Anomaly Detection for Data with Spatial Attributes

P. Deepak

Abstract The problem of detecting spatially-coherent groups of data that exhibit anomalous behavior has started to attract attention due to applications across areas such as epidemic analysis and weather forecasting. Earlier efforts from the data mining community have largely focused on finding outliers, individual data objects that display deviant behavior. Such point-based methods are not easy to extend to find groups of data that exhibit anomalous behavior. Scan statistics are methods from the statistics community that have considered the problem of identifying regions where data objects exhibit a behavior that is atypical of the general dataset. The spatial scan statistic and methods that build upon it mostly adopt the framework of defining a character for regions (e.g., circular or elliptical) of objects and repeatedly sampling regions of such character followed by applying a statistical test for anomaly detection. In the past decade, there have been efforts from the statistics community to enhance efficiency of scan statistics as well as to enable discovery of arbitrarily shaped anomalous regions. On the other hand, the data mining community has started to look at determining anomalous regions that have behavior divergent from their neighborhood. In this chapter, we survey the space of techniques for detecting anomalous regions on spatial data from across the data mining and statistics communities while outlining connections to well-studied problems in clustering and image segmentation. We analyze the techniques systematically by categorizing them appropriately to provide a structured birds-eye view of the work on anomalous region detection; we hope that this would encourage better cross-pollination of ideas across communities to help advance the frontier in anomaly detection.

1 Introduction

Anomaly may be broadly defined as something that departs from what is generally regarded as *normal* (i.e., common). Thus, a group of objects can be considered as an anomaly if their collective behavior deviates from what is regarded as common.

P. Deepak (✉)
IBM Research - India, Bangalore, India
e-mail: deepaksp@acm.org

© Springer International Publishing Switzerland 2016

M.E. Celebi, K. Aydin (eds.), *Unsupervised Learning Algorithms*,
DOI 10.1007/978-3-319-24211-8_1

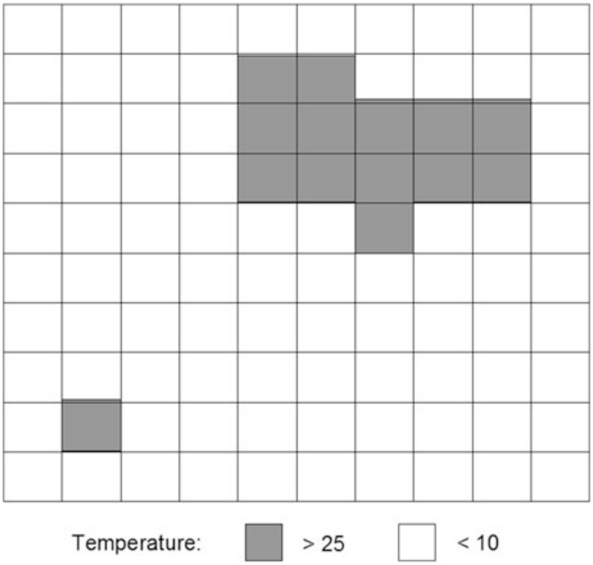

Fig. 1 Example grid dataset

For example, a county where schools record an average of 80 % pass in a test where the national average is 40 % would be classified as an anomaly under this definition. On data that have a spatial positioning, anomalous groups that are coherent on the spatial attributes (e.g., objects that fall within a county, as in the example) are often the anomalies of interest. Figure 1 illustrates a geographic region gridded into squares, each cell in the grid colored according to the temperature recorded within it. For simplicity, we consider that each cell takes a temperature from either of two ranges, the higher range being greater than 25 units, and the lower range set at less than 10 units. While most of the cells in the grid record the lower range, two regions, a large set of contiguous cells in the top and a single cell region at the bottom left are seen to record the higher range. According to the definition of anomaly, both these regions would intuitively be classified as anomalies due to being high-temperature regions within a largely cooler grid.

Outlier is a related but more fine-grained concept in that it quantifies uncommon behavior at the level of individual objects. Without going into the technical details of what would be an outlier, the cell in the lower left region of the grid in Fig. 1 would be regarded as an outlier since it is seen to be warm while both its local neighborhood as well as the entire grid are significantly cooler. This brings us to finer details of the definition of an outlier, where there is no consensus on what should be chosen as the baseline to contrast a candidate outlier with. This, however, is often due to good reason since some applications may inherently warrant a comparison with the local neighborhood whereas others may be more concerned about the level of contrast with the global behavior. Under the global comparison paradigm, each cell in the warm region in the top-half of the grid would be deemed to be an outlier. However, the local comparison would unveil a different story; most of the cells in the warm region have warm cells as neighbors, except perhaps the bottom-most warm cell that protrudes into a cooler neighborhood. Thus, the local comparison approach would classify most of the cells in the warm region as non-outliers.

Terminology Throughout this paper, we will use the term *spatial data* as a shorthand to describe data that has spatial attributes in addition to other attributes (like temperature in the example above); in particular, it is necessary to mention here that we exclude purely spatial data, those which have only spatial attributes, from our definition of spatial data, for this chapter.

Roadmap In this chapter, we intend to survey techniques from across diverse disciplines that could be used towards identifying anomalous regions on data with spatial attributes. We focus on unsupervised methods and restrict the discussion to simple objects that may be described by a set of values, one per attribute; for example, we would not consider anomaly identification in multimedia such as videos. We organize the content into seven sections, as described below.

- **Problem Definition and Taxonomy of Techniques:** We start with a section outlining the problem of anomaly detection in full generality, and present a taxonomy of approaches that address the problem. The rest of the sections are organized so as to mostly focus on specific areas within the taxonomy.
- **Object Anomalies:** In this section, we survey the various approaches for anomaly detection at the object level. We will consider the applicability of general outlier detection methods to the task of identifying anomalous regions on spatial data, as well as describe a method from literature that was specifically proposed for the task.
- **Region Anomalies—Global:** This section describes approaches that seek to identify globally anomalous regions, i.e., regions comprising groups of objects that are divergent from global behavior. This encompasses methods from statistics such as spatial scan as well as a few mining-based approaches.
- **Region Anomalies—Local:** Here, we consider approaches that look for regions that are anomalous when contrasted against behavior in the local neighborhood. These include the localized homogeneous anomaly (LHA) detection method as well as image segmentation methods that address a stricter version of the problem from a different domain.
- **Region Anomalies—Grouping:** In this section, we will survey techniques that target to group objects in such a way that groups are coherent while being mutually divergent. Techniques in this category group objects such that each group is divergent from other groups in the output; some of the groups in the output may be regarded as anomalies based on the level of contrast. We will specifically look at grouping methods for spatial data as well as clustering-based anomaly detection methods from intrusion detection literature.
- **Discussion:** In this discussion section, we will explore other ways of using clustering methods in the anomaly detection pipeline. In this context, we will briefly describe density-based clustering algorithms.
- **Directions for Future Work:** In this section, as the name suggests, we outline promising directions for future work in the area of anomaly detection.

2 Problem Definition and Taxonomy of Techniques

2.1 Problem Definition

We now define the problem of anomaly detection on spatial data at a level of generality so that most anomaly detection techniques would be applicable. Consider a set of objects $\mathscr{D} = \{d_1, d_2, \ldots, d_n\}$ where each object d_i may be represented by the values it takes for a set of attributes from a pre-defined schema:

$$d_i = [s_{i1}, s_{i2}, \ldots, s_{im_s}, v_{i1}, v_{i2}, \ldots, v_{im_v}] \tag{1}$$

The first m_s attributes are designated as spatial attributes, whereas the remaining m_v are non-spatial attributes; we will call the non-spatial attributes as value attributes. In our example from Fig. 1, each cell is a data object with the spatial attributes being spatial location defined by the x and y co-ordinates, and a single value attribute denoting the temperature. In the case of studies on epidemics, data points may be people with their spatial attribute being their geographic co-ordinates and their value attribute being a single boolean attribute indicating whether they are diseased or not. In case of weather modeling, temperature, humidity, etc., may form different value attributes.

Additionally, anomaly identification methods use a notion of proximity between data points, defined based on their spatial attributes. For data in the form of grid cells, the proximity measure may simply be the adjacency matrix that could then be used to determine whether a set of cells is connected or not. In the case of data in the form of points, this may be a similarity measure defined on the spatial attributes.

The problem of anomaly detection is to identify a set of anomalies $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$ such that each set $A_i \subseteq \mathscr{D}$ adheres to the following criteria:

- **Spatial Coherence:** The set of objects in A_i satisfy a coherence condition defined on the *spatial attributes*; examples are that they are the only objects in a circular or elliptical region within the space defined by the spatial attributes, or that they form a set of connected cells in a gridded dataset.
- **Contrast from Context:** A model built over objects in A_i using their *value attributes*, contrasts well with those built over objects in $C(A_i)$, the *context* of A_i. The model could simply be the *mean* of temperatures over cells and the context could be the set of cells adjoining A_i (i.e., local) or the set of all cells in the dataset (i.e., global). Typical models used in anomaly detection are simple in construction, with the mean being among the most commonly used models.

Some anomaly detection methods, especially, those that come from the clustering family, use a different construction to handle the second condition; we will elaborate on those aspects in respective sections. There are different possibilities to outline the spatial coherence condition; so is the case with building the value attribute model for each A_i, defining the context and specifying the contrast condition between models. Different anomaly detection techniques differ in the specifications used for each of the above phases.

Applications Beyond Spatial Data Though we will discuss algorithms and techniques designed for and motivated by scenarios involving spatial data where geographic attributes form spatial attributes, the problem itself is much more general. There are motivating scenarios for anomaly detection where metadata attributes replace the role of spatial attributes. For example, an enhanced rate of incidence of disease among teens in a specific county could be an anomalous pattern; the methods for spatial anomaly detection would be able to find such anomalies if age is modeled as a "spatial" attribute. Similarly, there are other motivating anomaly detection scenarios where ordinal attributes such as weight, height, education level, etc. may be modeled as "spatial" attributes followed by application of spatial anomaly detection methods. In short, the problem we are looking at in this chapter as well as techniques for addressing it, are applicable beyond data having conventional spatial attributes.

2.2 Taxonomy of Techniques

Figure 2 illustrates a taxonomy of methods that have been explored in anomaly detection on spatial data. The greyed boxes in the taxonomy represent methods for anomaly detection that do not differentiate between attributes as spatial and value attributes; thus, these are not readily applicable to the problem of anomaly

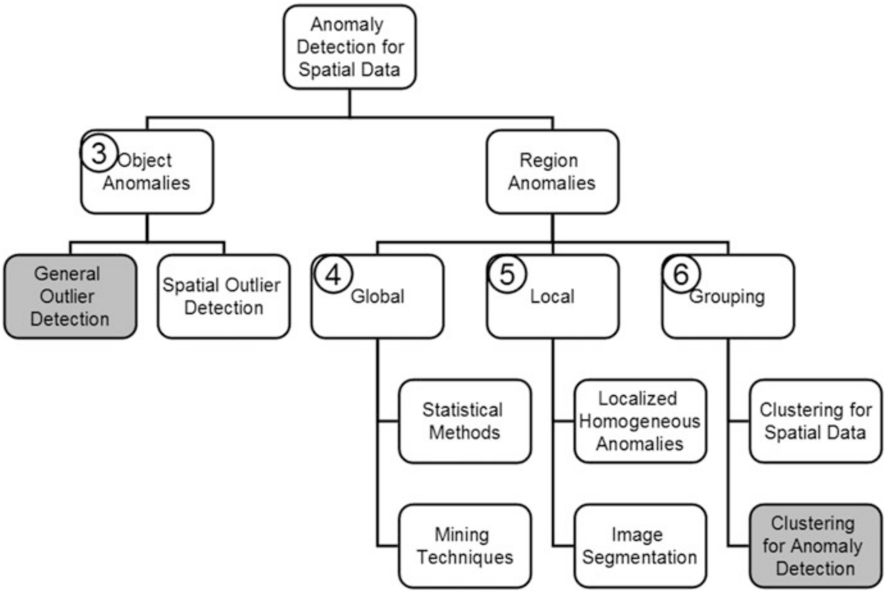
Anomaly Detection for Spatial Data
3
Object Anomalies
Region Anomalies
General Outlier Detection
Spatial Outlier Detection
4
Global
5
Local
6
Grouping
Statistical Methods
Mining Techniques
Localized Homogeneous Anomalies
Image Segmentation
Clustering for Spatial Data
Clustering for Anomaly Detection

Fig. 2 Taxonomy of approaches for anomaly detection

detection on spatial data. However, we include those in our taxonomy since they may be adapted to the problem in question. We now briefly describe each class in the taxonomy, leading up to a detailed treatment in subsequent sections. Figure 2 also indicates the section number where each sub-tree of the taxonomy is described.

- **Object Anomalies:** Techniques under this category enforce that the output anomalies are all made up of singleton sets, i.e., $\forall i, |A_i| = 1$, and are thus also called outlier detection techniques.
  - **General Outlier Detection:** General outlier detection methods,[1] the ones that we will survey under this class, do not differentiate attributes as spatial or value attributes. However, they frequently use concepts such as neighbors and neighborhood which are more meaningful in the context of spatial attributes; thus, these may be seen as techniques that do not use value attributes, i.e., those that assume $m_v = 0$. They mostly estimate the outlier-ness of an object based on whether there is disparity between density of objects around them and the density of objects around their neighbors. We will also comment on how such methods can be adapted to the problem of outlier detection on spatial data.
  - **Spatial Outlier Detection:** Unlike general outlier detection methods, outlier detection applied to spatial data is relatively less studied. Spatial outliers are those that take on values different from those of their spatial neighbors. Among the few methods that have been proposed under this category, we will take a detailed look at SLOM [7].
- **Region Anomalies:** As the name suggests, methods for region anomaly detection seek to identify anomalies that could potentially comprise multiple objects, i.e., $|A_i| \geq 1$, in a spatially coherent region as outlined in Sect. 2.1. We classify these techniques based on the construction of the context for estimating anomalousness.
  - **Global:** Under the global context setup, models built across objects in a candidate anomalous region are compared against those built over objects across the entire dataset. In a weather dataset, a hot desert surrounded by warm plains would be regarded as an anomaly by these methods regardless of whether its temperature differs substantially from its local neighborhood. There has been a large body of work under this category from the statistics community, e.g., SaTScan [20] and its numerous variants; we will survey such methods as well as techniques from the data mining community that fall under this category.
  - **Local:** Techniques that use local context estimate the qualify of a candidate anomaly by comparing the model against that built over *nearby* objects, with the notion of nearness defined over spatial attributes. In our example, the hot

[1] We use the prefix *general* to differentiate these from spatial outlier detection methods, that we will see shortly.

desert wont qualify as an anomaly as long as the contrast in warmth is not good enough with the surrounding plain, regardless of whether it is much warmer than the average temperature in the whole area under consideration. Under this category, we will describe the LHA anomaly detection method [33] and also comment on how image segmentation techniques are applicable for local anomaly detection.

- **Grouping:** Unlike anomaly detection methods that specify a context for comparison crisply, methods under this category use an indirect form of context comparison for anomaly estimation. These techniques, largely inspired from literature on data clustering, group objects into clusters in such a way that the clusters are mutually divergent on the value attributes (as well as, sometimes, on the spatial attributes). Since the notion of mutual divergence implies that each cluster can be different from the models corresponding to its neighboring clusters as well, these methods are closer in spirit to the *local* methods than to the *global* methods discussed above. While some methods are only for forming groups and do not perform anomaly detection, they can be easily extended to the problem of anomaly detection by employing a test for anomalous-ness on the groups so that the subset of clusters that qualify be labeled as anomalies. There has been a lot of work, mostly from network intrusion detection literature, on clustering for anomaly detection; however, these, being designed for intrusion detection scenarios where there are no spatial considerations, treat all attributes on an equal footing without differentiating them as spatial or value attributes. We will survey such methods too, in addition to clustering methods specifically designed for spatial datasets such as those we are interested in.

In the following sections, we will describe techniques under each category in the taxonomy in a greater amount of detail.

3 Object Anomalies: Techniques for Outlier Detection

Outlier detection methods, as observed earlier, address the problem of finding individual objects in the dataset that are anomalous, i.e., by using the constraint:

$$\forall\ i, |A_i| = 1$$

This is a field that has attracted tremendous attention in the data mining community for the last two decades. Among the popular definitions of outliers is one from Hawkins [16]: “An outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism”. The vast majority of techniques that have been proposed for outlier detection consider the general case, where attributes are not classified into categories such as value and spatial; though not strictly the case, these may be regarded to consider data as purely spatial attributes (i.e., $m_v = 0$) since they make use of notions such as *neighborhood* and *density* that are more meaningful in spatial data.

Due to the large volume of literature around general outlier detection, we will take a closer look at those before considering methods for spatial outlier detection.

3.1 General Outlier Detection

The problem of outlier detection on general datasets, those where attributes are not classified into two classes as in the case of data that we have been considering in this chapter, has attracted plentiful attention from the data mining community, notably since [4]. Though these methods do not address our problem directly, they could be adapted to our problem of anomaly detection on spatial data. Given the large amount of work in general outlier detection, and the consequent maturity of techniques, we will consider those techniques in some detail in this section. We start with a framework for general outlier detection methods proposed in [28], and then take a closer look at LOF [4]. We will then comment on how general outlier detection methods can be adapted to the problem of outlier detection on spatial data. A comprehensive and excellent survey of outlier detection techniques appears in [28], for further reading on general outlier detection methods.

3.1.1 General Framework for Outlier Detection

The survey paper by Schubert et al. [28] proposes a five phase framework that covers most well-known methods for general outlier detection. The task may be seen as that of associating each data object d_i in the database with a score that is directly related to the estimated outlier-ness of the object; objects can then be sorted based on this score and those that pass a threshold of outlier-ness may be output. While the framework uses terms such as proximity and neighborhood as if the data has only spatial attributes, the methods are applicable for non-spatial attributes too. It may be noted that these methods handle all attributes equally and are not designed to handle spatial and non-spatial attributes differently. The five phase approach for outlier-ness estimation is given below:

- **Context Estimation:** The context of an object is typically defined as a subset of the data that are proximal to the object under consideration. The simplest and most widely used model for context is the collection of k nearest neighbors of the object in question. Another intuitive model is the set of objects that fall within a distance threshold. In certain cases, however, the entire database is used as the context.
- **Model Building:** This step summarizes the set of objects in the context of the object in question, to form an object-specific model. The model typically summarizes the distance of the considered object from objects in its context using a metric such as the sum of the distances. The simplest models could simply

count the number of objects in the context, whereas sophisticated ones try to fit a distribution such as Gaussian.

- **Reference Estimation:** While the context determines the objects that would feed into building a model, the reference of an object is a subset of the dataset that would be used for comparing the object's model against. Like the context, the reference could range from simply choosing the k nearest neighbors to considering the entire dataset.
- **Comparison:** This phase does the first part of the outlier-ness estimation wherein the model of the object is compared with the models of the objects in its reference. In the case of simple models like object count or sum of distances, the comparison could simply be the quotient of the object model with those of objects in the reference; these separate quotients (one for each object in the reference) may then be simply averaged to obtain a single score denoting outlier-ness.
- **Normalization:** The normalization step, which is not used in many methods, would normalize the object-specific outlier scores obtained from the previous step. A simple normalization approach would be to divide the outlier score of each object by the sum of scores across objects so that an outlier probability distribution is obtained across all objects in the dataset.

Having outlined this framework, any general outlier detection approach may be simply specified by the way each of these steps are instantiated. It may be noted that another smaller category of general outlier detection methods, called *global outlier detection*, do not fall into the above framework. Some of them regard objects that do not have at least a fixed fraction of database objects within a specified distance threshold as global outliers (e.g., [19]). Clustering-based outlier detection methods leverage clustering algorithms that do not force every data instance to be assigned to a cluster (e.g., DBSCAN [10], ROCK [15], RGC [2]); a simple heuristic, such as that used in FindOut [35] would be to demarcate objects that do not belong to any cluster as outliers. To contrast from such methods, the methods under the framework described above are some times referred to as *local outlier detection* methods. We will now turn our attention to LOF, an early method for local outlier detection.

3.1.2 LOF

LOF [4] is among the earliest methods for outlier detection. This draws upon ideas from density-based clustering approaches such as DBSCAN [10]. LOF uses the set of k nearest neighbors to each object as the context of the object. In the model building phase, LOF calculates a smoothened version of pairwise distance between each object d and every object c in its context; this is called the *reachability distance* and is calculated as follows:

$$RD_p(d, c \in context(d)) = max\{dist(d, p\text{th}NN(d)), dist(d, c)\} \tag{2}$$

where $p\text{th}NN(d)$ denotes the pth nearest neighbor of d based on a distance measure defined on the attributes. This is a smoothened version of the distance since objects far away from d would have their $RD_p(d, .)$ equal to their actual distance, whereas the pth$NN(d)$ would dominate in the case of objects closer to d. It may be noted that if $p = k$ (as is set in many cases), $RD_p(d, .)$ would be equal to the kNN distance for all objects in the context of d. The object-specific model used by LOF is called local reachability density that is defined as follows:

$$lrd(d) = 1 \bigg/ \frac{\sum_{c \in context(d)} RD_p(d, c)}{|context(d)|} \tag{3}$$

Thus, $lrd(d)$, the model for d, is the reciprocal of the average reachability distance of d to objects in its context. Moving on to the third phase in the framework, LOF uses the reference of an object as its k nearest neighbors itself, exactly as in the case of the context. In the model comparison phase, the model of d is compared to those built for objects in its reference space to obtain an outlier factor called the local outlier factor.

$$LOF(d) = average\left\{ \frac{lrd(r)}{lrd(d)} \middle| r \in reference(d) \right\} \tag{4}$$

LOF does not use a global normalization phase, and thus outputs objects with high LOF values as outliers. Informally, $lrd(.)$ captures the density of the space around an object, and those that are in a sparser neighborhood (i.e., with high distance to nearest neighbors) as compared to the neighborhoods of the objects in their context would end up with large LOF values and thus be considered outliers.

3.1.3 Adapting for Spatial Outlier Detection

The task of spatial outlier detection is that of finding objects whose values for the value attributes contrast well with those in its specified spatial context. We could use general outlier detection methods for anomaly detection on spatial data by using the following approach:

1. *Partitioning:* Partition the dataset based on solely the value attributes such that the objects within each partition are reasonably homogeneous with respect to the *value attributes*. This could be achieved by clustering the dataset on the value attributes using any well-known partitional clustering algorithm [5], so that objects within each cluster has similar values for the value attributes. In simpler cases of datasets with just one value attribute such as temperature, one could create buckets on the range of values of the attribute to create partitions; each partition would then comprise the subset of data that have objects having temperature within the range of the respective bucket.
2. *Partition-wise Outlier Detection:* In this phase, each partition is *separately* fed to the general outlier detection methods which will then work on finding outliers based on only the *spatial attributes*. These would then identify partition-specific outliers as those objects whose neighborhood within that partition is sparse.

> Given the nature of our partitioning step, the sparse neighborhood of the outlier is likely to be caused due to many objects in close proximity to the identified outlier having been assigned to other partitions. Thus, this satisfies our condition of spatial outliers as those that contrast well with local neighborhood on the value attributes. It needs to be noted, however, that there could be false positives under this approach. The sparse neighborhood of the partition-specific outlier could be caused also due to the general sparsity of the full dataset (i.e., across partitions) as against the objects in the neighborhood being assigned to other partitions; a post-processing step would need to be designed to weed out cases where the partition-specific outlier is induced by the former factor than the latter. It may also be noted that this phase treats all objects in a partition as if they are identical on the value attributes; this could lead to missing some legitimate outliers.

Though adaptations of general outlier detection to spatial data have not been considered much, a recent approach adapting LOF to spatial data [28] is worthy of attention.

3.2 *Spatial Outlier Detection*

Spatial outlier detection methods, as observed earlier, look to identify objects whose values for the value attributes contrast well with those in its specified spatial context. The common approach used by spatial outlier detection methods is to estimate a deviation score, where the value attributes of an object are compared to the value attributes of its spatial neighbors [30], with the spatial neighbors forming the context. Among the relatively few methods that have been proposed for the same, we will look at SLOM [7] in this section.

3.2.1 SLOM

SLOM [7] starts by defining a context for each object that would be used for comparison later on. The context of each object in SLOM is simply the set of spatial neighbors in gridded or tessellated data; in the case of a non-gridded dataset of multi-dimensional point objects, the context may be defined as the set of *kNN* objects. SLOM then calculates the trimmed mean for the object, which is its average distance with those on its context, distance measured on the value attributes only; the context object with the largest distance is ignored to reduce sensitivity to noise, thus giving this measure the adjective *trimmed*. In addition to calculating the trimmed mean, denoted by $\tilde{d}$, of the object itself, the trimmed mean is also calculated for all objects in the context, and its average is estimated:

$$avgtm(d) = average\{\tilde{d}(c) | c \in \{context(d) \cup d\}\} \tag{5}$$

This is used to define an oscillation measure β that estimates the extent of fluctuation of the trimmed mean around the object in question.

$$\beta(d) = \frac{|\{c \in econtext(d) \wedge \tilde{d}(c) > avgtm(d)\}| - |\{c \in econtext(d) \wedge \tilde{d}(c) < avgtm(d)\}|}{|econtext(d) - 2|} \tag{6}$$

where $econtext(d)$ indicates the set of objects in $context(d)$ and itself. In essence, β captures the normalized asymmetry of the distribution of the trimmed means around $avgtm(d)$; if there are equal number of objects on either side, β evaluates to 0. The SLOM measure for an object is then defined as follows:

$$SLOM(d) = \frac{\tilde{d}(d) \times \beta(d)}{1 + avg\{\tilde{d}(c)|c \in context(d)\}} \tag{7}$$

Thus, objects with a large $\tilde{d}(.)$ score with respect to those in their context would then be good SLOM outliers. As seen from the construction, higher β values also favor labeling an object as an outlier. To position SLOM on the framework for general outlier detection introduced earlier, estimation of $\tilde{d}(d)$ may be seen as the model building phase. The reference is the same neighborhood as used for the context, and the estimation of β followed by the SLOM score form part of the comparison phase. As in the case of LOF, there is no global normalization in SLOM as well.

4 Region Anomalies: Global

Turning our attention to anomalous region detection techniques where the sets of anomalies output could comprise more than one point, we will now look at methods that quantify anomalous-ness by contrasting the behavior of the set of objects in question against a global behavior model estimated across all the objects in the dataset. Such methods may be classified into two groups: the larger group is that consisting of statistical methods that build upon scan statistics such as SaTScan [20], whereas methods in the second group explore data mining approaches for the task. We will look at these two kinds of techniques in separate sub-sections herein.

4.1 Statistical Approaches: Spatial Scan Statistics

Among the earliest statistical techniques for globally divergent anomaly detection is the spatial scan statistic proposed by Kulldorff [20]. Numerous extensions have been proposed to extend the basic spatial scan statistic while preserving the core framework. We will consider the basic spatial scan statistic as well as one of its extensions, in this section.

While the basic spatial scan statistic is general and can be applied across domains, we illustrate it by means of an example application domain. Consider the problem of identifying spatial clusters with high incidence of a particular disease that is under study; these clusters would be anomalies due to the disparity between the cluster disease rate and the global rate. With coarse-grained modeling where we record statistics at the county level, each data object d_i could be county, with its population denoted by a_i, and the number of diseased people denoted by y_i. The core parameter of interest is the disease rate (or more generally, the *response rate*),

$$r_i = \frac{y_i}{a_i} \tag{8}$$

Due to this construction, response rate varies within the range [0, 1]. When the cell is denoted using a variable such as d, we will use r_d to denote the response rate in the cell. The spatial scan statistic is designed to identify *hotspots* (i.e., anomalies) that are spatial regions (e.g., county) with an elevated (or decreased) response rate when assessed against the rate in the entire dataset. These techniques usually do not allow to identify irregular closed shapes such as county boundaries, but restrict themselves to the identification of regular-shaped spatial anomalies (e.g., circular regions). Hotspots are regions of cells that satisfy at least the following two properties:

- **Spatial Coherence:** The initial version of the spatial scan statistic enforced that hotspots be circular in shape, i.e., a hotspot be a region defined by a circle, intuitively comprising of all cells that fall within the circle. In the case of non-gridded data where each observation is a point object, all objects that fall within the circle would then be considered. There have been extensions that have allowed for elliptically shaped [21] hotspots, and even arbitrarily shaped regions that are connected [25].
- **Limited Size:** Hotspots should not be excessively large with respect to the entire dataset. For example, if the region were to be more than half the size of the dataset, it could be argued that the region, and not its exterior, should be considered as the background. The initial version of the spatial scan statistics, and most of its variants, impose a constraint that a hotspot not be more than 50 % of the size of the entire dataset.

In addition to the above properties that are agnostic to the response rate, we would want to ensure that the response rate within the candidate hotspot be high (low) before it be declared as a hotspot. The spatial scan statistic uses a hypothesis testing approach to assess this hotness. The null hypothesis assumes that the response rate is the same for all regions in the dataset, and thus, consequently, that there is no hotspot in the dataset:

$$H_0 : \forall d \in \mathscr{D}, r_d = p_a \tag{9}$$

The alternative hypothesis, on the other hand, assumes that there is a hotspot in the dataset, and is outlined as follows:

$$\begin{aligned} H_1 : \exists p_0, p_1, Z \ &\text{such that} \\ 0 \leq p_0 &< p_1 \leq 1 \\ Z \subset \mathscr{D} &\wedge Z \neq \phi \\ \forall d \in Z, &r_d = p_1 \\ \forall d \in (\mathscr{D} - Z), &r_d = p_0 \end{aligned} \tag{10}$$

Informally, H_1 assumes that there is a region Z such that the response rate is a high value, p_1, for cells within it, and is a low value, p_0, for other cells in the dataset $\mathscr{D}$. In addition to the response rate conditions as outlined above, the alternative hypothesis also requires that Z, the hotspot region, adheres to the spatial coherence and size restrictions as outlined earlier. H_1, as outlined above, assumes that the hotspot exhibits an elevated response rate; the analogous hypothesis for reduced response rate hotspots may be obtained by simply inverting the inequality between p_0 and p_1 so that p_1 be lesser than p_0.

As may be seen, there are three unknowns in the alternative (i.e., hotspot) hypothesis, viz., Z, p_0, and p_1. When all of these are known, one can estimate the likelihood of the data being generated under the hotspot hypothesis, by assuming that the data is distributed according to one of the well-known models such as Bernoulli or Poisson distributions. Let the likelihood of the combination of parameters be denoted by $L(Z, p_0, p_1)$; now, the problem is to find values $\hat{Z}, \widehat{p_0}, \widehat{p_1}$ that maximize the likelihood $L(Z, p_0, p_1)$ for a given dataset. For a given value of Z and an assumed distribution of data, it is easy to determine the values of p_0 and p_1 that maximize the likelihood, as follows:

$$\{\widehat{p_{0Z}}, \widehat{p_{1Z}}\} = \underset{p_0, p_1}{\arg\max}\, L(Z, p_0, p_1) \tag{11}$$

where $\widehat{p_{iZ}}$ denotes the optimal value of the parameter for a fixed Z. The difficult part, however, is estimating the value of $\hat{Z}$ that maximizes $L(.)$ since the space of possible values of Z, $Dom(Z)$ is typically large. The optimization problem we are talking about is the following:

$$\hat{Z} = \underset{Z \in Dom(Z)}{\arg\max}\, L(Z, \widehat{p_{0Z}}, \widehat{p_{1Z}}) \tag{12}$$

For example, any circular region in the dataset that satisfies the size restriction is a candidate value of Z, and it is not difficult to see that there could be infinitely many circles. Strategies to do this optimization fall under two main categories:

- **Parameter Space Reduction:** Instead of looking for all candidate values across $Dom(Z)$, parameter space reduction strategies look to find optimal values of Z

within a sub-space $R(Z)$. The reduced space may be defined by a tessellation of the space followed by a topological sort, as used in ULS Scan [25]. However, this approach could potentially lead to sub-optimal result since $\hat{Z}$ may not always be contained within the search space $R(Z)$.

- **Stochastic Optimization Methods:** Stochastic optimization methods such as genetic algorithms and simulated annealing start from an initialization of Z, and scan the space through perturbations of the value of Z. These could miss $\hat{Z}$ and discover a sub-optimal value, but, are expected to converge to the optima under certain assumptions. An example of a simulated annealing approach for spatial cluster detection appears in the technique presented in [9].

It is not just enough to have a high likelihood under H_1; we would need to compare with the likelihood under the competing hypothesis, H_0. Having determined the $\hat{Z}$ that maximizes the likelihood for H_1, the spatial scan statistic uses a likelihood ratio test to determine the number of times the data is more likely under H_1 as compared to the null hypothesis, H_0. To do this, we would need to also estimate the likelihood of the data under the null hypothesis. It may be observed that H_0 is the limit of H_1 as $p_1 \rightarrow p_0$. Thus, the likelihood under the null hypothesis may simply be written as $L(Z, p_a, p_a)$. Further, it is easy to observe that Z is not identifiable in the limit[2]; any value of Z would serve just as fine since the response rate within and outside are assumed to be equal. Thus, we can omit Z and simply denote the likelihood under the null hypothesis as L_0, a shorthand for $L(Z, p_a, p_a)$. The likelihood ratio is then:

$$\lambda = \frac{L(\hat{Z})}{L_0} \tag{13}$$

We have omitted the parameters p_0 and p_1 from the numerator for simplicity. Beyond just estimating the value of the test statistic, λ, we additionally would like to see whether it is high enough under an expected distribution of the test statistic. For example, even a synthetic dataset of $|\mathscr{D}|$ objects generated under the null hypothesis may have a yield a high value of λ due to localized regions that have a high response rate. Monte Carlo simulation methods [34] may now be used to generate a large number of synthetic datasets, each of $|\mathscr{D}|$ objects and overall response rate (i.e., p_a) equal to the dataset under consideration. Consider 10,000 such generated datasets, each of which are subjected to the same analysis to yield a separate λ value; now, if the λ value for the dataset in question (i.e., the non-synthetic dataset) is higher than 9500 of the 10,000 generated λ values, the test statistic may be adjudged to be significant at a p-value of <0.05. Similarly, if the test statistic is within the range of the top 100 values, the test statistic may be deemed to be significant at a p-value of <0.01. If the hotspot hypothesis is indeed

[2] An intuitive likelihood estimates the chances of generating the data points as against the expected probability, and aggregates it across the objects. Under the condition that the expected probability is the same for objects within and outside Z, any value of Z would yield the same likelihood.

statistically significant, the corresponding region, $\hat{Z}$, is regarded as a hotspot or an anomaly. There could be many such regions—i.e., values of Z—in the dataset that yield a statistically significant λ; in cases where one would like to identify multiple hotspots, the likelihood ratio provides an intuitive ranking.

4.1.1 ULS Scan Statistic

Having described the basic approach, which is also the core framework that most spatial scan statistics variants operate upon, we will now look at the upper level set scan statistic [25] which is interesting due to being able to detect arbitrarily shaped anomalies (i.e., hotspots). Most of the details of the likelihood and test statistic computation remain similar as in the spatial scan statistic; we will focus on the construction of the candidate hotspots and computational aspects therein, which are interesting from a data mining perspective. Spatial scan statistics that restrict hotspots to be of regular shapes (e.g., circular and elliptic) would be unable to detect clusters in geographic regions that are culturally and politically coherent while being rather elongated geographically such as a settlement on a valley. Many epidemiological patterns are generated due to policies determined at a country/province level, neither of which need to be circular or regular or even convex in shape. The ULS Scan Statistic may be applied in such scenarios.

ULS Scan Statistic works on datasets in a tessellated space where the presence or absence of a shared boundary between regions is well-defined. ULS can admit hotspots that are composed of multiple connected cells, i.e., a set of cells such that there exists a pairwise path between any two cells that is entirely confined to within cells in the set. The ULS Scan Statistic works by first constructing a data structure called the ULS Tree that is composed of cells as nodes, and the edges determined using a combination of connectedness and response rates.

Figure 3 illustrates the working of the ULS on a sample dataset and the construction of the ULS Tree. Figure 3a shows a tessellated dataset with different regions colored with an intensity that is proportional to their response rate. The cells are numbered and the corresponding response rates are seen to satisfy the following:

$$r_1 < r_3 < r_2 < r_5 < r_4 \tag{14}$$

The y-co-ordinate of each cell in Fig. 3b is equivalent to the response rate; thus, the cell 4 appears at the top whereas cell 1 appears at the bottom. The x-co-ordinate in the same figure does not have any meaning beyond visualization ease, and cells could be arbitrarily moved around on the X-axis. The construction of the ULS Tree starts from the top and proceeds downwards, and follows the following simple rules:

- If the next cell (the cell with the highest response rate among those yet to be considered) is not spatially adjacent to any higher cell, do not introduce any edges. Since this step allows to add nodes that are not connected to any existing

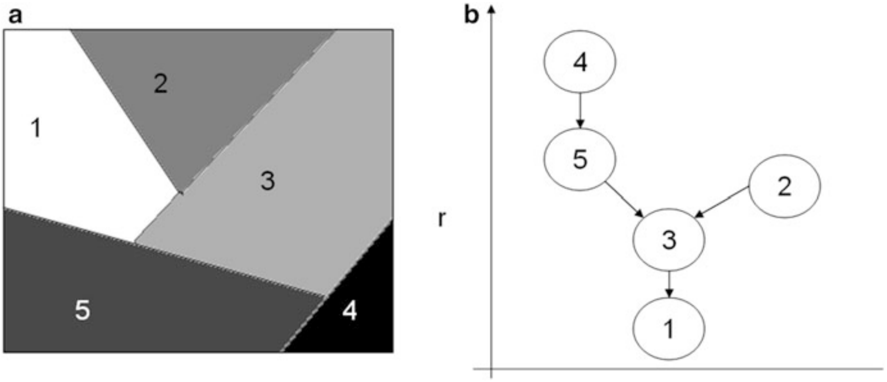

Fig. 3 Illustration of ULS Tree construction (**a**) dataset, (**b**) ULS Tree

node, it allows to create multiple components that are mutually disconnected, across steps; we will refer to each such component as a tree.

- Else, in the case that the next cell is spatially adjacent to a higher cell, add an edge from the lowest node in the tree containing the latter cell. If there are multiple trees containing cells that are spatially adjacent to the next cell, add edges from each of them.

This construction process ensures a unique entry point for each cell in the tree, as may be seen in the example below.

Tree Construction in Our Example: In our example, 4 starts off as a singleton node. The next cell, 5, is seen to be adjacent to 4, and this induces the link between them. The third cell, 2, is not adjacent to any cell considered so far, and thus starts another tree on its own. When the algorithm processes 3, it is seen to be spatially adjacent to 4, 5, and 2. 4 and 5 appear in the same sub-tree necessitating just one edge for both, whereas an additional edge is introduced to connect to 2. 1 is then added as the lowermost cell, completing the tree construction process.

Identifying hotspots under ULS Scan: Each node in the ULS Tree defines a candidate region that comprises all cells above it in its sub-tree. Thus, the region at node 5 is to be understood to stand for the set of cells $\{4, 5\}$, whereas the set at 3 is $\{4, 5, 2, 3\}$. The tree construction method ensures that each such region is a set of connected cells. As is obvious, the bottom-most node is the set of all cells in the dataset; this would be excluded from consideration as a hotspot due to the size constraint that excludes regions that cover more than 50 % of the dataset. It is likely, but not necessary, that the hotspots with high likelihood scores would be towards the upper part of the tree. All candidate regions are then processed using the statistical machinery from that of the spatial scan statistic introduced earlier. ULS Scan is computationally efficient since the number of candidate hotspots (i.e., nodes in the tree) is intuitively bounded by the number of cells in the dataset.

4.1.2 Other Extensions

As observed earlier, there have been numerous extensions of the basic spatial scan statistic. The Bayesian spatial scan statistic [24] proposes a Bayesian method that is shown to be easier to incorporate prior information and can provide much better response time by avoiding the randomization testing phase. Among the most active areas of work have been to relax the shape restriction of hotspots in the original framework to ellipses [21], and arbitrary shapes [9, 32]. Readers interested in scan statistics may find the book on scan statistics [13] useful.

4.2 Mining Approaches

Unlike the statistics community, the data mining community has paid lesser attention to the problem of identifying globally divergent anomalies. The two intuitive directions for the mining community towards this problem are top-down or bottom-up. Top-down approaches could start with the entire dataset and progressively zoom-in on candidates for anomalies, whereas bottom-up approaches could start with small anomalous regions and merge regions while keeping track of how they fare with respect to anomalous-ness. We will look at Bump Hunting [12], a top-down approach, in this section.

4.2.1 Bump Hunting

Bump Hunting [12] is a top-down approach for spatial cluster detection designed for datasets with only one value attribute. It starts with the entire spatial dataset in one spatial *box*, B. The algorithm iteratively peels of spatial sub-boxes from B, in accordance with the following greedy strategy:

$$B = B - \arg\max_{b \in C(b)} average\{v_{i1} | d_i \in B - b\} \tag{15}$$

$C(b)$ denotes the set of spatial sub-boxes that are available to be peeled off. v_{i1} is the only value attribute of the object d_i. Thus, this strategy peels off the spatial sub-box such that the average value of the value attribute among objects in the remainder of B is maximized. The peeling process stops when the number of objects in B is small enough that further peeling would violate a support threshold. To partially offset for the sub-optimality in B due to the greedy strategy employed, an additional phase of pasting is applied that would enlarge the box by pasting sub-boxes to it, using considerations similar to that employed in the peeling phase.

As is obvious from the description, this addresses a special case of globally divergent anomaly detection. The approach would need to be adapted to address the problem of detecting multiple anomalies. Extending the technique to work with datasets having multiple value attributes is possible, but is not straightforward. As may be inferred from the description, the basic approach can only identify anomalies where the average value of the value attribute is higher than the average value in the dataset; similar deviations to the lower side of the average may also be deemed anomalous and can be identified by changing the arg max condition to arg min.

5 Region Anomalies: Local

We now consider techniques addressing the problem of discovering anomalies that exhibit differences from their local neighborhood. We will first describe a recent data mining algorithm which identifies homogeneous regions that contrast well with their generalized local neighborhood as anomalies. This will be followed by a discussion on how the task of image segmentation relates to the problem.

5.1 Localized Homogeneous Anomalies

LHA detection [33] addresses the problem of discovering regions that are spatially coherent and homogeneous on the value attributes, while also contrasting well on the value attributes with their *generalized local neighborhood*. This is designed to work on gridded or tessellated data, where the adjacency/neighborhood relation is well-defined. Homogeneity is measured on the value attributes; thus, a region with people of similar incomes would be considered homogeneous, while a city with luxury residences interspersed with slums would not satisfy the homogeneity constraint on the income attribute. The generalized neighborhood condition is best illustrated by means of the example in Fig. 4 which depicts two plots of spatial datasets that use a single value attribute, with each point colored according to the intensity of the value attribute (e.g., temperature) at them. In each, the middle circle is the region under consideration and the ring-shaped enclosing region bounded by the dotted lines form the local neighborhood. The circle in Fig. 4a is not considered as an LHA due to being of medium intensity, which would not differ much from the average intensity of its neighborhood (which is composed of two halves of low and high intensity). Telang et al. [33] propose that it be considered a non-anomalous *transitional* region of intermediate values between the low values in the left and the high values in the right. On the other hand, Fig. 4b is clearly anomalous since its left and right neighborhoods contain low and intermediate values, respectively, both contrasting well with the high values in the central circle. Thus, despite both the circles contrasting well with their local neighborhoods separately, the one in (a) does not differ much from the generalized neighborhood since any measure of central tendency (recall response rate in scan statistics) estimated over the objects in the neighborhood would be quite close to the values within the circle.

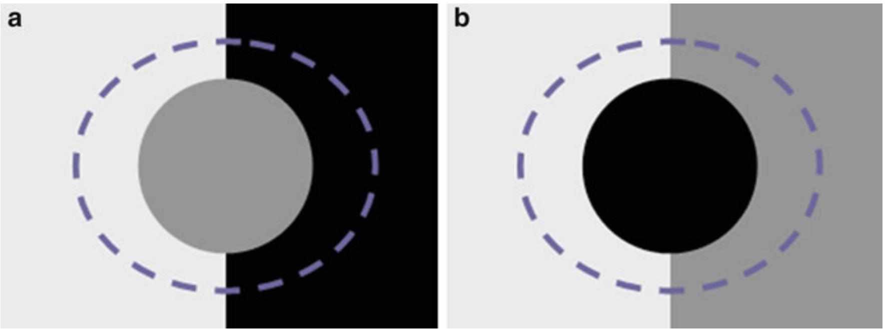

Fig. 4 Generalized neighborhood example (**a**) transition, (**b**) anomaly

The algorithm proceeds in two phases, discovering homogeneous clusters in the first phase, followed by a phase where non-anomalous clusters are filtered out. The cluster detection algorithm starts by initializing all cells as unclustered, followed by picking up each unclustered cell and trying to grow a cluster from them. A set of cells, C, is grown iteratively by greedily adding cells that keep the gini co-efficient,[3] *measured on the value attributes*, as low as possible.

$$C = C \cup \operatorname*{arg\,min}_{c \in C.neighbors} Gini\{C \cup \{c\}\} \tag{16}$$

Each cluster is grown as long as there are neighbors whose inclusion would retain the gini co-efficient to within a threshold; if no such neighbors exist, the cluster is marked as completed, and the algorithm proceeds to the next unclustered cell. In the second phase, each cluster C is compared against its generalized neighborhood where the neighborhood is defined as a set of cells:

$$N_C = \{d | d \in \mathscr{D} : \exists d' \in C, dist(d, d') \leq \rho\} \tag{17}$$

Informally, the neighborhood of C comprises all cells that are at-most ρ hops away from any cell in C; thus, the neighborhood of a circular cluster would be a ρ-width ring just enclosing the cluster. Conceptually, the neighborhood forms a set of cells that form a sheet of width ρ around the cluster C. Each cluster is then assessed for anomalous-ness using the likelihood ratio test on the value attributes under the following hypotheses:

$$H_0 : \forall d \in C \cup N_C, r_d = p_a \tag{18}$$

$$H_1 : \exists p_0, p_1 \text{ such that} \tag{19}$$

[3] http://en.wikipedia.org/wiki/Gini_coefficient.

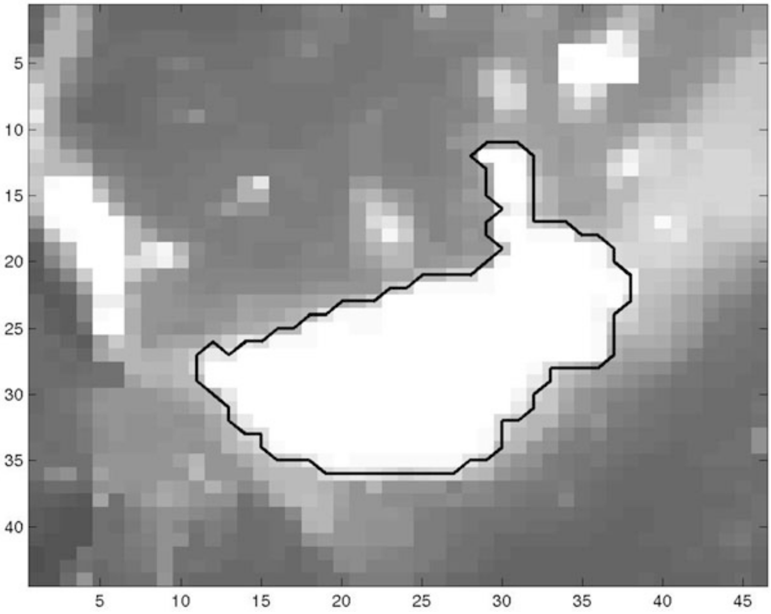

Fig. 5 Example LHA anomaly

$$\begin{aligned}
&\forall d \in C, r_d = p_1\\
&\forall d \in N_C, r_d = p_0\\
&0 \le p_0, p_1 \le 1\\
&p_0 \ne p_1
\end{aligned}$$

Basically, H_1 proposes that the response rate is different between the interior and exterior of C whereas H_0 assumes uniform response rate across the dataset. The response rate r_d above may be replaced by the average temperature in scenarios such as climate modeling where the value attribute is the temperature. The likelihood ratio, the number of times the data is likely under H_1 as compared to H_0, is compared against the χ^2 value to a desired statistical significance. All clusters that qualify are output as anomalies, whereas the others are discarded. Figure 5 illustrates, using a black border line, an anomaly detected using the LHA identification approach.

5.2 Image Segmentation

We will now consider the applicability of image segmentation methods towards the problem of identifying region anomalies that contrast with their local neighborhood. Image segmentation is the task of partitioning a digital image into multiple segments, i.e., sets of pixels, such that pixels within each set share certain characteristics. Since the segments are typically expected to stand for objects, pixels within each segment are usually expected to be connected and to have similar colors. To exemplify the connection of such methods to our problem, consider the pixels to be akin to data points in a spatial dataset, where the positioning of the pixel is determined by the spatial attributes and the color of the pixel being determined by the value attributes. In the case of an image, there are typically only two spatial attributes and three value attributes (i.e., RGB values); however, most image segmentation methods can be easily adapted to deal with more or fewer spatial and value attributes.

To illustrate the working of simple methods for image processing, consider the example in Fig. 1 again. It is not hard to see that the simplest possible cluster (i.e., anomaly) membership propagation algorithm that could identify the anomaly could be similar to flood-fill [29]. Under flood-fill, the cluster is grown on all directions, with the growth restricted such that boundaries on value attributes be not crossed. This criteria of stopping the cluster growth upon seeing a significant difference in value attributes at the frontier implicitly ensures that the cluster contrasts with the local neighborhood on the value attributes. In simple painting tools, the boundary would be a cell that has a different color from that of the interior of the cluster; simplistically, image segmentation algorithms such as blob detection may be thought of as a relaxation of the boundary condition of flood-fill (this is indeed an over-simplification). In our case, the boundary detection piece could look for deviation on the value attributes (i.e., temperature in the case of the example in Fig. 1). Advanced image segmentation methods such as blob detection techniques use a variety of methods such as histograms [3], graph partitioning [14, 31], and region growing [11, 27].

Differences from LHA Despite being a locally contrasting method, clusters discovered by image segmentation methods have striking differences from LHA anomalies. LHA anomalies classify regions as anomalies based on whether they differ enough from the *generalized* local neighborhood. On the other hand, image segmentation-style approaches do not differentiate between clusters that differ from the local neighborhood in piecewise fashion and those that differ from the generalized local neighborhood. Thus, the central circle in Fig. 4 that would be discarded as a transition region by LHA would be identified as an anomaly by an image segmentation-style method.

Adaptations Though image segmentation methods are probably the most mature family of techniques from another domain that can handle spatial and value attributes differently and in accordance with the semantics as necessary for our problem, they need to be adapted to be used for anomaly detection meaningfully. We list some possible adaptations:

- **Data Sparsity:** Unlike the case of digital images where every possible combination of (x, y) values would be taken by a pixel, spatial data are often sparse. For example, in the case of disease or weather modeling, we need to model data points corresponding to humans or weather sensors, respectively; intuitively, these objects are far fewer in number as compared to the number of possible

combinations of [*lat*, *long*] values. One simple way of handling this case is to coarsen the space by gridding it so that each grid cell has data points within it, with the value attributes of the grid cell estimated as the mean of the value attributes of the data points within it. The image segmentation technique can then be applied on the dataset comprising cells of the grid. This introduces a parameter, i.e., the width of the grid cell. Devising better ways of transforming sparse data so that image segmentation methods could be applied would be an interesting direction for future work.
- **Anomaly Sizes:** Image segmentation methods, when applied on Fig. 1, could produce the entire white region as a segment. Similarly, they could identify even small regions with very few cells as objects too. There are intuitive reasons to exclude very large as well as very small anomalies. For example, if the candidate anomaly is larger than half of the dataset, it might be argued that the background, and not the region, be considered an anomaly; the 50 % size constraint in scan statistics is motivated by such concerns. A lower bound on the anomaly size might be necessary to avoid a lot of very small and non-noteworthy anomalies in the output.
- **Scoring:** The segments generated out of the image segmentation methods do not have an intuitive scoring. For scenarios where a user, due to time or screen size constraints, is interested in only the top-k anomalies, there needs to be a mechanism to rank the segments based on their anomalous-ness. An intuitive idea would be to score segments based on the difference in value attributes between objects in the segment and those in their immediate local neighborhood.

6 Region Anomalies: Grouping

Clustering is probably the most widely studied task in unsupervised learning. Given that clustering targets to find groups of data that have similar behavior, it is only obvious that clustering techniques be adapted to the problem of finding anomalous, i.e., out-of-the-common behavior. General clustering algorithms target to partition the data into clusters so that the intra-cluster object similarity is maximized while the inter-cluster object similarity is minimized. There have been relatively few efforts to adapting clustering to form clusters in spatial data, i.e., to form clusters as regions that are spatially coherent while differing from other regions based on the value attributes of the component objects. However, using clustering to address the problem of finding outliers in general data, i.e., where the attributes are not differentiated into value and spatial, has been the subject of much study; most efforts have been motivated from use cases in network intrusion detection. In this section, we will look at two classes of techniques and possible adaptations to our problem:

- Clustering methods for spatial data and how they can be adapted to identify anomalies.
- Clustering-inspired anomaly detection methods on general data, and their suitability to anomaly detection on spatial data.

In particular, we do not consider how clustering methods for general data can be used for the task of anomaly detection on spatial data in this section.

6.1 Clustering for Spatial Data

We will now consider techniques for clustering spatial data in a way that the clusters are *spatially contiguous* while being coherent on the value attributes. Thus, these methods use different considerations for spatial and value attributes. While contiguity is typically correlated with coherence (i.e., similarity), these considerations are markedly different in semantics. As an example, no matter how similar an object is, to a cluster, on the value attributes, it cannot be part of the cluster unless it is spatially contiguous with the cluster; no such membership constraint exists on the value attributes. The clustering techniques that we discuss in this section output clusters from the data. Depending on the usage scenario, the output clusters may need to be filtered using constraints on size, homogeneity, or contrast, in order to find a subset of clusters that would be regarded as anomalies. In this section, we consider two methods for clustering spatial data.

6.1.1 HAC-A

This is a simple adaptation of hierarchical agglomerative clustering [36], proposed in [33] as a technique to identify spatially contiguous clusters. Hierarchical agglomerative clustering works by initially assigning all objects to singleton clusters, followed by merging the most similar pair of clusters iteratively, until only the desired number of clusters remains. In HAC-A, the search for pairs of clusters is directed to find only such pairs that are spatially adjacent:

$$PairToMerge = \operatorname*{arg\,max}_{c_1, c_2 \in Clusters, isAdjacent(c_1, c_2) = true} Similarity(c_1, c_2) \tag{20}$$

Similarity is assessed using the value attributes, while the *isAdjacent*(., .) relation is determined using the spatial attributes. This, as may be obvious, is applicable to scenarios where the spatial adjacency relation is well-defined, such as tessellated or gridded data. Extending this to work with datasets of multi-dimensional points such as sensors distributed across a spatial region could lead to interesting future work.

6.1.2 Clustering Ensuring Spatial Convexity

Deepak et al. [8] proposes a method that treats spatial and value attributes differently in clustering. In particular, it exploits the property of the K-Means algorithm [22] in generating clusters with spatially contiguous convex shapes. The method operates in two phases as below:

- **All Attribute Clustering:** In this phase, any clustering algorithm may be used to cluster the data objects across all attributes, without differentiating between spatial and value attributes.
- **K-Means:** The means of the clusters from the first phase are considered, and these are then projected to the spatial attributes alone (i.e., the value attributes are excluded from the cluster means). Now, each object is assigned to the closest mean, similarity measured on just the spatial attributes:

$$cluster(d_i) = \arg\max_{c \in clusters} sim(\pi(d_i), \pi(mean(c))) \tag{21}$$

where $\pi(.)$ denotes the projection on the spatial attributes.

The second phase ensures that the clusters are spatially contiguous; this is due to the property of the K-Means assignment in generating disjoint and convex shaped clusters. This is in contrast with the first phase clustering that assigns data objects to clusters based on their similarity on all attributes. However, the second phase can significantly alter the cluster memberships from that obtained in the first phase. Evidently, there is room for future work to build a principled technique where convexity and homogeneity conditions are considered together.

6.2 Clustering-Based Anomaly Detection

This section covers methods for clustering-based anomaly detection for general data where attributes are not differentiated as value and spatial attributes. Similar to the structure of Sect. 3.1 on general outlier detection, we will describe methods without considering how they can be adapted to spatial data. We will then comment on how they can be adapted to our problem of anomaly detection on spatial data.

Most techniques for clustering-based anomaly detection have been designed keeping in mind the problem of intrusion detection in networks. It is easy to think of these methods as being instantiations of a two-phase framework as follows:

- **Clustering:** Cluster the data so that the data is partitioned into multiple groups.
- **Anomaly Scoring:** Use statistical measures based on the clustering produced in the first step to identify anomalous groups of data.

The various algorithms for clustering-based anomaly detection differ in the kind of clustering process in the first step, and in the way in which clusters are scored as anomalous in the second step. We will look at a few of them now.

CLAD [23] is an approach for clustering-based anomaly detection. In the first phase, it uses fixed-width clustering to cluster the objects in the dataset to form clusters. CLAD allows an object to be a member of multiple clusters. Fixed-width clustering, which has been quite popular in network intrusion detection, works by traversing the dataset object by object. It starts by setting the first object to the center of a cluster. Every subsequent object would then be added to those existing clusters whose centers are at most w unit of distance away from it; if there are no such clusters, a new cluster is formed with the object in question as the center. As a final phase, CLAD reconsiders every object and adds it to all clusters whose centers are not more than w distance away. In the second phase, every cluster is assigned an inter-cluster distance (ICD) with the ICD for cluster c defined as:

$$ICD_c = \frac{1}{|C|-1} \sum_{c' \in C, c \neq c'} distance(c, c') \tag{22}$$

where C is the set of clusters and $distance(.,.)$ is a distance function between clusters. CLAD considers clusters that are distant and either sparse or dense, as anomalous. Thus, a cluster c is flagged as anomalous if both the below conditions hold:

- If ICD_c is more than one standard deviation away from the ICDs of clusters in C.
- Number of objects in c is numerically away from the average number of objects across clusters in C by at least one median absolute deviation.

This completes the anomaly detection process in CLAD.

Another approach, proposed in [18], uses an adaptation of K-Means clustering in the first phase. In the second phase, a graph is formed with clusters as nodes and distance between cluster centers acting as the edge weight. A minimal spanning tree is built out of this graph; the longest edge is then removed from the spanning tree, and the clusters in the smaller tree are labeled as outliers. He et al. [17] also uses a two-phase method, where any clustering method is used to identify clusters. This is followed by determining a threshold such that clusters having more and less objects than the threshold are called large and small clusters, respectively. Each object d is then scored as follows:

$$CBLOF(d) = \begin{cases} |d.cluster| \times dist(d, d.cluster) & if\ d.cluster\ is\ small \\ |d.cluster| \times min\{dist(d,l)|l \in large\ clusters\} & if\ d.cluster\ is\ large \end{cases} \tag{23}$$

where $d.cluster$ denotes the cluster to which d belongs. $CBLOF$ functions as an outlier-ness score, and tends to be high for ds that is far away from the center of low-cardinality clusters. Though this score is assigned at an object level, and is hence more close to outlier detection, whole clusters could be marked as anomalous by virtue of their small size. Yet another approach [26] that uses clustering uses a scoring that prefers to label small clusters as anomalous. KD-trees have been used instead of clustering in an anomaly detection method proposed in [6].

6.2.1 Adapting for Anomaly Detection on Spatial Data

Similar to the discussion in Sect. 3.1.3, a straightforward method of using general clustering-based anomaly detection methods for spatial data would be to partition data using value attributes, followed by feeding them separately to the anomaly detection method with just the spatial attributes. Since the techniques targeting network intrusion that we saw in the previous section mostly employ clustering algorithms that produce contiguous convex clusters in the attribute space they work with, using them on data with just spatial attributes would produce spatially contiguous clusters (a desirable property). The partitioning approach, however, could pose more serious issues for region anomaly detection than outlier detection since region anomalies comprise multiple objects. Consider the case of anomalies that straddle the value attributes-based partitioning boundary; these could go undetected under the partitioning-based adaptation. Partitioning data using fuzzy clustering methods could be an interesting direction to pursue to get to a more principled approach to leverage clustering-based anomaly detection for our task.

7 Discussion

We have so far seen many methods for anomaly detection on spatial data. Additionally, we also considered applications of methods proposed for general data (i.e., where attributes are not classified into value and spatial) such as outlier detection and clustering-based anomaly detection. Another potential avenue for consideration is that of usage of clustering algorithms for general data, for anomaly detection in spatial data.

The various clustering approaches differ in how they operate; while density-based clustering algorithms (as well as fixed-width clustering as seen in Sect. 6.2) use a bottom-up style approach by aggregating data objects to form clusters, other popular algorithms like K-Means [22] use a more global strategy by assigning data points to clusters based on their proximity to various cluster models. The former class of methods, especially, DBSCAN [10] and variants such as OPTICS [1] have been shown to be useful for detecting patterns in spatial data. One may recollect that many of the region anomaly methods do not have a very sophisticated approach for determining candidate anomalies; while most spatial scan statistics use a completely data agnostic approach to determine candidates (e.g., for the basic spatial scan statistics, any circular region, regardless of the sparsity or density of data within it, was regarded as a candidate), the LHA method for finding homogeneous regions is applicable only for gridded data. Since we are ultimately interested in characterizing groups of objects, rather than spatial regions, it is intuitive to think of modifying the candidate discovery phase by ideas from density-based clustering algorithms such as DBSCAN. Specifically, one could run DBSCAN on the whole dataset using the spatial attributes only; the clusters generated therein can then used as candidate regions for anomalies which would then be scored by statistical methods

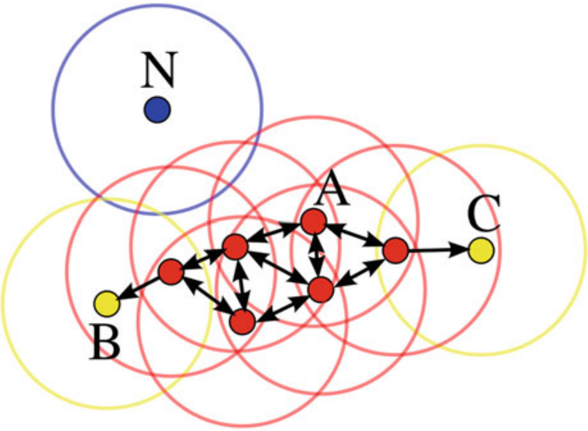

Fig. 6 DBSCAN example (from http://commons.wikimedia.org/wiki/File:DBSCAN-Illustration.svg)

such as those used in SaTScan and LHA. Motivated by this application, we will briefly introduce density-based clustering algorithms in this section, to complete the discussion on anomaly detection for spatial data.

Density-Based Clustering Density-based clustering is a family of algorithms that started with DBSCAN [10]. This algorithm takes in two parameters ϵ and τ, and determines the clustering of a data object based on whether there are at least τ points within an ϵ distance from it. It may be noted that ϵ is a distance threshold applied on the spatial attributes only. The algorithm scans the database for unclustered points, and works by considering such points one by one. Consider the example in Fig. 6 where the search for a cluster starts at the point A. All circles in the figure are of radius ϵ and the example uses $\tau = 3$. Since there are at least 3 points within ϵ distance from A, all data points within ϵ distance are added to A's cluster. All of those points that have at least τ neighbors within an ϵ distance from them are used to further extend the search outward to include more points in the cluster. As shown in the figure, the cluster growth stops after including the points B and C, since neither of them satisfy the ϵ density constraint; in this case, however, there isn't even one new point within ϵ radius of either B and C. The point N in the example, gets labeled as a *noise* point since it does not satisfy the ϵ density constraint nor is it in the neighborhood of a cluster object that satisfies the ϵ density constraint. Since DBSCAN, there have been various extensions, notably OPTICS [1], that serves to identify clusters in data where objects are distributed in clusters of varying density.

8 Directions for Future Work

While the literature on anomaly detection has grown tremendously over the last fifteen years, there are still a lot of interesting problems yet to be addressed in the space of methods to detect anomalous regions. We will outline some potential research directions, with the hope of encouraging more work in this space.

- **Tracking Spatial Anomalies over Time:** Many of the techniques discussed in this chapter have been designed for identifying spatially coherent anomalous regions. A trivial extension of these to cover the time attribute would be to find spans in the time attribute for which these anomalies exist in space. For circular anomalies like those of the basic spatial scan statistic, these could give cylinders. However, such an approach does not allow to capture spatio-temporal anomalies that travel in space with time. For example, a weather anomaly caused by a tornado could travel in space with time, and thus, such a pattern would get split into multiple anomalies each of which have a small time span, under the trivial time extension method. Devising methods to extend spatial anomaly detection to include time in a more sophisticated manner would be an interesting direction for future work. Instead of just a timespan, the time model for an anomaly could be the direction and speed of the movement of the spatial anomaly.
- **Design of Neighborhood Region:** Local neighborhood is defined for local outliers using the set of adjacent cells in tessellated data, or as the k nearest neighbors in multi-dimensional point data. LHA uses a sheet of specified width as the neighborhood, which roughly extends the point neighborhood idea to regions. However in cases where the anomalies are not regularly shaped, this neighborhood may turn out to be non-intuitive. For example, see Fig. 7 where the middle black candidate hotspot is shown along with its sheet neighborhood in grey; note that the colors are used to demarcate the objects, and not to signify value attribute intensity as we have done previously. Consider two portions of the object and the neighborhood, those under *A* and *B*, respectively. Majority of the region covering the neighborhood falls under *A*, whereas majority of the region covering the candidate hotspot falls under *B*. Thus, the comparison approach ends up comparing mostly spatially distant regions while actually intending to

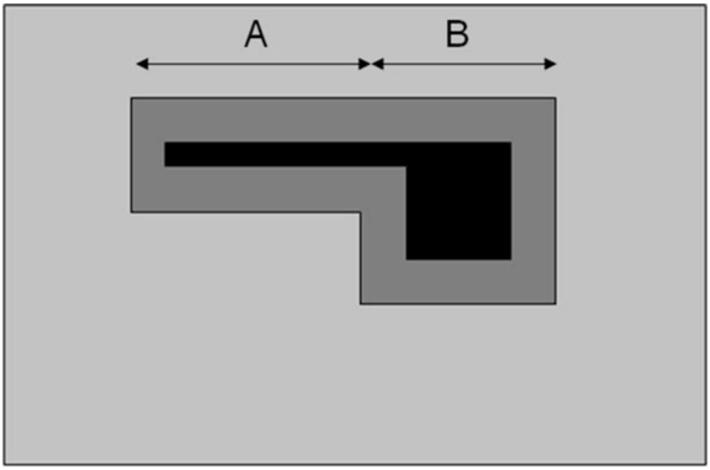

Fig. 7 Issues with local neighborhood definition

compare a region and its neighborhood. This is so since the long left extension of the candidate hotspot brings in a larger region to the neighborhood as compared to its fractional contribution to the anomaly itself. Addressing the neighborhood definition so that it remains intuitive even under arbitrarily shaped anomalies would lead to interesting directions.

- **Anomaly Detection with Varying Local Neighborhoods:** The local neighborhood definition would itself typically involve a parameter; this is typically k (in kNN) for outlier detection methods, and the sheet width ρ for LHA. Instead of fixing the value of such a parameter, it would be interesting to explore anomaly detection where the sheet width is variable. For example, we would then be asking for both an anomaly as well as the neighborhood sheet width under which its anomalous-ness is maximized, as the output.

9 Conclusions

In this chapter, we looked at various methods for anomaly detection. Table 1 summarizes the subset of techniques that were considered, in some detail. We started off by outlining a taxonomy of methods for anomaly detection, focusing on classifying them under features such as the neighborhood used for comparison, and size restrictions for the anomaly. This was followed by a discussion of point-anomaly (i.e., outlier) detection methods covering density based and spatial outliers. We then looked at global anomaly detection methods which is mostly due to the statistics community, where the literature around spatial scan statistics were developed. Our attention was then on discussing methods to find anomalies that contrast well with a local neighborhood. Under this, we saw the LHA method and discussed applicability of image segmentation methods. In the section on grouping-based anomaly detection, we outlined spatial data clustering methods, as well as anomaly detection methods developed for intrusion detection. We then discussed

Table 1 List of techniques for anomaly detection covered in this chapter

Category	Methods
General object anomalies	LOF [4]
Spatial object anomalies	SLOM [7]
Region anomalies—global	Spatial scan statistic [20]
	ULS scan statistic [25]
	Bump hunting [12]
Region anomalies—local	LHA [33]
Region anomalies—grouping	Clustering ensuring spatial convexity [8]
	CLAD [23]
Density-based clustering	DBSCAN [10]

how bottom-up clustering methods, such as those for density-based clustering, could be used to improve anomaly detection. This was followed by an outline of potential research direction in anomaly detection. We hope that this chapter provides the interested reader with a mixture of different perspectives towards the problem of anomaly detection. Apart from encouraging more research in this exciting area, we also hope that this would help accelerate cross-pollination of methods across communities towards extending the frontier in anomaly detection on spatial data.

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: Optics: ordering points to identify the clustering structure. In: ACM Sigmod Record, vol. 28, pp. 49–60. ACM, New York (1999)
2. Balachandran, V., Deepak, P., Khemani, D.: Interpretable and reconfigurable clustering of document datasets by deriving word-based rules. Knowl. Inf. Syst. **32**(3), 475–503 (2012)
3. Bonnet, N., Cutrona, J., Herbin, M.: A 'no-threshold' histogram-based image segmentation method. Pattern Recogn. **35**(10), 2319–2322 (2002)
4. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: ACM Sigmod Record, vol. 29, pp. 93–104. ACM, New York (2000)
5. Celebi, M.E.: Partitional Clustering Algorithms. Springer, New York (2015)
6. Chaudhary, A., Szalay, A.S., Moore, A.W.: Very fast outlier detection in large multidimensional data sets. In: DMKD (2002)
7. Chawla, S., Sun, P.: Slom: a new measure for local spatial outliers. Knowl. Inf. Syst. **9**(4), 412–429 (2006)
8. Deepak, P., Deshpande, P., Visweswariah, K., Telang, A.: System and method for clustering ensuring convexity in subspaces. Prior Art Database (IP.COM) (2013)
9. Duczmal, L., Assuncao, R.: A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. Comput. Stat. Data Anal. **45**(2), 269–286 (2004)
10. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol. 96, pp. 226–231 (1996)
11. Fan, J., Yau, D.K., Elmagarmid, A.K., Aref, W.G.: Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Trans. Image Process. **10**(10), 1454–1466 (2001)
12. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Stat. Comput. **9**(2), 123–143 (1999)
13. Glaz, J., Pozdnyakov, V., Wallenstein, S.: Scan Statistics: Methods and Applications. Springer, Berlin (2009)
14. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 469–475 (2006)
15. Guha, S., Rastogi, R., Shim, K.: Rock: a robust clustering algorithm for categorical attributes. In: Proceedings of the 15th International Conference on Data Engineering, 1999, pp. 512–521. IEEE, New York (1999)
16. Hawkins, D.M.: Identification of Outliers, vol. 11. Springer, New York (1980)
17. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recogn. Lett. **24**(9), 1641–1650 (2003)
18. Jiang, M.-F., Tseng, S.-S., Su, C.-M.: Two-phase clustering process for outliers detection. Pattern Recogn. Lett. **22**(6), 691–700 (2001)
19. Knox, E.M., Ng, R.T.: Algorithms for mining distance based outliers in large datasets. In: Proceedings of the International Conference on Very Large Data Bases, pp. 392–403. Citeseer, New York (1998)

20. Kulldorff, M.: A spatial scan statistic. Commun. Stat. Theory Methods **26**(6), 1481–1496 (1997)
21. Kulldorff, M., Huang, L., Pickle, L., Duczmal, L.: An elliptic spatial scan statistic. Stat. Med **25**(22), 3929–3943 (2006)
22. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, vol. 1, pp. 281–297 (1967)
23. Mahoney, M.V., Chan, P.K., Arshad, M.H.: A machine learning approach to anomaly detection. Technical report, Tech. rep. CS–2003–06, Department of Computer Science, Florida Institute of Technology Melbourne (2003)
24. Neill, D.B., Moore, A.W., Cooper, G.F.: A bayesian spatial scan statistic. Adv. Neural Inf. Process. Syst. **18**, 1003 (2006)
25. Patil, G., Taillie, C.: Upper level set scan statistic for detecting arbitrarily shaped hotspots. Environ. Ecol. Stat. **11**(2), 183–197 (2004)
26. Portnoy L., Eskin E., Stolfo S.: Intrusion detection with unlabeled data using clustering. In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). pp. 5–8 (2001)
27. Revol, C., Jourlin, M.: A new minimum variance region growing algorithm for image segmentation. Pattern Recogn. Lett. **18**(3), 249–258 (1997)
28. Schubert, E., Zimek, A., Kriegel, H.-P.: Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Min. Knowl. Disc. **28**(1), 190–237 (2014)
29. Shaw, J.R.: Quickfill: an efficient flood fill algorithm. The Code Project (2004)
30. Shekhar, S., Lu, C.-T., Zhang, P.: Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 371–376. ACM, New York (2001)
31. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
32. Tango, T., Takahashi, K.: A flexibly shaped spatial scan statistic for detecting clusters. Int. J. Health Geogr. **4**(1), 11 (2005)
33. Telang, A., Deepak, P., Joshi, S., Deshpande, P., Rajendran, R.: Detecting localized homogeneous anomalies over spatio-temporal data. Data Min. Knowl. Disc. **28**(5–6), 1480–1502 (2014)
34. Turnbull, B.W., Iwano, E.J., Burnett, W.S., Howe, H.L., Clark, L.C.: Monitoring for clusters of disease: application to leukemia incidence in upstate new york. Am. J. Epidemiol. **132**(Suppl. 1), 136–143 (1990)
35. Yu, D., Sheikholeslami, G., Zhang, A.: Findout: finding outliers in very large datasets. Knowl. Inf. Syst. **4**(4), 387–412 (2002)
36. Zepeda-Mendoza, M.L., Resendis-Antonio, O.: Hierarchical agglomerative clustering. In: Encyclopedia of Systems Biology, pp. 886–887. Springer, Berlin (2013)