the density tracing algorithm for each spatial ordering. The dissimilarity values are normalized using the *min-max* technique described in Section 3.3. Both GLS, NN-5 and DFS are good approximations of the optimal ordering. Figure 9d shows that the optimal order is very similar to the orders generated using our three other techniques with the same starting region. In particular, the GLS order is most similar. The worst case scenario for generating the spatial ordering is when the optimal order reflects a completely different region ordering. Figure 9e shows the unconstrained optimal

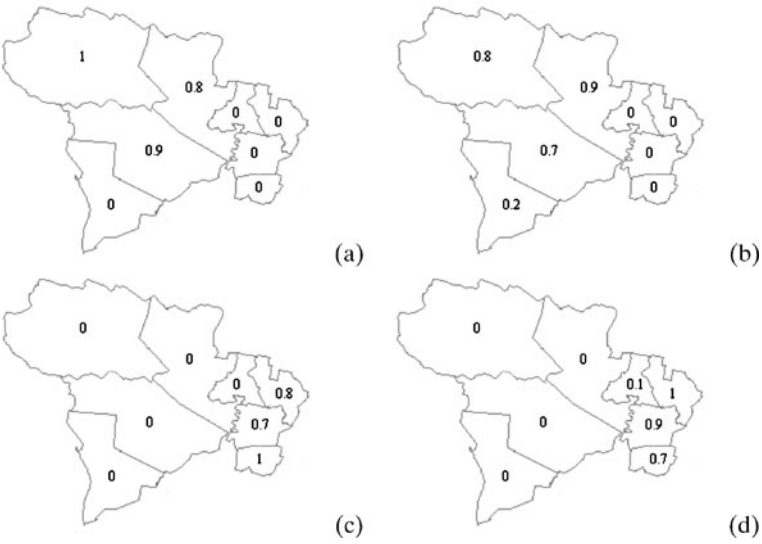


Fig. 8 A subset of regions from the Brisbane base map with synthetic data: **a–d** Datasets $dataset_{a-d}$

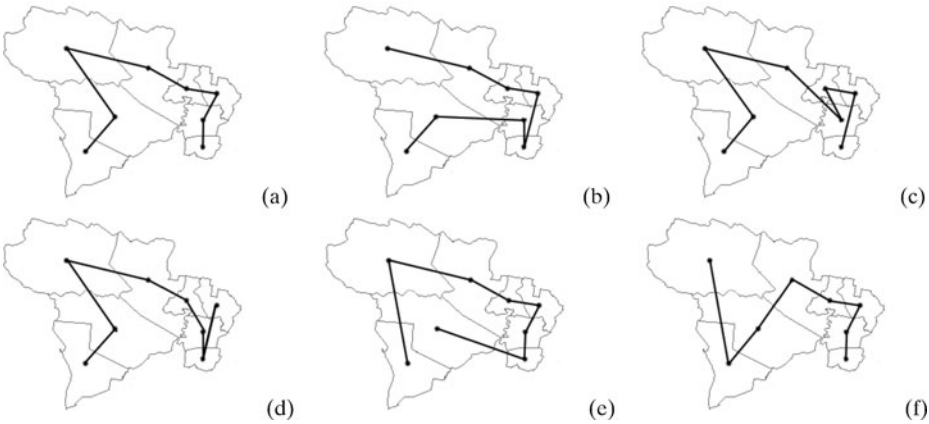


Fig. 9 Optimal ordering: **a** GLS; **b** DFS; **c** BFS; **d** NN-5; **e** Optimal order with same starting region; **f** Optimal order

order for this study region. For these experiments the unconstrained optimal order is not the worst case scenario. It is noted that we do not need an optimal order, only one that can model spatial neighborhood information so that we can compare the spatial distribution of datasets. From the comparison of this small subset and visually from Fig. 4 we can see that GLS is suited to capture spatial information for these study regions.

4.3 Experiments with real crime datasets

This section examines the real crime dataset from 216 urban suburbs of Brisbane. The study region is highly dynamic and active. It continues to experience significant and sustained population growth and various criminal activities [20]. The Queensland Police Service (QPS) releases crime data in areal aggregated format due primarily to privacy concerns. We combine these crime datasets with spatial feature datasets so that interesting relations can be discovered. We use a total of 29 crime datasets and 5 features (reserves, schools, hospitals, university/colleges and parks) in this experiment. The crime dataset from the QPS has three main categories: personal

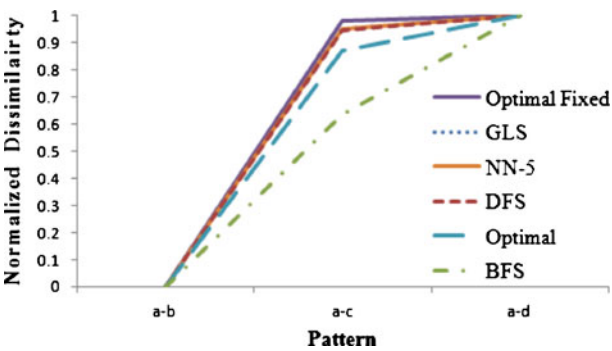


Fig. 10 Comparison of normalized density dissimilarity of synthetic datasets using various orders

Table 1 Crime abbreviations

TOHO	Total homicide
MSLT	Manslaughter (excl. by driving)
DRCD	Driving causing death
SEAS	Serious assault
OTAS	Other assault
RAAR	Rape and attempted rape
OTSO	Other sexual offences
ARRO	Armed robbery
UNRO	Unarmed robbery
EXTO	Extortion
KAAE	Kidnapping & abduction etc.
OFAP	Other offences against the person
TOAPR	Total offences against property
TOUE	Total unlawful entry
ARSO	Arson
OTPD	Other property damage
MOVT	Motor vehicle theft
OTTH	Other theft (excl. unlawful entry)
STFD	Stealing from dwellings
SHST	Shop stealing
OTST	Other stealing
FBCH	Fraud by cheque
FBCC	Fraud by credit card
OTFR	Other fraud
TOOO	Total other offences
TRAV	Trespassing and vagrancy
GOOO	Good order offences
TARO	Traffic and related offences
MIOF	Miscellaneous offences

safety (offences against person), property security (offences against property) and other offences. Table 1 lists all crime types we study in this experiment. GLS is used as the region ordering due to its low edge overlap and low edge variance (Section 3.3).

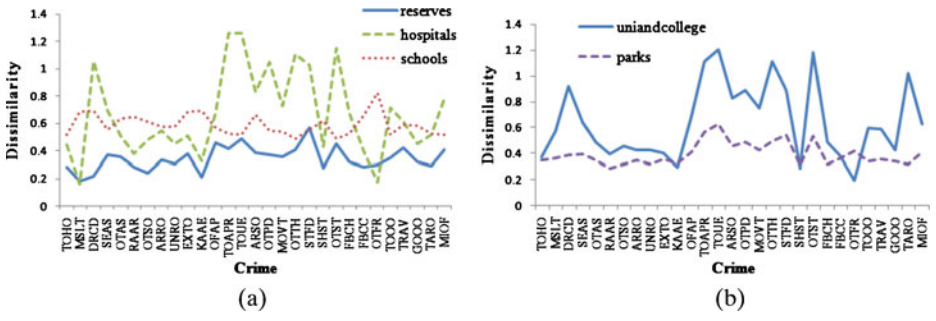


Fig. 11 Similarity between features and various crime datasets: **a** Reserves, hospitals and schools; **b** University/colleges and parks

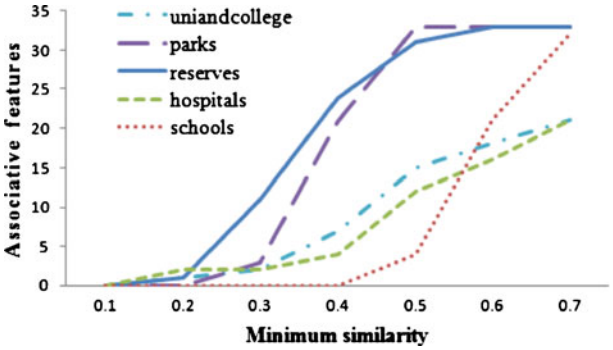


Fig. 12 Lowering minimum similarity to reduce associative features

Figure 11 shows the result of this experiment. An interesting pattern that we detected is that the reference feature *reserves* shows a very high similarity to both *kidnapping (KAAE)* and *manslaughter (MSLT)* (the *2 Most Similar*). It is interesting to note that *parks*, while similar physically to *reserves*, do not exhibit the same pattern. Another pattern discovered is *universities and colleges* show a highly similar density trace to both *shop stealing (SHST)* and *other fraud (OTFR)*, again these are the *2 Most Similar*. These results can then be used by domain experts to further investigate the cause of these specific patterns.

There is a point where the number of discovered patterns can become too much for the end user to handle—so called 'information overload'. As can be seen from Fig. 12 a low minimum similarity will not reveal any patterns, whereas a high limit may introduce unwanted 'noise'. We offset this drawback by allowing the user to retrieve the *k Most Similar* and/or *k Least Similar* results. It is noted that even with a high minimum similarity the number of patterns generated by our approach is much smaller than the number of patterns returned by ARM. We compare and contrast the two approaches in Section 4.4.

Figure 13 shows a comparison between the four region ordering approaches GLS, DFS, BFS and NN-5. The four techniques show similar results in this case, demonstrating the minimal role of different orderings for this particular dataset.

In the next experiment we introduce clustering results into the framework. We select the reference feature *reserves* and assign weights to specific regions. These

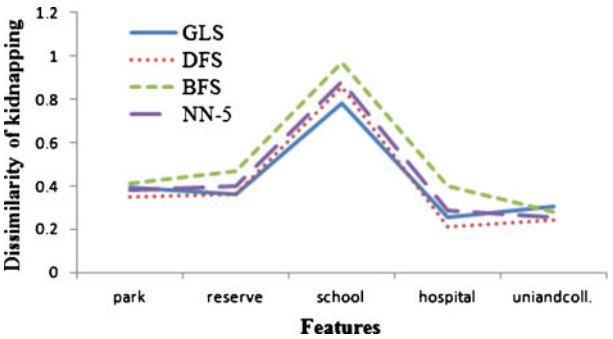


Fig. 13 Comparison between GLS, DFS and BFS for `kidnapping`

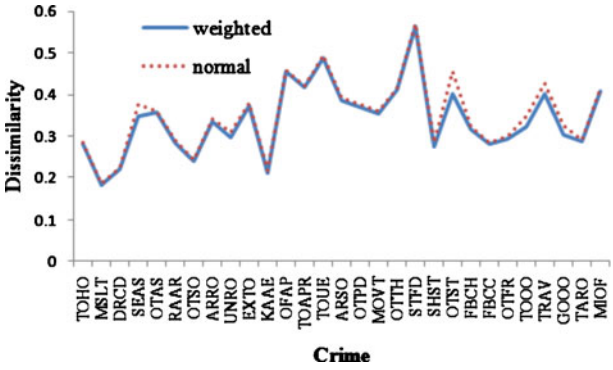


Fig. 14 Comparison between weighted and non-weighted regions for `reserves`

regions can be found using a clustering technique or may be a user defined influence on the study area. We discover two clusters in the *reserves* dataset; suburb *Coorparoo* is the densest cluster and is assigned a region weight of 0.1 while suburb *Tennyson* a weight of 0.5. Figure 14 compares the dissimilarity scores of *reserves* when cluster weights are used and when they are not. The region weights have lowered the dissimilarity of a number of crime types, with the most affected being *Serious Assault (SEAS)*, *Other Stealing (OTST)* and *Trespassing (TRAV)*.

4.4 Comparison with association rules mining

ARM discovers co-patterning relationships by capturing point-to-point association of frequent patterns that have support and confidence greater than some user specified minimum support and minimum confidence thresholds. Our approach aims to discover co-patterning relationships by modeling the spatial distribution of datasets. We use the same areal aggregated dataset as described in Section 4.3 for a comparison with ARM. Before we can use ARM, we must first transform the data into categorical values. We classify the density values into three groups; high, medium and low (using normalized values). This results in a database of 318 columns (items). ARtool [7] (Apriori algorithm) is used to perform the association mining. It is noted that while many patterns discovered by our density tracing approach may be the same as patterns discovered from ARM, the two are not directly comparable. ARM measures the point-to-point association of datasets while our approach measures the spatial distribution.

One of the drawbacks of ARM is that typically a lot of *uninteresting* rules are found. With a minimum support and minimum confidence of 0.8, the Apriori algorithm generated 2,319,036 rules. This increases the complexity for the end user as they must sift through results looking for *useful information*. The column graph in Fig. 15 shows that as the confidence is increased the number of rules generated decreases, however there are still an unmanageable number of patterns (support is fixed at 0.8). The line chart in Fig. 15 compares the results obtained from the Top-100 density tracing approach to ARM. The percentage depicts the number of patterns from the Top-100 that were also discovered by ARM at varying confidence levels. With fixed support of 0.8 and confidence of 0.85, all the patterns discovered by our approach are also discovered by ARM. With a confidence level of 0.95, 57% of the

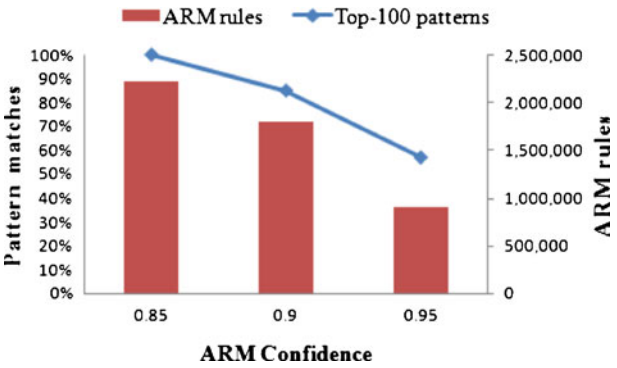


Fig. 15 Comparison between the Top-100 patterns from our density tracing approach and ARM with support 0.8 and varying confidence levels

patterns match; these patterns are especially interesting as they show a similar density distribution and also strong point-to-point association.

One of the reasons many other reasoning techniques use a combination of clustering and ARM is because ARM is generally slow for large datasets. With a minimum support and minimum confidence of 0.8 the computation time for ARM was 32,876 msec, this is compared with an average of 867 msec for our experiments in Section 4.3 (both performed on a Intel P4 3.2Ghz with 1GB RAM). Our density trace based approach efficiently produces interesting co-patterning relationships for areal aggregated crime datasets. These patterns can then be used by domain experts for confirmatory analysis to help answer the *why* question of crime analysis which can have the greatest impact on crime management and prevention.

5 Final remarks

We have presented a novel reasoning approach that uses density tracing to allow autonomous exploratory analysis and knowledge discovery in areal aggregated crime datasets. The spatial distribution of a dataset is represented as a density trace to allow the discovery of co-patterning relationships that can offer a deeper insight into the complex nature of criminal behavior. It successfully discovers both positive and negative co-patterning among crime incidents and spatial features and is computationally efficient. We overcome the drawbacks of current areal aggregated reasoning approaches by using the global spatial distribution (density trend) that models density change between regions. Through the use of regions weights we are also able to incorporate the use of both context sensitive weighting and clustering into our system.

The approach presented here is part of a larger research project aimed at reasoning within massive crime datasets. Future work includes extending our framework to consider temporal density traces over the same study region and improving the algorithm efficiency by investigating the further integration of clustering methods within the density traces. We plan to further investigate the use of density tracing as a preprocessing step for ARM for the discovery of patterns that have a strong combination of both point-to-point association and spatial distribution. A similar

neighborhood graph based approach for using changes in density to discover co-patterning is also being researched.

References

1. Agrawal R, Imielinski T, Swami AN (1993) Mining association rules between sets of items in large databases. In: Buneman P, Jajodia S (eds) Proceedings of the ACM SIGMOD'93 international conference on management of data. ACM Press, Washington, DC, pp 207–216
2. Bailey TC, Gatrell AC (1995) Interactive spatial analysis. Longman Scientific & Technical, Harlow, UK
3. Boba R (2005) Crime analysis and crime mapping. Sage Publications, Thousand Oaks, California
4. Chen H, Chung W, Xu JJ, Wang G, Qin Y, Chau M (2004) Crime data mining: a general framework and some examples. Computer 37(4):50–56
5. Craglia M, Haining R, Wiles P (2000) A comparative evaluation of approaches to urban crime pattern analysis. Urban Stud 37(4):711–729
6. Cressie NAC (1991) Statistics for spatial data. Wiley Series in Probability and Statistics, New York
7. Cristofor L (2002) ARtool: association rule mining algorithms and tools. http://www.cs.umb.edu/~laur/ARtool/
8. Dent BD (1999) Cartography: thematic map design. WCB McGraw Hill, Boston
9. Estivill-Castro V, Lee I (2001) Data mining techniques for autonomous exploration of large volumes of geo-referenced crime data. In: Pullar DV (ed) Proceedings of the 6th international conference on geocomputation, Brisbane, Australia. GeoComputation CD-ROM
10. Estivill-Castro V, Lee I (2002) Argument free clustering via boundary extraction for massive point-data sets. Comput Environ Urban Syst 26(4):315–334
11. Han J, Kamber M, Tung KH (2001) Spatial clustering methods in data mining. In: Miller HJ, Han J (eds) Geographic data mining and knowledge discovery. Cambridge University Press, Cambridge, UK, pp 188–217
12. Hirschfield A, Brown P, Todd P (1995) Gis and the analysis of spatially-referenced crime data: experiences in Merseyside UK. J Geogr Inf Syst 9(2):191–210
13. Huang Y, Pei J, Xiong H (2006) Mining co-location patterns with rare events from spatial data sets. Geoinformatica 10(3):239–260. doi:10.1007/s10707-006-9827-8
14. Huang Y, Shekhar S, Xiong H (2004) Discovering co-location patterns from spatial datasets: a general approach. IEEE Trans Knowl Data Eng 16(12):1472–1485
15. Koperski K, Han J (1995) Discovery of spatial association rules in geographic information databases. In: Proceedings of the 4th international symposium on large spatial databases. LNCS. Springer, Portland, Maine, pp 47–66
16. Lee I, Phillips P (2008) Urban crime analysis through areal categorized multivariate associations mining. Appl Artif Intell 22(5):483–499
17. Lee S (2001) Developing a bivariate spatial association measure: an integration of Pearson's r and Moran's I. J Geogr Syst 3(4):369–385
18. Mennis J, Liu JW (2005) Mining association rules in spatio-temporal data: an analysis of urban socioeconomic and land cover change. Trans GIS 9(1):5–17. doi:10.1111/j.1467-9671.2005.00202.x. URL: http://www.blackwell-synergy.com/doi/abs/10.1111/j.1467-9671.2005.00202.x
19. Miller HJ, Han J (2001) Geographic data mining and knowledge discovery. Taylor and Francis, London
20. Murray AT, McGuffog I, Western JS, Mullins, P (2001) Exploratory spatial data analysis techniques for examining urban crime. Br J Criminol 41:309–329
21. Oatley G, Ewart B, Zeleznikow J (2006) Decision support systems for police: lessons from the application of data mining techniques to soft forensic evidence. Artif Intell Law 14(1):35–100. doi:10.1007/s10506-006-9023-z
22. Okabe A, Boots BN, Sugihara K, Chiu SN (2000) Spatial tessellations: concepts and applications of voronoi diagrams, 2nd edn. Wiley, West Sussex
23. Pelekis N, Kopanakis I, Marketos G, Ntoutsi I, Andrienko G, Theodoridis Y (2007) Similarity search in trajectory databases. In: TIME '07: proceedings of the 14th international symposium on temporal representation and reasoning. IEEE Computer Society, Washington, DC, USA, pp 129–140. doi:10.1109/TIME.2007.59

24. Ratcliffe J (2004) The hotspot matrix: a framework for the spatio-temporal targeting of crime reduction. In: Police practice and research, vol 5, pp 5–23
25. Ratcliffe J, McCullagh M (1998) Identifying repeat victimization with Gis. Br J Criminol 38(4):651–662
26. Rigaux P, Scholl M, Voisard A (2001) Spatial databases: with application to GIS. Morgan Kaufmann, San Francisco, CA
27. Samet H (2005) Foundations of multidimensional and metric data structures (the Morgan Kaufmann series in computer graphics and geometric modeling). Morgan Kaufmann, San Francisco, CA, USA
28. Shalabi LA, Shaaban Z, Kasasbeh B (2006) Data mining: a preprocessing engine. J Comput Sci 2:735–739
29. Shekhar S, Huang Y (2001) Discovering spatial co-location patterns: a summary of results. In: Jensen CS, Schneider M, Seeger VJ, Tsotras B (eds) Proceedings of the 7th international symposium on the advances in spatial and temporal databases. Lecture notes in computer science, vol 2121. Springer, Redondo Beach, CA, pp 236–256
30. Tobler W (1979) Cellular geography. Philos Geogr, pp 379–386
31. Voudouris C (1997) Guided local search for combinatorial optimisation problems. PhD thesis, Department of Computer Science, University of Essex, Colchester, UK
32. Voudouris C, Tsang E (2003) Handbook of metaheuristics, chap Guided Local Search. Springer, pp 185–218
33. Wortley R, Mazerolle L (2008) Environmental criminology and crime analysis. Willan Publishing
34. Yoo JS, Shekhar S (2006) A joinless approach for mining spatial colocation patterns. IEEE Trans Knowl Data Eng 18(10):1323–1337. doi:10.1109/TKDE.2006.150

Peter Phillips is a PhD candidate in the School of Business, Discipline of IT at James Cook University, Australia. His research interests cover intelligent crime analysis combining heterogeneous data types to facilitate the exploratory analysis of complex spatio-temporal datasets and to allow reasoning towards better decision making.

Ickjai Lee obtained his PhD in 2002 from the School of Electrical Engineering and Computer Science, University of Newcastle, in Australia. After a year as a postdoctoral research fellow at the Business and Technology Laboratory in the University of Newcastle, Australia, Lee joined the School of IT at James Cook University, Australia. He has been actively involved in working on broad areas of geoinformatics and geocomputation. His research interests include geospatial data mining, multiple classifiers, geospatial databases, conceptual spaces, and Voronoi tessellations. Recently, he focuses on space tessellations through generalized Voronoi diagrams for effective emergency management. He is currently Head of IT Discipline and Associate Professor in the School of Business, Discipline of IT at James Cook University.